Thaisa Cardoso Lacerda

A METHOD FOR SELF-ASSESSING THE CAPABILITY OF THE USABILITY PROCESS IN SMALL ORGANIZATIONS

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Ciência da Computação.
Orientadora: Prof. Dr. rer. nat. Christiane Gresse von Wangenheim, PMP.
Coorientador: Prof. Dr. Jean Carlo Rossa Hauck

Florianópolis
2018

Dedicated to those who supported me to achieve this goal.

ACKNOWLEDGMENTS

To God, for the life, for the family that He has granted me and for the privilege of being able to study and to be able to improve myself as a professional.

To my parents, who, besides giving me love and affection, grant me the most important gifts a child can receive, education in the Christian faith and encouragement to always seek knowledge.

To professor Christiane for the orientations, revisions, suggestions that contributed in a unique way not only for the improvement of the quality of the research developed, but also for my development as a professional.

I thank all those who contributed to this research, participating in the process assessments or sharing their knowledge and expertise on this subject, and thus adding more value to the results obtained.

RESUMO

Um mundo cada vez mais transformado e conectado digitalmente representa desafios cada vez mais significativos para as empresas de TI, exigindo maior atenção à usabilidade dos seus produtos de software. Investir no desenvolvimento de software com uma melhor usabilidade, pode reduzir o custo total de desenvolvimento, evitando o retrabalho nas etapas finais do ciclo de vida e acelerando o desenvolvimento. Além disso, produzir produtos com uma melhor usabilidade pode oferecer uma vantagem competitiva aumentando as vendas e a retenção de clientes, aumentando a satisfação do usuário e a aceitação de software no mercado. Assim, é fundamental que as empresas de software estabeleçam seus processos de usabilidade de forma sistemática. Uma das melhores maneiras para uma empresa iniciar um programa de melhoria de processo de software é realizando uma avaliação do seu processo. Essas avaliações geralmente são realizadas por avaliadores externos à empresa. No entanto, uma alternativa para as empresas que procuram avaliações mais leves, especialmente as pequenas empresas, são auto-avaliações. As auto-avaliações podem ser realizadas pela própria empresa para avaliar a capacidade de seu processo. Embora existam métodos de avaliação específicos para avaliar o processo de usabilidade, nenhum deles fornece um método de auto-avaliação, além disso, nenhum deles foi desenvolvido considerando as características específicas das pequenas empresas. Portanto, o objetivo desta pesquisa é propor um método prescritivo para auto-avaliação da capacidade do processo de usabilidade em pequenas empresas. O método contém um modelo de referência de processo de engenharia de usabilidade, uma estrutura de medição, um modelo de avaliação e um processo de auto-avaliação. Para desenvolver tal método, foram realizados mapeamentos sistemáticos da literatura sobre métodos de capacidade/maturidade de processos de usabilidade e métodos de auto-avaliação de processos de software. Uma vez que o estado da arte foi sintetizado, o método para auto-avaliação da capacidade do processo de usabilidade de pequenas empresas (chamado UPCASE) foi desenvolvido usando uma metodologia específica para customizar os modelos de capacidade/maturidade de processo de software e para o design de questionários. Para orientar o desenvolvimento da UPCASE, foram estabelecidas uma série de requisitos para garantir que o método atenda às necessidades das pequenas empresas e que fato avalie a capacidade do processo de usabilidade. O método proposto contém um modelo de referência de processo (PRM), uma estrutura de medição e um modelo de avaliação de processo. A estrutura do método é elaborada com base em ISO/IEC 29110-3 e ISO/IEC 15504. O conteúdo do PRM é baseado em ISO/IEC 18529, um relatório técnico descrevendo um modelo de referência de processo para o processo de design centrado no usuário. Para avaliar o método de avaliação proposto em termos de sua confiabilidade, usabilidade, compreensão e consistência interna, uma série de estudos de caso foi realizada. Os resultados fornecem uma primeira indicação de que o método proposto tem boa usabilidade e compreensão a serem aplicadas por avaliadores não especialistas e que seu questionário é confiável e que seus itens apresentam consistência interna.

Palavras-chave: avaliação de processos, modelos de maturidade/capacidade, pequenas empresas, usabilidade, design centrado no usuário.

RESUMO EXPANDIDO

Introdução

Usabilidade é a medida em que um produto pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso. Investir em usabilidade ao projetar software por meio de um processo de projeto centrado no usuário pode reduzir os custos gerais de desenvolvimento, evitando o retrabalho nos últimos estágios do ciclo de vida e acelerando o desenvolvimento. Além disso, a usabilidade pode fornecer uma vantagem competitiva aumentando as vendas e retendo os clientes, aumentando a satisfação do usuário e a aceitação do software. Assim, a questão é como desenvolver aplicativos de software com alta usabilidade? Como qualquer outra qualidade de produto, a usabilidade é diretamente influenciada pela qualidade do processo de software e, portanto, é importante definir e implementar processos de usabilidade apropriados. O processo de usabilidade contém definições de artefatos e atividades (que transformam entradas em saídas) necessárias para o desenvolvimento de um produto que permite a interação com os usuários. Para orientar a definição, a implementação e o aprimoramento dos processos de software, geralmente são usados os modelos de capacidade/maturidade (SPCMMs). Motivados pela necessidade de métodos de avaliação menos complexos e mais ágeis, métodos de avaliação mais leves, que permitem a realização de auto-avaliação, tem sido desenvolvidos, tipicamente para pequenas empresas ou organizações que usam métodos de desenvolvimento ágil. As auto-avaliações são a maneira mais comum de realizar um SPA em organizações que não têm como objetivo a certificação. Eles são executados por uma organização para avaliar a capacidade/maturidade de seu próprio processo, não exigindo o envolvimento de especialistas externos da SPI.

Objetivos

Esta pesquisa tem como objetivo desenvolver e avaliar um método de auto-avaliação para avaliar a capacidade do processo de usabilidade em pequenas organizações (SEs), de acordo com a ISO/IEC 29110, a fim de responder à questão de pesquisa: é possível avaliar a capacidade do processo de usabilidade de pequenas organizações através de auto-avaliações?

Metodologia

Os procedimentos metodológicos desta pesquisa são definidos com base na proposta de Saunders (2009), que classifica o método científico em camadas. Em termos de filosofia, esta pesquisa é predominantemente interpretativa, pois se assume que o objeto de pesquisa (autoavaliação dos processos de usabilidade) é interpretado e avaliado do ponto de vista social pelos atores envolvidos na pesquisa (avaliadores de processos). No que se refere à abordagem de pesquisa, utiliza-se uma abordagem indutiva, pois não faz parte de uma hipótese pré-estabelecida, mas busca alcançar a solução do problema a partir das inferências dos estudos de caso do objeto de estudo. Como estratégia de pesquisa, são adotados multi-métodos, pois são utilizados diferentes métodos qualitativos e quantitativos, como pesquisa bibliográfica, mapeamento sistemático, estudo de caso, GQM (Objetivo/Question/Métrica), Metodologia de Personalização do processo de software/maturidade (SPCMM), diretrizes para o desenho de questionários e análise estatística. Do ponto de vista de sua natureza, trata-se de uma pesquisa aplicada, pois visa gerar conhecimento para aplicação prática, contribuindo para a solução de problemas específicos que ocorrem na realidade (MARCONI; LAKATOS, 2010). Em relação aos objetivos da pesquisa, ela é classificada como exploratória, pois está interessada em proporcionar uma maior compreensão do fenômeno que é investigado.

Resultados e Discussão

Os resultados obtidos fornecem uma primeira evidência da confiabilidade, usabilidade, compreensibilidade e consistência interna do método UPCASE, aceitável para avaliar a capacidade do processo de usabilidade do SE. Em relação à confiabilidade da UPCASE, analisando o coeficiente de correlação intraclasse, os resultados gerais indicam uma reprodutibilidade aceitável de medidas quantitativas feitas por diferentes observadores usando UPCASE. Apenas os itens do questionário com relação à UP1 obtiveram resultados inconclusivos, para ser capaz de analisá-lo será necessário realizar mais estudos de caso. A análise do coeficiente kappa ponderado indicou que a UPCASE possui uma boa e boa concordância entre avaliadores, ou seja, a UPCASE é um método adequado para avaliar o mesmo objeto em diferentes situações, permitindo a concordância entre os avaliadores. No geral, o resultado indicou que o questionário UPCASE apresenta um grau razoável de confiabilidade quando utilizado em momentos diferentes para avaliar o mesmo objeto. Também observamos que, em geral, os SEs participantes dos estudos de caso observados puderam realizar o processo de avaliação conforme o esperado, utilizando apenas o material fornecido, em um período de tempo razoável, deixando os participantes satisfeitos com o processo usado para realizar a avaliação. Apenas uma das SEs não realizou o processo de avaliação como esperado. Todas as reuniões de avaliação foram realizadas em um período de tempo razoável, em torno de 1 hora; e os participantes sentiram-se satisfeitos. Dessa forma, esses resultados indicam que o UPCASE possui boa usabilidade. Quanto à facilidade de compreensão do conteúdo do método UPCASE, a fim de permitir a realização de uma correta reunião de avaliação e a correta finalização do questionário de avaliação, verificou-se que nos estudos de caso os participantes tiveram dificuldades em compreender o itens 2, 3 e 4, portanto, esses itens precisam ser revisados. Além disso, exceção para um dos quatro estudos de caso, todos eles realizaram o processo de avaliação adequadamente. Portanto, pode-se concluir que o método UPCASE possui uma compreensão adequada. Em termos de consistência interna, os resultados da análise indicam um α de Cronbach satisfatório, indicando que o conjunto de itens do questionário UPCASE está medindo um único fator de qualidade. Assim, evidenciando que os itens do questionário UPCASE são consistentes e precisos no que diz respeito à avaliação da capacidade do processo de usabilidade do SE.

Considerações Finais

Observando uma lacuna no estado da arte atual para métodos de avaliação de processo usabilidade para pequenas empresas, este estudo objetivou desenvolver o UPCASE, o método de autoavaliação da capacidade do processo de usabilidade em SEs. O método baseia-se na norma ISO/IEC 29110 e ISO/IEC 18529 e é personalizado em relação a requisitos específicos de SEs. O UPCASE inclui um modelo de referência de processo de usabilidade personalizado para SEs, uma estrutura de medição e um modelo de avaliação, que inclui um questionário, um roteiro de avaliação, um glossário e exemplos de produtos de trabalho a serem usados como suporte durante a avaliação. Além do desenvolvimento do método de avaliação, foi desenvolvida uma ferramenta on-line gratuita para facilitar a aplicação do método por organizações que desejam auto-avaliar seus processos, bem como às respostas dos questionários de coleta de dados. O UPCASE foi aplicado em quatro estudos de casos observados e em 32 estudos de casos remotos, nos quais as organizações de software auto-avaliaram seu processo de usabilidade utilizando a ferramenta on-line desenvolvida. Analisando os dados coletados nos estudos de caso, obtivemos resultados de uma avaliação inicial que indica que o método é confiável, tem boa usabilidade

e compreensibilidade e possui um questionário com uma consistência interna aceitável.
Palavras-chave: avaliação de processos, modelos de maturidade/capacidade, pequenas empresas, usabilidade, design centrado no usuário.

ABSTRACT

A world becoming more digitally transformed and connected poses significant challenges for IT organizations, requiring increased attention to the usability of their software products. Investing in developing software with better usability, can reduce overall development cost by avoiding rework at late stages in the lifecycle and speed up development. Moreover, providing products with better usability can offer a competitive edge increasing sales and retaining customers, increasing user satisfaction and software acceptance. Thus, it is of fundamental importance that software organizations systematically establish usability processes. One of the best ways for an organization to start a software process improvement program is performing a process assessment. These assessments are typically performed by external assessors. Yet, an alternative for organizations seeking for lighter assessments, especially the small organizations, are self-assessments. Self-assessments can be carried out by an organization on its own to assess the capability of its process. Although there are specific assessment methods to assess the usability process, none of them provides a self-assessment method, nor has been developed considering the specific characteristics of small organizations. Therefore, the objective of this research is to propose a method for self-assessing the capability of the usability process in small organizations. The method contains a usability engineering process reference model, a measurement framework, an assessment model, and a self-assessment process. To develop such method, systematic mapping studies on usability capability/maturity models and software process self-assessment methods were performed. Once the state-of-the-art was synthesized, the method for self-assessing the capability of the usability process of small organizations (UPCASE) was developed using a methodology specific for customizing software process capability/maturity models and for questionnaire design. To guide the development of UPCASE, a series of requirements have been established to ensure that the method meets the needs of small organizations and in fact measures the capability of the usability processes. The proposed method contains a Process Reference Model (PRM), a Measurement Framework and a Process Assessment Model. The structure of the method is elaborated based on ISO/IEC 29110-3 and ISO/IEC 15504. The content of the PRM is based on ISO/IEC 18529, a technical report describing a process reference model for the human-centred design process. In order to evaluate the proposed assessment method UPCASE in terms of its reliability, usability, comprehensibility and internal consistency a series of case studies has been performed. Results provide a first indication that the proposed method has good usability and comprehensibility to be applied by non-expert assessors, and that its questionnaire is reliable and has internal consistency.

Keywords: process assessment, capability/maturity model, small organizations, usability, human-centered-design.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AM – Assessment method
GQS – Grupo de qualidade de software
GQM – Goal/Question/Metric
PAM – Process assessment model
PRM – Process reference model
SE – Small organizations
SPCMM – Software process capability/maturity models
SPA – Software process assessments
SPI – Software process improvement
SW – Software
UE – Usability engineering
UCMM – Usability capability/maturity model
UPCASE – Method for self-assessing the capability of the usability process in SE
USPA – Usability sub-process percentage of achievement
UPPA – Usability process percentage of achievement

SUMMARY

1 INTRODUCTION

This chapter presents an introduction to the development of a method to perform self-assessment of the capability of the usability process in small organizations (SE).

1.1 CONTEXTUALIZATION

Software applications nowadays are present in a diverse range of devices, such as computers, tablets, mobile phones, digital TVs, refrigerators, etc. for numerous kinds of activities, from researching a health condition, entertainment to accessing educational resources (KRUMM, 2016). Such changes have a significant impact on the nature of user interaction, as they offer new ways of interaction anywhere, anytime by anyone (POUS; CECCARONI, 2010; WASSERMAN, 2010). This, on the other hand, makes usability an even more important quality attribute of software today (POUS; CECCARONI, 2010; TREERATANAPON, 2012).

Usability is the extent to which a product can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a specific context of use (ISO/IEC, 2001a). Usability flaws may impede the users to complete their tasks or annoy them when interaction is designed unnecessarily complex or time-consuming (GRINDROD; LI; GATES, 2014). Furthermore, in critical contexts, such as health applications, which may impact on the health of humans, usage errors may compromise patient safety leading to injury or even death (MARCILLY et al., 2015). On the other hand, investing in usability by designing software through a user-centered design process can reduce overall development cost by avoiding rework at late stages in the lifecycle (NIELSEN, 1994) and speed up development (BIEL; GRILL; GRUHN, 2010; CONKLIN, 1995). Moreover, usability can provide a competitive edge increasing sales and retaining customers, increasing user satisfaction and software acceptance (BIAS; MAYHEW, 2005; JOKELA, 2004a). Thus, the question is of how to develop software applications with high usability?

As any other product quality, usability is directly influenced by the software process quality (MORAIS; SBRAGIA, 2012), and, thus it is important to define and implement appropriated usability processes. The usability process contains definitions of artifacts and activities (which transform inputs into outputs) required for the development of a product that allows interaction with the users.

To guide the definition, implementation and improvement of software processes, typically capability/maturity models (SPCMMs), such as CMMI (TEAM, 2010), and the ones provided by ISO/IEC 15504 (ISO/IEC, 2004) or ISO/IEC 29110 (ISO/IEC, 2016a) are used. SPCMMs aim at supporting organizations to define and continually improve their process using software engineering best practices. Software process improvement is a way to increase product quality, developing software with fewer resources in less time, improve productivity, increase organizational flexibility and customer satisfaction and, thus, to allow software organizations to stand out from competitors (HARTER; KEMERER; SLAUGHTER, 2012) (BOEHM, 2006; KUILBOER; ASHRAFI, 2000)(KALINOWSKI et al., 2014; PAULK, 1995; SOLINGEN, 2004). One of the best ways for an organization to start a software process improvement (SPI) program is to perform a process assessment in order to elicit the gap between its current practices and the ones indicated by a reference model (MCCAFFERY; TAYLOR; COLEMAN, 2007)(MACMAHON; MC CAFFERY; KEENAN, 2015). The result of such an assessment is an indicator of how well the organization's processes meet the requirements of the process reference model (KOMI-SIRVIÖ, 2004) indicating also the improvement opportunities.

Besides generic SPCMMs intended to be applicable in any context, we can also observe a trend to the customization of such models to target more specifically certain contexts. So far diverse kinds of customizations of SPCMMs have been developed (VON WANGENHEIM et al., 2010), such as for certain types of systems, including medical devices (ISO/IEC, 2003a), e-Government (HUANG; XING; YANG, 2008) or automotive sector (SIG AUTOMOTIVE, 2010) or focusing on specific quality aspects, such as systems security (ISO/IEC, 2008a), user centered design process or specific types of software development approaches, such as hybrid traditional-agile approaches (GILL; HENDERSON-SELLERS; NIAZI, 2016) among others. In such cases customized models may provide more specialized support facilitating process improvement and assessment by adapting process requirements and/or providing further support for their application, for example, through low cost assessment methods or reducing the need for documentation (PAULK, 1998; TEAM, 2010; TIM, 2004). However, taking into account that usability is an important software product quality characteristic, it seems that neither generic SPCMMs nor customized SPCMMs include processes specifically aiming at usability (RABELLO et al., 2012). This means, that, even software organizations at the highest level of maturity seem not to be required to have established any usability engineering processes (JOKELA; LALLI, 2003).

On the other hand, there exist few SPCMMs focusing exclusively on usability processes (such as UCDM (EASON; HARKER, 1997), ULMM (FLANAGAN, 1995), UMM-P (EARTHY, 1999)) (JOKELA et al., 2006). Although these models specify high-level requirements to the usability process, they seem not to provide enough information on how to implement them in practice, which may hinder a large-scale adoption. And, although such generic SPCMMs are supposed to be applicable in any kind of context, it remains questionable, if they are also valid, reliable and cost efficient in current software development contexts due to a lack of application and validation of these models (JOKELA et al., 2006).

Furthermore, besides the popularity of the capability/maturity models, they are mostly applied in large organizations, not becoming popular among SE and/or agile enterprises (LARRUCEA et al., 2016). This may be due to their detailed assessment procedure requiring considerable effort with significant costs, making their adoption often impossible for SEs (KAR et al., 2012)(JOKELA, 2002) (CHANG, 2001) (YUCALAR; ERDOGAN, 2009) (MCCAFFERY; TAYLOR; COLEMAN, 2007) (VON WANGENHEIM; C. ANACLETO, 2006) (ISO/IEC, 2016b) (ABUSHAMA, 2016). In addition, organizations believe that software process assessments (SPA) require certain degree of detail that increases corporate bureaucracy (KROEGER; DAVIDSON; COOK, 2014). Another reason that makes SPAs less attractive to small organizations is the difficulty of understanding and implementing them in practice (YUCALAR; ERDOGAN, 2009) (HABRA et al., 2008)(KALPANA; JEYAKUMAR, 2010). This fact leads many organizations to seek even more for simplicity within the organizations' processes and as result they are increasingly attracted to agile methods. In this context, agile principles should be incorporated into any potential PAM (MCCAFFERY; TAYLOR; COLEMAN, 2007). Despite these challenges that many organizations encounter to assess their processes, SPAs cannot be discarded as they allow to identify the organization's strengths and weaknesses and, thus, are a fundamental input to start a software improvement program (HABRA et al., 2008) (KAR et al., 2012).

Motivated by the need for less complex and more agile assessment methods, lighter assessment methods are developed, typically for SEs or organizations that use agile development methods, in form of self-assessments. Self-assessments are the most common way to conduct a SPA in organizations that does not aim for certification (PATEL; RAMACHANDRAN, 2009). They are carried out by an organization to assess the capability/maturity of its own process, not requiring the involvement of external SPI experts. The sponsor of a self-assessment is normally internal to the organization as are the member(s) of the assessment team responsible for collecting and analyzing data and reporting the assessment results (ISO/IEC, 2004). Data collection is typically done by using a single data collection method such as questionnaire, interview or workshop. Data analysis depends on the specific assessment purpose, for instance, to find gaps between the organization's current practices and the process assessment model (PAM) (ISO/IEC, 2004), to educate the organization on the requirements of a formal assessment method or to perform a benchmarking against other organizations (BLANCHETTE; KEELER, 2005).

The popularity of self-assessments relies in their low cost and good accessibility (ABUSHAMA, 2016; PATEL; RAMACHANDRAN, 2009). As self-assessments use the organization's own human resources and are less bureaucratic, they enable a more simplified way to perform a process assessment which can be performed in a shorter period of time with fewer resources (ABUSHAMA, 2016; PINO et al., 2010). In addition, in organizations where capability is a new concept, self-assessments allow an easy way to improve the process (BLANCHETTE; KEELER, 2005), as they do not significantly intrude the daily routine of the organization. Self-assessment is also effective in generating an "ownership feeling" among managers, regarding the process quality, as it forces them to examine their own activities (BLANCHETTE; KEELER, 2005; DER WIELE et al., 2000).

Despite the benefits presented, self-assessments are not without shortcomings. Organizations using self-assessments found difficulties in planning the assessment process and allocating human recourses to lead and execute them. Another difficulty is the scarcity of literature regarding the "best" approach to perform a self-assessment, as there is no guidance on which self-assessment method organizations should be used (RITCHIE; DALE, 2000). This leads to the conclusion that applying a variety of data collection methods can be useful to construct a more comprehensive picture of the needs and gaps that need to be addressed (GOETHALS, 2013). Another concern when performing self-assessment is often the absence of competent assessors. As the assessors in self-assessment are not necessarily experts in process assessment and may not be familiar with SPCMMs, there is a considerable risk of misinterpretation of the attributes to be assessed, which may impact the validity of the results of the assessment (BLANCHETTE; KEELER, 2005). Therefore, data collection instruments used in self-assessments must be explanatory in a way that non-experts may understand the items to be measured sufficiently to correctly judge their degree of performance, e.g. preventing to wrongly considering a Gantt chart to be a project plan. Furthermore, the response scale has also to be defined carefully, as assessors in self-assessment may not have sufficient experience to correctly classify the degree of satisfaction of an item on a finer grained scale (SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009), e.g. deciding between partially achieved and largely achieved. Thus, if two or more points on a scale appear to have the same meaning, respondents may be puzzled about which one to select, leaving them open to making an arbitrary choice (KROSNICK; PRESSER, 2010). In order to minimize the assessment effort, data collection instruments should also be comprehensive enough to measure the essential information, but at the same time be succinct enough to encourage their completion.

1.2 PROBLEM

An important step towards improving a process is to assess its current status and, thereby, identify what needs to be improved. Such assessments are typically performed using well-established models based on CMMI and/or ISO/IEC 15504 (YUCALAR; ERDOGAN, 2009). For SE, however, these models may not be attractive and, thus, not largely used (LARRUCEA et al., 2016). Thus, there exists a trend towards developing lighter approaches allowing the performance of self-assessments. Self-assessment is a more attractive assessment method for organizations that do not aim for certification, and so can carry out assessments more quickly and with lower cost.

As usability is an important factor, nowadays, for successful software development, there is some research on the development of method for assessing the usability process. These methods, however, tend to be generic, not focusing on a domain. In this way, no custom usability process assessment method was found for small organizations. In addition, no related research has been found that proposes a usability process self-assessment method. Thus, this indicates a lack of a method to assess the usability capability in small organizations which to not intend to obtain a certification nor to hire an external consultant.

1.3 OBJECTIVES

This research aims at developing and evaluating a self-assessment method for assessing the capability of the usability process in small organizations (SEs), in accordance with ISO/IEC 29110, in order to answer the research question: is it possible to assess the capability of the usability process of small organizations through self-assessments?

To achieve this, the following specific objectives are defined:

O1. Synthesize the state-of-the-art of usability processes assessment methods.

O2. Synthesize the state-of-the-art of SW processes self-assessment methods.

O3. Develop a usability processes reference model focusing on SEs.

O4. Develop a usability processes measurement framework focusing on SEs.

O5. Develop a usability processes self-assessment process focusing on SEs.

O6. Apply and evaluate the self-assessment method in SEs.

1.4 DELIMITATION AND SCOPE OF WORK

The method for self-assessing the capability of the usability process of small organizations is exclusively to assess the usability process in small software development organizations. The assessment method is prescriptive, defining how the assessment process should be performed. As an output of the self-assessment, it is expected to obtain a score, which represents the extent to which the organization's processes are aligned with the assessment model, as well as indicating which sub-processes or practices have a low level of capability. Thus, this work does not cover the assessment of usability process in large organizations or in organizations that develop other products than software systems.

1.5 CONTRIBUTIONS

This work has two major scientific contributions. The first one is the synthesis of the state-of-the art on usability capability/maturity models and software process self-assessment methods. This synthesis indicated a gap in the current field of self-assessment of usability process in small organizations, which this research aims to cover. Another contribution is the UPCASE method itself, a method for self-assessing the capability of the usability process of small organizations. Thus, within the research area of Software Engineering of the PPGCC/INE/UFSC, specifically in the area of Software Quality, this work presents the following contributions:

Scientific contribution. The present work results in a self-assessment method for assessing the capability of usability processes in small organizations. The proposed assessment method is composed by an assessment process, which provides 1) a systematic support for the realization of process self-assessments, 2) a usability process reference model and 3) a measurement framework. In addition, the method includes the instruments for collecting data and generation of the assessment results.

The secondary scientific contributions of this research are the results of the two systematic mappings of the literature. Through the systematic mappings, we obtained a survey of the state of the art, identifying existing methods for the assessment of usability processes and for software process assessment using self-assessment methods.

Technological contribution. The main technological contribution of this work is the self-assessment method of the usability process of small organizations. In addition to the assessment method, an online tool was implemented to carry out to support the realization of the assessment. The tool is available in English and Portuguese and covers the entire content of the self-assessment method, description of the assessment process, assessment questionnaire, with description of the items and examples of techniques and work products, glossary; examples of typical work products, and report generation with the assessment results.

Social contribution. In the context of small software organizations, the application of the proposed method allows the identification of the capability of the organization's usability processes through a quick and low self-assessment. In addition, the method allows identifying the weaknesses of the usability process and, thus, creates a basis for a systematic improvement of its quality. By helping small software organizations to improve the quality of their usability processes, it is expected to contribute to the optimization of the resources used in the interface development process, dissemination of knowledge about usability within small software organization, and especially, to the improvement of the usability of the developed products, and consequently the quality of the product available on the market.

1.6 RELATIONSHIP WITH THE PPGCC SOFTWARE ENGINEERING RESEARCH LINE

This section presents the adherence of the subject of this dissertation to the objectives of the Post-Graduation Program in Computer Science of the Federal University of Santa Catarina (PPGCC/UFSC), and more specifically to the objectives of the Software Engineering research line. According to the 1º article of internal regiment nº01/PPGCC/2013, publishing on 01/10/2013, the objectives of the program are (PPGCC/UFSC, 2013):

I. the training of researchers and teachers of higher education in Computer Science and related areas;
II. the development of new knowledge in Computer Science.

Computer Science is an academic discipline that addresses the study of computers and algorithms, including their fundamentals, applications, software and hardware design, and their impact on society (ACM; IEEE CS, 2013). Computer

Science also transcends the integration of technology into teaching, aiding student learning in academic disciplines (CSTA, 2013). In this context, SE is a discipline that integrates Computer Science, understood as the systematic application of quantifiable approaches to the development, operation, and maintenance of software (BOURQUE; FAIRLEY; OTHERS, 2014).

The importance of topics related to usability process improvement within SE is highlighted when analyzing relevant academic events and journals of SE, where there are presence of research lines directed to usability process and process improvement, such as IEEE Transactions on Software Engineering (TSE), International Journal of Human-Computer Studies, European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE), International Conference on Software Engineering (ICSE), International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI) and Conference on Human Factors in Computing Systems (CHI).

In this context, the present dissertation develops new knowledge relevant to Computer Science. The main knowledge produced is the development of a method for self-assessing the usability processes in small organizations, including its application in software development organizations and its evaluation. This knowledge is relevant as the assessment method was developed to target an audience who does not have easy access to resources that enable them to enhance their usability process. Thus, it is expected that the use of appropriate usability processes will allow small organizations to avoid wasting time, and consequently money developing the interface of their software, and therefore, produce products with better degree of usability. In this way getting in advantages over their competitors (BIAS; MAYHEW, 2005; BIEL; GRILL; GRUHN, 2010; CONKLIN, 1995; JOKELA, 2004b).

Other relevant knowledge for Computer Science was also produced, such as the state-of-the-art survey usability process assessment models and software process self-assessment method, through Systematic Mapping Studies. More specifically, when analyzing the objectives of the research line in SE of the PGCC/UFSC, we observe the adherence of the present dissertation to these objectives, which are:

"Software Engineering: aims to qualify individuals capable of conducting the software development process and to investigate new methodologies, techniques and tools for system design" (PPGCC/UFSC, 2015).

In relation to the objective of qualifying individuals capable of conducting the software development process, this dissertation focuses on the availability of an assessment method that allows professionals who do not have usability expertise to obtain feedback on what can be improved in their process. In addition, the method provides other materials that can help small organizations to initiate improvement programs, as it presents the definition of basic concepts, techniques and tools that can be used, thus allowing professionals to assess their processes with this method, consequently ample their knowledge in this area. Therefore, it is understood that there is correlation of dissertation subject to this objective of the software engineering search line.

In relation to the second part of the objective - investigation of new methodologies, techniques and tools for system design - this is evidenced by the state of the art which points to the lack methods for assessing the capability of usability process that may be afforded by small organizations that address methods, techniques and tools, for developing software. Thus, with the adherence of results derived from this dissertation with the objectives of the PPGCC/UFSC and the line of research of Software Engineering, it is understood that this dissertation is adherent to Computer Science.

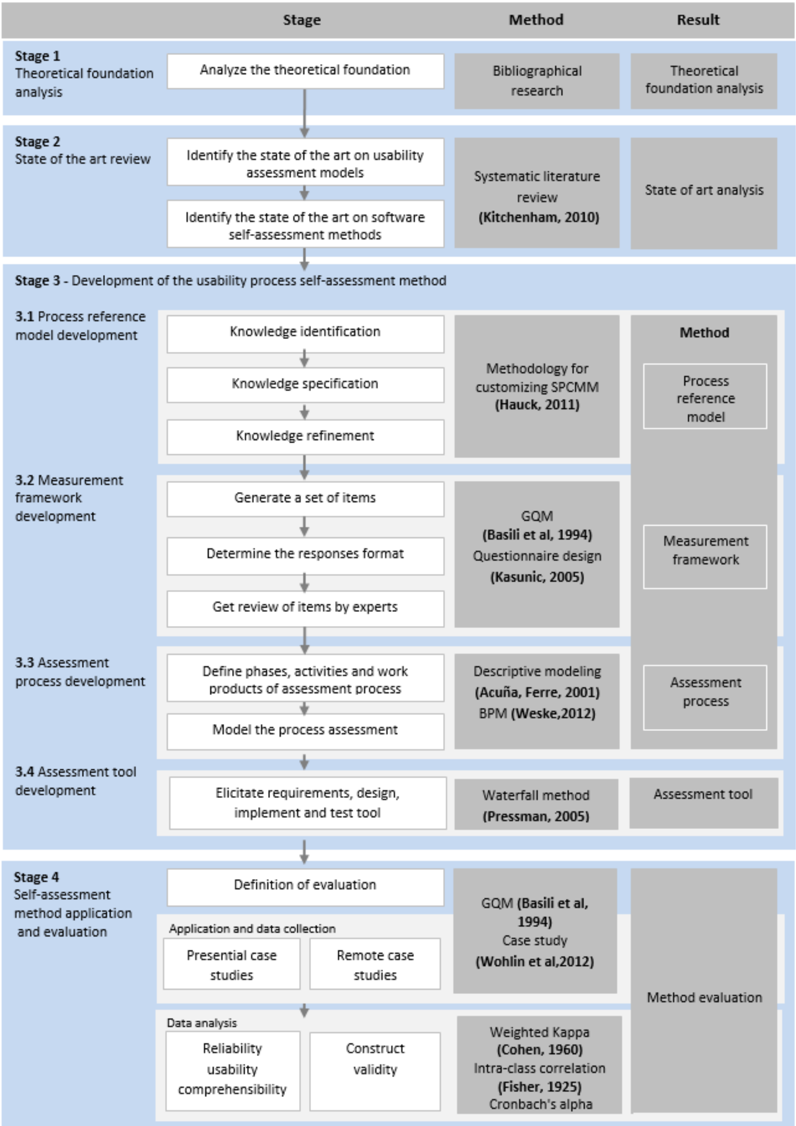
1.7 WORK STRUCTURE

This work is divided into 7 chapters. In the following chapter, it is described the methodology use in this work. In chapter 3 it is described the theoretical foundation in order to facilitate the understanding of the main concepts used in this research. In chapter 4 is provided an overview on the state-of-the-art of software process self-assessment methods and usability process assessment models. Chapter 5 presents the development of UPCASE, a method for self-assessing the capability of usability process in small organizations. In Chapter 6 is presented the application and the evaluation of UPCASE, through the realization of expert panels, case studies in 4 software projects and a survey. The discussion and final considerations are presented in Chapter 7.

2 METODHOLOGY

The methodological procedures of this research are defined based on the proposal of Saunders (2009), which classifies the scientific method in layers. In terms of philosophy, this research is predominantly interpretative, as it is assumed that the research object (self-assessment of usability processes) is interpreted and evaluated from the social point of view by the actors involved in the research (process assessors). Regarding the research approach, an inductive approach is used, since it is not part of a pre-established hypothesis, but rather seeks to reach the solution of the problem from the inferences from case studies of the object of study. As research strategy, multi-methods are adopted, since different qualitative and quantitative methods are used, such as, bibliographic research, systematic mapping (KITCHENHAM et al., 2010), case study (YIN, 2009), GQM (Goal/Question/Metric) (BASILI; CALDIERA; ROMBACH, 1994), software process capability/maturity (SPCMM) Customization Methodology (HAUCK, 2011), guidelines for designing questionnaires (KASUNIC, 2005) and statistics analysis (DEVELLIS, 2016). From the point of view of its nature, this is an applied research, as it aims to generate knowledge for practical application, contributing to the solution of specific problems that occur in reality (MARCONI; LAKATOS, 2010). Regarding the research objectives, it is classified as exploratory, as it is interested in providing a greater understanding of the phenomenon that is investigated. The research stages, methods used, and the expected results are summarized in Figure 1.

Figure 1 - Research methodology

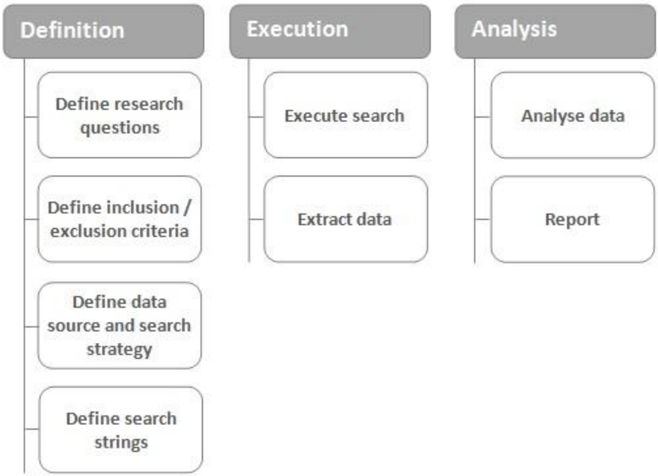


Source: developed by the author

1 Analyze theoretical foundation. In the first stage of the research the theoretical foundation is defined conducting a bibliographical research on the main subjects used throughout this work. This includes a theoretical revision on usability engineering, usability, software processes assessment; and software and usability assessment models.

2 Synthesize state of the art. In this step, two systematic mappings are carried out, the first aims at identifying researches that developed usability maturity/capability models. The second review aims at identifying researches that developed software processes self-assessment methods. Both systematic mappings are carried out following the procedure proposed by Kitchenham et al. (2010), as presented in Figure 2.

Figure 2 - Research methodology (based on (KITCHENHAM et al., 2010; PETERSEN et al., 2008)



In the definition phase, the research questions, the research repositories and the review protocol were defined. The protocol contains the selection criteria, to determine which studies will be included in the review, the data sources, search strategy and the definition of search strings.

The execution phase was carried out based on the review protocol and consisted in conducting the search in the selected repositories. The initial search results were analyzed with respect to their relevancy applying the inclusion/exclusion and quality criteria. Once identified the relevant studies, the data needed to answer the search questions was extracted. The extracted data are analyzed with respect to the defined research questions and the results are interpreted during the analysis phase.

3 Develop a usability process self-assessment method. In this step the self-assessment method for assessing usability process is developed in accordance with ISO/IEC 29110 (ISO/IEC, 2016a). The development of the assessment method is carried out in three stages: development of the reference model, development of the measurement framework and development of the assessment process.

3.1 Develop the reference model. At this stage the process reference model is developed, considering the state of the art (Step 1). The development of the model follows the methodology for customizing SPCMM proposed by Hauck (2011):

- Knowledge identification. The main objective of this step is to obtain a familiarization with the domain and to characterize the context for which the assessment method will be customized, defining its scope and objectives.
- Knowledge specification. In this step, a first version of process reference model is developed.
- Knowledge refinement. During this step the process reference model is validated by evaluating and consolidating the draft model.

3.2 Develop the measurement framework. The measurement framework defines an ordinal scale for the assessment of the process capability. Its development follows the procedures of the questionnaire design guide proposed by Kasunic (2005):

- Generate a set of items. Using GQM, the process reference model is systematically decomposed into questionnaire items, relating the data collection instrument to a factor to be measured in alignment with the reference model.
- Determine the responses format. The response format for the data collection instrument items is defined.
- Get review of items by experts. Aiming at analyzing the face validity of the instrument of data collection from the point of view of specialists is performed an expert panel (BEECHAM et al., 2005).

3.3 Develop the assessment process. The development of the assessment process has as input the steps of the process of empirical studies proposed by Wohlin (2016). The process is modeled in a prescriptive way, which defines how the assessment should be executed (ACUÑA et al., 2000). The modeling of the assessment process is performed using the Business Process Modeling Notation (BPMN) (WESKE, 2012) that represents the process in a standardized way.

3.4 Develop the assessment tool. In this step, a software tool is developed that allows the realization of an online self-assessment of the capability of the usability process. The development of the assessment tool was carried out following a waterfall development process, composed of steps (PRESSMAN, 2005):

- Requirements analysis: in this stage the functional and non-functional requirements of the tool are defined, as well as each use case is described.
- Design: in accordance to the identified requirements the tool to be developed is modeled, including the model of its architecture.
- Implementation: the tool is constructed as a web-application.
- Test: throughout its development the tool is being tested on different levels, concluding with system tests to ensure that all functional requirements have been met.

4 Apply and evaluate self-assessment method. In this stage the application and evaluation of the UPCASE method is performed, aiming at evaluating its quality in terms of reliability, usability, comprehensibility and internal consistency. The method is evaluated by defining specific criteria, using the Goal Question Metric (GQM) technique (BASILI; CALDIERA; ROMBACH, 1994). GQM, a goal-oriented measurement approach, assists in the definition and analysis of measurement, through the definition of goals and in the unfolding of these goals into operationally collectible measures. The quality characteristics evaluated are:

Reliability, usability and comprehensibility: Reliability is the overall consistency of a measure, that is, if the same measuring process provides the same results. A measure is said to have a high reliability if it produces similar results under consistent conditions (TROCHIM, 2006). Usability is the extent to which a product can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a specific context of use (ISO/IEC, 1998) and Comprehensibility is the extent to which a text as a whole is easy to understand (ISSCO, 2016). To evaluate these characteristics we performed a series of **case studies**, as proposed by (WOHLIN et al., 2016), in which SE applied the UPCASE method in order to assess their usability process and their assessment data is collected. Two types of cases studies are performed (revised and unrevised). In the unobserved case studies the assessment has been performed unobserved and data has been collected via the online tool. To perform this case studies, the proposed method was made available online, in English and Portuguese versions and was spread on email lists of small software organization associations, small organizations and usability groups. The observation of the use of UPCASE in practice allows to draw conclusions about its usability, comprehensibility (DAVIS, 1989; MATOOK; INDULSKA, 2009; RITTGEN, 2010; SINISCALCO; AURIAT, 2005). In the revised case studies, besides the unobserved assessment, an independent assessment has been performed by the author of this work in order to compare the assessment results, in which a researcher observed the UPCASE method being used by small organizations to self-assess the capability of their own usability process. The results of the self-assessments and the researcher's assessment are used in a comparative analysis to verify the reliability of UPCASE's assessment process.

For the reliability analysis, a concordance analysis was performed using the Weighted Kappa coefficient (COHEN, 1960), as well as an analysis of the intraclass correlation coefficient (FISHER, 1925). The intraclass correlation coefficient is often applied for assessing the consistency or reproducibility of quantitative measurements made by different observers measuring the same quantity (FISHER, 1925). However, considering only the results of the evaluations, this coefficient does not allow to evaluate agreement. Weighted Kappa coefficient, on the other hand, is a widely used index for assessing agreement between raters. When the rated categories are ordered or ranked (like UPCASE questionnaire “0-Not achieved, 1-Parttilay achieved 2-Fully achieved”), then a weighted kappa coefficient is computed that takes into consideration the different levels of disagreement between categories.

Internal consistency: Internal consistency is related to the degree in which a set of items are measuring a single quality factor, i.e., the capability of a usability process. (CRONBACH, 1951). The internal consistency of the assessment method questionnaire was evaluated through the application of statistic methods in data collected via the case studies. The answers obtained in the case studies were statistically analyzed using Cronbach's alpha method (CRONBACH, 1951).

3 THEORETICAL FOUNDATION

In this chapter the main concepts used throughout this dissertation are presented in order to provide a better understanding about the research. Concepts related to software processes improvement are presented, such as: software process, process assessment and process assessment model. In addition, concepts regarding to the usability process, such as, usability and usability standards are addressed.

3.1 SMALL ORGANIZATIONS

Small organizations are commonly defined in Brazil either by considering the number of professionals of the organization or considering the annual revenues. A service/commerce sector organization is considered small if it has between 10 and 49 employees, according to Table 1; or if it has gross annual sales between R$ 360,000.01 and R$ 3,600,000.00.

Table 1 - Criterion of classifying the size of companies by number of employees. (Source: (SEBRAE, 2014))

Size	Services and trade
Micro organization	Up to 9 employees
Small organization	From 10 to 49 employees
Medium organization	From 50 to 99 employees
Large organization	Over 100 employees

In this work, we characterize "small organizations" (SE) by their number of employees, having up to 49 employees. In addition, our definition of small software organization includes software development departments that have the same characteristics as SE (SEBRAE, 2014).

Over the past 30 years, SEs (along with micro-enterprises) have acquired great relevance in the national economy. Micro and small organizations generated, in 2011, 27% of the value added of the set of activities surveyed (GDP); and the Services and Commerce sector represented 19% of the value added (ABES, 2017). In addition, micro and small enterprises, in the sector of services and commerce, accounted for 98% of the total number of organizations formalized; 44% of formal service jobs, and approximately 70% of the jobs generated in commerce (ABES, 2017).

In addition to the small number of employees, SEs have other characteristics that make their needs and way of working unique. SEs often do not have enough staff to develop specialized functions that would enable them to perform complex tasks or develop secondary products (ANACLETO, 2004; MISHRA; MISHRA, 2009). Also, in general these organizations do not use formal processes and, therefore, may have many problems to complete their projects under time and cost constraints (SÁNCHEZ-GORDÓN; O'CONNOR, 2016). The limitation of financial resources presents many consequences, such as hinders the hiring of specialists and the training of employees; and difficult the execution of processes improvement programs, which typically require a considerable amount of time and money (ANACLETO, 2004; MIRNA et al., 2012; SÁNCHEZ-GORDÓN; O'CONNOR, 2016; SULAYMAN et al., 2012).

These particular characteristics of SEs directly impact on how they assess and improve their software processes. As a consequence of the limitation of resources, SEs main motivation to implement SPI initiatives is not to obtain a certificate, but rather to make the organization's process more efficient and effective (GARCIA; PACHECO; CRUZ, 2010; MISHRA; MISHRA, 2009). Thus, they may not need to use costly assessment methods that demand audits, as they may require only an assistance in the identification of the process status and improvement opportunities (MIRNA et al., 2012); and thus, assisting in the beginning of an improvement program. On the other hand, the use of well-established and recognized standard-based assessment methods is an interesting alternative, considering that the extent to which an organization is getting more mature, it may desire a certification, and in this case, it would already meet some of the necessary requirements (ANACLETO et al., 2004; MISHRA; MISHRA, 2009; PINO; GARCIA; PIATTINI, 2008).

The lack of financial resources (and consequently of time) makes it necessary for SEs to use light assessment methods that can be used with the resources (human and non-human) that are available, so that the assessment can be carried out within a reasonable time with low cost, preferably being publicly available (ANACLETO, 2004; MIRNA et al., 2012; MISHRA; MISHRA, 2009; PINO et al., 2010). One way to reduce the assessment time is using a single tool that contains all the information necessary to carry out the assessment, and that (semi-)automates and/or supports assessment steps (ANACLETO et al., 2004; MIRNA et al., 2012; SULAYMAN et al., 2012).

The lack of specialized professionals in SEs makes it necessary to use tools that can be used by non-experts in SPI or in the domain being assessed (ANACLETO et al., 2004; MISHRA; MISHRA, 2009). So, it is important that the assessment method provides accesses to a detailed definition of the process reference model and the assessment model, with descriptions of process purpose, process outcomes provided by the PRM, capability levels and process attributes (ISO/IEC, 2016a). In this context is also important that the process assessment adequately guides the activities that need to be performed in the process assessment. It should provide a clear definition of roles, and their responsibilities, and a detailed description of the assessment process, with simple recommendations. This also applies in relation to the usability process, SEs generally present a low level of maturity, including little knowledge about relevant usability standards and concepts, not recognizing the benefits that the proper implementation of a usability process can provide (ANACLETO et al., 2004) (FUCHS; RITZ; STRAUCH, 2012; HERING et al., 2015; O'CONNOR, 2009; RENZI et al., 2015). These difficulties have motivated researchers and practitioners to investigate and design new assessment methods to address the needs of SE (ALEXANDRE; RENAULT; HABRA, 2006; ANACLETO et al., 2004; CIGNONI, 1999; KUVAJA; PALO; BICEGO, 1999; ROUT et al., 2000).

Another reason that makes process assessment less attractive to SEs is the difficulty of understanding and implementing them in practice (KALPANA; JEYAKUMAR, 2010; YUCALAR; ERDOGAN, 2009). This fact leads many organizations to seek even more for simplicity of processes and as result they are increasingly attracted to agile methods. Considering this, those agile approaches must be incorporated into any potential process assessment method (MC CAFFERY, 2007).

Despite the challenges that most companies have to assess their processes, this activity cannot be ruled out since it allows identifying organization's strengths and weaknesses and, thus, being a first step towards improving the software development process (HABRA et al., 2008; KAR et al., 2012). Thus, the motivation for this kind of organization should, therefore, be to improve their processes in order to make them more effective and efficient units, rather than to gain the certification itself. These strategies must be aligned with the widely recognized standards in order to enable SEs to establish a solid basis for improvement which will lead them towards certification in these standards in the future (MISHRA; MISHRA, 2009; PINO; GARCIA; PIATTINI, 2008; VON WANGENHEIM; C. ANACLETO, 2006).

3.2 SOFTWARE PROCESS

The use of appropriate SW processes directly contributes to the development of high quality software products (ISO/IEC, 2003b). Process can be defined as 'a system of operations to produce something, a series of actions, changes or functions to achieve a purpose or an outcome' (TEAM, 2010), or a sequence of steps performed for a given purpose (ISO/IEC, 1990). SW process usually specifies the actors who perform the activities, roles, and artifacts produced (ACUÑA et al., 2000). The basic elements of an SW process are:

• An activity is an atomic operation, or a step of a process. Activities aim to generate or modify a set of artifacts.

• The set of artifacts that must be developed, delivered and maintained in the project is the product.

• A resource is a necessary asset for the activity to be performed. There are two main features for SW development: executors and tools.

• Tools are usually connected with the activities in which they are used, and the implementers are connected to their roles.

The use of appropriate SW processes is important for a number of reasons: "Facilitates human understanding, communication and coordination, assists in the management of SW projects, measures and enhances the quality of SW products efficiently and provides support for the improvement process" (BOURQUE; FAIRLEY; OTHERS, 2014). As a consequence, it provides greater stability, control, and organization for activities that, if not controlled, may become chaotic (PRESSMAN, 2005).

3.3 USABILITY PROCESS

Usability is one of the characteristics that comprise the software quality model (ISO/IEC, 2010). The quality model determines which characteristics should be considered in the assessment of a software product. The quality of a system is the degree to which the system meets the needs of stakeholders, these needs may be classified in 8 characteristics, presented in Figure 3.

Figure 3 - System/software product quality (Source: adapted from (ISO/IEC, 2010))

Usability does not have yet a homogenous definition among researchers and standardization bodies (Table 2), however, they agree that usability refers to a set of concepts such as performance, user satisfaction or ease of learning.

Table 2 - Definitions of usability

Author	**Definition**
ISO/IEC 9241-11 (1998)	Usability is the extent to which a product can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a specific context of use.
Shackel (1984)	The capability in human functional terms to be used easily (to a specified level of subjective assessment) and effectively (to a specified level of performance) by the specified range of users, given specified training and user support, to fulfill the specified range of tasks, within the specified range of environmental scenarios.
Nielsen (1994)	Is a quality attribute that assess how easy user interfaces are to use. It also refers to methods for improving ease-of-use during the design process.

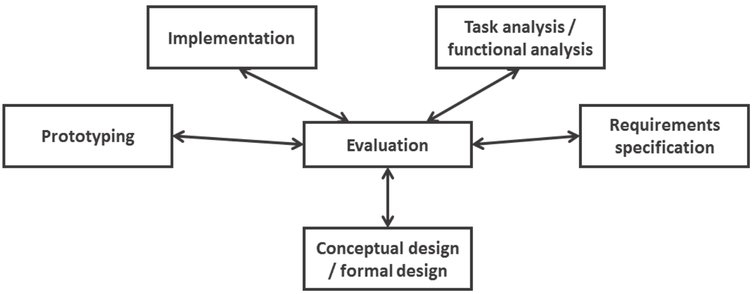
Shackel (1984) and Nielsen (1994) define usability as easy to use and learn, but excludes utility. On the other hand, the definition of ISO/IEC 9241-11 (1998) is more comprehensive, including not only the utility but also the reliability. In the context of this work, the definition of usability proposed by ISO/IEC 9241-11 (1998) is adopted as it provides the most comprehensive concept, being the result of a collaborative work among several researchers and being the standard considered to have the greatest impact (BEVAN, 2001). This standard characterizes usability with: efficacy, efficiency and satisfaction. **Efficacy** is the accuracy and completeness with which users achieve interaction and is generally evaluated in terms of completeness of a task, and the quality of the result obtained. **Efficiency** refers to the amount of effort to reach a certain goal. The deviations that the user makes, the duration of the interaction and the amount of errors made, can be used to evaluate the level of efficiency of the software. **Satisfaction**, refers to the level of comfort that the user feels when using the interface (ISO/IEC, 1998).

Usability engineering (UE) is the application of systematic, quantifiable methods to the development of interactive software systems to achieve high quality in use (METZKER; REITERER, 2002). It is generally concerned with human-computer interaction and specifically with the development of human-computer interfaces that have high usability or ease of use. UE provides structured methods for achieving efficiency and elegance in interface design (MAYHEW, 1999), adapting the general components of software engineering to provide an engineering process to develop products with interfaces that have good usability.

These process, denominated here as **usability process,** serve as a model for developing high usability interfaces, such as the usability engineering cycle proposed by Mayhew (1999), Star Life Cycle (HIX; HARTSON, 1993), KESSU (JOKELA, 2004a) and ISO/IEC 9241-210 (2001). These process models have in common the fact that they are user-centered.

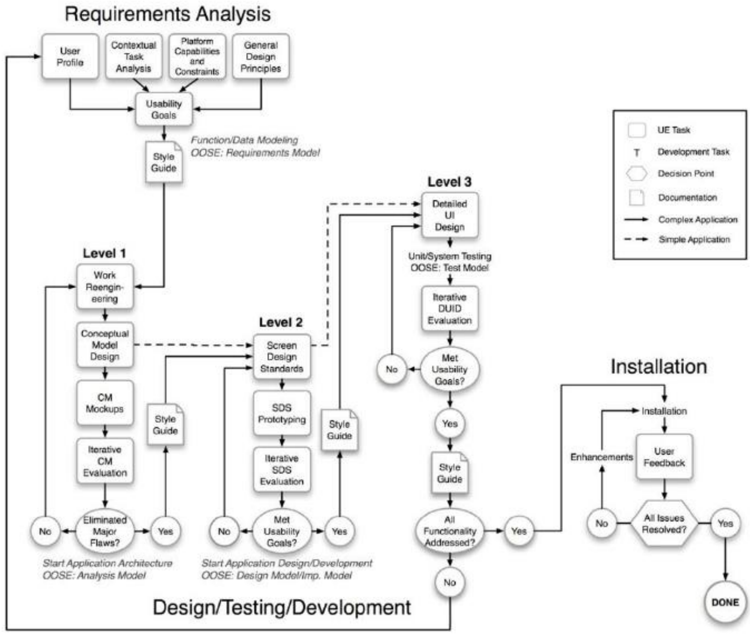
User-centered design (UCD) is an interactive system development approach that aims to develop software with high usability focusing on the user, their needs and requirements, and applying techniques and knowledge on ergonomics and usability (ISO/IEC, 2001b). Throughout this work the expression "usability process" is used to refer to all the processes of interface development whose goal is to produce interfaces with high usability. The Star Model, proposed by Hix and Hartson (1993), focuses on the evaluation of usability as the center of the activities of the development process, Figure 4. Around this main activity are implementation, task analysis/functional analysis, requirements specification, prototyping and conceptual design.

Figure 4 - Star model (Source: Hix and Hartson (1993))



Usability Engineering lifecycle, proposed by Mayhew (1999), is an attempt to redesign the process of software development around the knowledge, methods and activities of the usability engineering (Figure 5). The process begins with a structured analysis of questions related to usability, such as the user profile, the tasks and the context of use. The information raised in this step serves as input to define the usability goals. In the second stage an iteration of the usability engineering methods occurs. This includes the conceptual model development, mock-ups development, prototypes and usability testing. This iteration ends when the usability goals that have been defined are met. In the final step, the system is installed. It is important to consider some difficulties in relation to this model, such as the change of the entire development process.

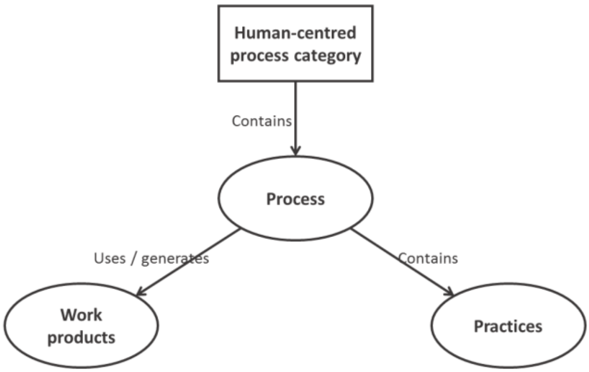
Figure 5 - Usability engineering lifecycle (Source: Mayhew (1999))



ISO/IEC TR 18529 (ISO/IEC, 2000) is a technical report which intends to assist those who wish to make their system development process more human-centred. It presents a definition of a usability processes and lists their components, outcomes and the information used and produced them. The usability process model presented in this Technical Report is a formalized definition of the processes described in ISO/IEC 13407 (ISO/IEC, 1999), in order to make it accessible to process assessment.

The usability process is described in conformance with ISO/IEC TR 15504 (ISO/IEC, 2004). The primary use of this series is for measuring how well an organization carries out the usability processes, however, it can also be used as a description of what is required in order to design products with an assured degree of usability. The components of the process model presented in (ISO/IEC, 2004) are shown in Figure 6.

Figure 6 - Entity relationship of the process reference model (Source: (ISO/IEC, 2000))



The usability process reference model consists of seven sub-processes (ISO/IEC, 2004), as shown in Figure 7. Each sub-process contains practices that describe what has to be done in order to include the users in the lifecycle as shown in Figure 7.

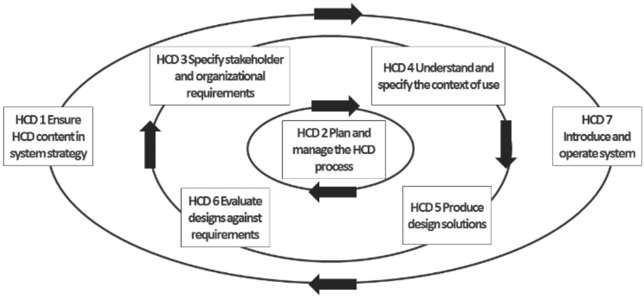
Figure 7 - Human centred design process (Source: (ISO/IEC, 2000))

Human-centred system development

HCD 1	HCD 2	HCD 3	HCD 4	HCD 5	HCD 6	HCD 7
Ensure HCD content in systems strategy	**Plan and manage the HCD process**	**Specify stakeholder and organisational requirements**	**Understand and specify the context of use**	**Produce design solutions**	**Evaluate designs against requirements**	**Introduce and operate the system**
represent stakeholders collect market intelligence define and plan system strategy collect market feedback analyse user trends	consult stakeholders plan user involvement select human-centred methods ensure a human-centred approach plan HCD activities manage HC activities champion HC approach support HCD	clarify system goals analyse stakeholders assess H&S risk define system generate requirements set quality in use objectives	identify user's tasks identify user attributes identify organisational environment identify technical environment identify physical environment	allocate functions produce task model explore system design develop design solutions specify system and use develop prototypes develop user training develop user support	specify context of evaluation evaluate for requirements evaluate to improve design evaluate against system requirements evaluate against required practice evaluate in use	manage change determine impact customisation and local design deliver user training support users conformance to ergonomic legislation

Figure 8 presents the cyclical nature of the usability sub-processes. In general, sub-processes 3-6 are more technical and may be performed several times during the software development. Sub-process 2 covers management and control of human-centred activities. It uses information generated by the sub-processes 3-6 and connects the human-centred lifecycle to other processes in software development. Sub-process 1 connects the human-centred lifecycle to higher management processes and goals for projects which process 3- 6 are implemented. HCD 7 is concerned with the use of the system, it connects the other processes to the support phase of the system lifecycle.

Figure 8 - Linking human-centred sub-process in the lifecycle (Source: (ISO/IEC, 2000))



3.4 PROCESS IMPROVEMENT AND ASSESSMENT

Regardless of the goal of the organization, in general, all of them want to develop quality products that satisfy the customers, increase their market share and thereby increase their profit. One of the best ways for an organization to achieve success with its products is by improving the processes used throughout the product development. Process improvement is an action taken to change an organization's processes so that they follow the organization's business needs and achieve business goals more effectively (ISO/IEC, 2004).

In any process improvement program, it is important to identify the specific problems and opportunities for improvement. Identifying the organization's real problems and improvement opportunities that effectively bring benefits is essential to the success of the improvement program (ANACLETO, 2004). In this way, process assessment is among the first activities performed when an organization starts to improve its process (ISO/IEC, 2003c). Process assessment is an evaluation of an organization's processes against a process reference model. The assessment result is an indicator of how well the organization's processes meet the requirements of the process reference model (ISO/IEC, 2004). In addition to starting the process improvement program, process assessment can be performed at other moments during the SPI, such as during monitoring and at the end, to compare with the outcome of the assessment made before the improvement or even to achieve a Certification. Process assessment can be carried out using different approaches, through a self-assessment (performed by the organization itself being assessed), or through an independent assessment (performed by assessor extern to the organization being assessed).

The processes of an organization may be assessed in terms of its capability or in terms of its maturity. A process is capable if it satisfies product quality specifications and process performance objectives (TEAM, 2010). A capable process consequently produces outputs according to specifications. According to CMMI (2010), the capability of a process "describes a range of expected outcomes that can be achieved by following a software process." The software process capability of an organization provides a means of predicting the most likely outcomes that can be expected from the next software project with which the organization commits itself. On the other hand, the maturity of a process means what the organization is doing, it is documenting adequately, everyone knows what they are expected to do, and they do their activities accordingly. It is "the degree to which a specific process is defined, managed, controlled, measured, and effective." Maturity implies a potential for capability growth and indicates both the wealth of an organization's software process and the consistency with which it is applied in projects". Capability assessment targets a process on an individual basis and represents the current state of process execution. The process maturity assessment, however, provides the current state of an organization in relation to how it controls a set of processes (TEAM, 2010).

As the objective of this work is to develop a method of assessing the capability of usability process, the focus of this dissertation is on process capability assessment, rather than maturity assessment.

In process capability assessment, the process used by the organization is examined in order to determine if it is effective in achieving the expected results and outcomes. The assessment characterizes the current practice of an organization in terms of the capability of certain processes (ISO/IEC, 2004). An assessment allows a greater understanding of the current process, the problems and outputs generated, as well as exploring software's best practices (ISLAM; ZHOU, 2011). In addition, assessing the processes benefits the organization by encouraging "a culture of continuous improvement (TATE, 2003). The method for process assessment may be qualitative or quantitative. Qualitative assessments are based on the judgment of a specialist, and quantitative assessments assign a numerical score to the process based on an analysis of objective evidence that indicates that the goals and results of the process are achieved (BOURQUE; FAIRLEY; OTHERS, 2014). A typical process assessment method includes planning, inquiry (collecting evidence through questionnaires, interviews and observations of work practices), collection and validation of data, analysis and results. The results should go beyond simply showing the result of a checklist but should provide a basis for the improvement process.

The quality of the assessment results depends on the assessment method used, the integrity of the quality of the data obtained and the objectivity and capability of the assessment team (BOURQUE; FAIRLEY; OTHERS, 2014). Formal (or external) assessment processes are well documented and provide guidance on a process assessment model for process assessors. Assessors (typically external) should have competencies that include education, training and experience (ISO/IEC, 2003d) to ensure that assessments are comparable across the world. In the context of process improvement, formal assessment may not offer an adequate cost-benefit ratio. However, most of its principles can also be applied in a self-assessment (VARKOI, 2009).

3.4.1 Process Self-Assessment

Self-assessment is performed by an organization to assess the capability of its own process. The assessor of a self-assessment is usually a member of the organization, as well as the people on the assessment team (ISO/IEC, 2003b). In a self-assessment it is advisable for the assessor to have experience with the processes and the process model applied. A convenient way to conduct an assessment is to interview staff and review related documentation. Respondents are usually responsible for the process and the people who actually implement it. The assessment result is usually reserved for internal use in order to support process improvement (VARKOI, 2009). Carrying out a self-assessment can bring many benefits when compared to other methods such as:

• Does not significantly impact the company's work routine, as it does not require the organization to accommodate an external assessment team. There is no need for documentation as evidence of conformity to a reference model (VARKOI, 2009).

• Usually, it tends to be less expensive than external assessment, for being less formal (VARKOI, 2009).

• The self-assessment process is effective as compels business managers to examine their own activities and develop their own plans within their areas of expertise (VARKOI, 2009).

• It can be a quick and inexpensive way to assess a specific process before executing more formal process assessment (VARKOI, 2009).

While self-assessment offers many benefits, it also has some shortcomings. For example, people tend to assign a better rating than an external assessor would do. This can happen as people are not very familiar with process improvement or maturity and capability models, because they misinterpret the issues, or because they give the answer they believe to be the one the boss would like to receive.

3.5 PROCESS ASSESSMENT MODELS

In order to perform a process assessment, typically Process Assessment Models (PAM) are used. They describe the life cycle processes and the process management principles (TEAM, 2010). There exists several PAMs, the most widely used of them currently are the ones related to CMMI, as for example, CMMI-DEV (TEAM, 2010), CMMI-SRV (TEAM, 2010) and the ones that conform to ISO/IEC 15504, as for example, ISO/IEC 15504-5 (ISO/IEC, 2005) and Automotive SPICE (SIG AUTOMOTIVE, 2010).

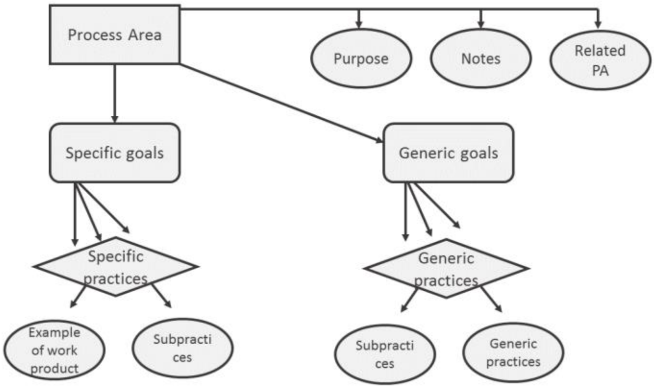
3.5.1 CMMI

Capability Maturity Model Integration (CMMI) models are collections of best practices that help organizations to improve their processes. These models are developed by product teams with members from industry, government, and the Software Engineering Institute (SEI). The basic structure of these models is provided by CMMI Framework, which organizes CMMI components and combines them into CMMI constellations and models. A constellation is a collection of CMMI components that are used to construct models, training materials, and appraisal related documents for an area of interest (e.g., development, acquisition, services).

The CMMI Framework contains all the goals and practices that are used to produce CMMI models, which belong to CMMI constellations. To allow the use of multiple models within the CMMI Framework, the components of the models are classified as either common to all CMMI models or applicable to a specific kind of product or service.

The components of the model are grouped into three categories: required, expected, and informative, as shown in Figure 9.

Figure 9 - CMMI model components (Source: (TEAM, 2010))



The required components of a CMMI model are:

Process area: a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making improvement in that area. All CMMI models contain 16 core process areas. A core process area is a process area that is common to all CMMI model.

Specific Goals: describes the unique characteristics that must be present to satisfy the process area. A specific goal is a required model component and is used in appraisals to help determine whether a process area is satisfied.

Generic Goals: these goals are called “generic” because the same goal statement applies to multiple process areas. A generic goal describes the characteristics that must be present to institutionalize processes that implement a process area. A generic goal is used in appraisals to determine whether a process area is satisfied.

CMMI supports two improvement approaches using levels. One approach enables organizations to incrementally improve processes corresponding to an individual process area (or group of process areas) selected by the organization. The other approach enables organizations to improve a set of related processes by incrementally addressing successive sets of process areas.

These two improvement approaches are associated with the two types of levels: capability levels and maturity levels. These levels correspond to two approaches to process improvement called “representations”. The two representations are called “continuous” and “staged”. Using the continuous representation enables to achieve “capability levels”. Using the staged representation enables to achieve “maturity levels.” Table 3 illustrates the structures of the continuous and staged representations. The staged representation uses maturity levels to characterize the overall state of the organization’s processes relative to the model as a whole, whereas the continuous representation uses capability levels to characterize the state of the organization’s processes relative to an individual process area.

Table 3 - Comparison on capability and maturity levels

Level	**Continuous Representation Capability levels**	**Staged Representation Maturity Levels**
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4		Quantitatively Managed
Level 5		Optimizing

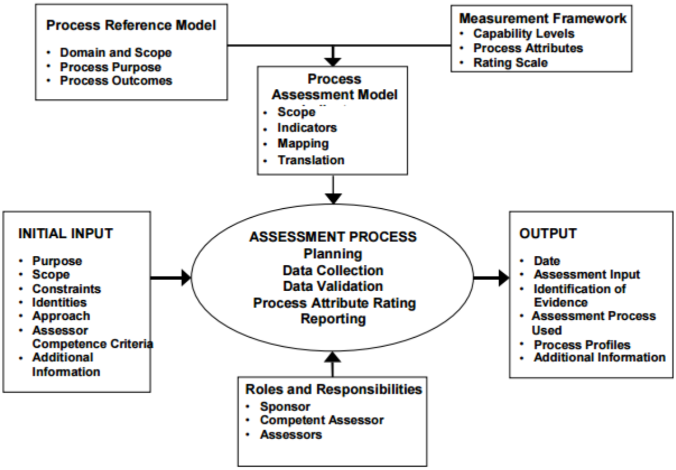
3.5.2 ISO/IEC 15504 - Process assessment

ISO/IEC 15504 is a series that presents a model for performing process assessment in software organizations. It is composed of the following parts:

- Part 1 - provides a general introduction to the concepts of process assessment.
- Part 2 - defines the minimum requirements for a coherent assessment, which allows for repeatability and provides evidence to substantiate classifications and verify compliance with requirements.
- Part 3 - provides guidance for interpreting the requirements for conducting an assessment.
- Part 4 - provides guidance for the use of process improvement and process capability determination.
- Part 5 - contains an exemplary process assessment model that is based on the Process Reference Model defined in ISO/IEC 12207.

ISO/IEC 15504 summarizes the context of a process assessment and presents the minimum requirements for the performance of a process assessment. During the process assessment a defined assessment process is followed, which contains roles and responsibilities, receives initial inputs and generates certain outputs. The assessment process is based on an assessment model, which also contains a process reference model and a measurement structure. Figure 10 presents the relation of these components.

Figure 10 - Components of the assessment framework (Source: (ISO/IEC, 2003b))



3.5.2.1 Process Reference Model

In an organization, the processes of different projects tend to follow the same standard process as the best practices are informally recognized, or because the community has elevated them to the status of standards. Therefore, it makes sense to capture these similarities in a process representation, which describes these common characteristics and promotes the cultural homogeneity of the community (DERNIAME; KABA; WASTELL, 2006). These standardized practices may be represented as a process reference model. These models try to capture the main characteristics of a set of activities necessary to develop a SW product (ACUÑA et al., 2000). They contain the definitions of the life cycle processes in terms of their purpose and results, together with an architecture describing their relationship between the processes (ISO/IEC, 2004). The **purpose** describes the high-level goals that the process must achieve. The **results** are the expected outcomes of the successful execution of the process. They can be the production of an artifact, a significant change of status, or the fulfillment of specific restrictions or requirements (ISO/IEC, 2003d). There are several reference models for SW development. Software organizations focus on the use of these models in order to ensure that during the software development a certain process is followed (PINO; GARCIA; PIATTINI, 2008). As a consequence, it is expected that a higher-quality software product is obtained, increasing productivity and cost efficiency (APRIL; ABRAN; DUMKE, 2004). Examples of software process reference models are ISO/IEC 12207 (ISO/IEC, 2008b) and for hardware/software systems processes to ISO/IEC 15288 (ISO/IEC, 2015), 15504-6 (ISO/IEC, 2013) and MPS.BR (SOFTEX, 2011).

3.5.2.2 Measurement framework

The measurement framework provides a schema for use in characterizing the capability of an implemented process regarding to a Process Assessment Model. The measure of the capability is based on a set of process attributes (PA). Each PA defines an aspect of the process capability. The degree in which a process capability is achieved is represented on a rating scale defined using an ordinal scale of measurement as for example:

- **N** - Not achieved: There is little or no evidence of achievement of the defined attribute in the assessed process.
- **P** - Partially achieved: There is some evidence of an approach to a significant achievement of the defined attribute in the assessed process.
- **L** - Largely achieved: There is evidence of a systematic approach to, and significant achievement of, the defined attribute in the assessed process.

- **F** - Fully achieved: There is evidence of a complete and systematic approach to, and full achievement of the defined attribute in the assessed process.

The combination of the PA ratings defines the process capability level, as shown in Figure 11.

Figure 11 - Example of measurement framework (Source: (ISO/IEC, 2004))

Process Attribute ID	Capability Levels and Process Attributes
	Level 0: Incomplete process
	Level 1: Performed process
PA 1.1	Process performance
	Level 2: Managed process
PA 2.1	Performance management
PA 2.2	Work product management
	Level 3: Established process
PA 3.1	Process definition
PA 3.2	Process deployment
	Level 4: Predictable process
PA 4.1	Process measurement
PA 4.2	Process control
	Level 5: Optimizing process
PA 5.1	Process innovation
PA 5.2	Continuous optimization

The capability dimension presents capability levels, process attributes and a measurement scale. The capability levels are used to measure the capability of a process (ISO/IEC, 2004). ISO/IEC 15504 defines capability levels on a six-point ordinal scale with each level of capability associated with an attribute of the process. The attributes of the process are "measurable characteristics of the capability of a process". The levels of the measurement scale and their respective attributes are shown in Table 4.

Table 4 - Capability levels x process attributes (Source: (ISO/IEC, 2005))

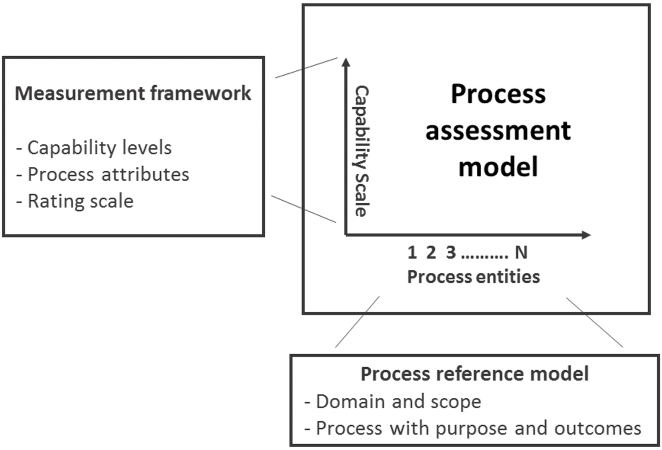
Capability level	**Process attribute**
Level 0 – Incomplete	The process is not implemented or fails to achieve its purpose.
Level 1 – Performed	The process is implemented and achieves its purpose.

Level 2 – Managed	The process, as the generated product, is managed.
Level 3 – Established	The process, in addition to being executed and managed, is defined and documented.
Level 4 – Predictable	The established process operates within defined limits to achieve the expected outputs.
Level 5 – Optimizing	The process is in constant optimization and undergoes innovations and continuous improvements.

3.5.2.3 Process assessment model

The Process Assessment Model (PAM) forms the basis for collecting evidence and rating the process capability. A PAM provides a two-dimensional view of process capability. The Process dimension describes a set of process that relate to a Process Reference Model. The Capability dimension describes the capabilities related to the Capability levels and the process attributes. The relationship among both dimensions is presented in Figure 12.

Figure 12 - Process assessment model dimensions (Source: (ISO/IEC, 2003b))



3.5.2.4 Assessment process

The assessment process defines the activities required to carry out the assessment process. The assessment process contains at least five specified phases:

- **Planning:** in this phase the assessment inputs are defined, such as the activities to be carried out during the assessment, the resources and schedule of the activities, identification of the responsibilities of the participants, criterion to verify if the requirements of the norm were fulfilled and description of the outputs of the assessment.
- **Data collection:** should be done systematically. The technique for data collection should be defined; and the data collection and analysis must be explicitly identified and demonstrated, there must be a correspondence between the process units and the elements of the Process Assessment Model. Each identified process should be assessed objectively, and the evidence should be recorded to serve as the basis for the classification of the process.

• **Data validation:** it should confirm that the collected evidence is objective, ensure that the evidence is sufficient for the scope of the assessment and that the data are consistent.

• **Classification of process attributes:** the rating should be assigned based on valid data regarding the process assessed. The set of process attribute ratings should be documented, the set of indicators used in the assessment should be used to assist in the judgment of the process attributes ratings, the decision process used to make the judgment should also be recorded.

• **Reporting:** assessment results should be documented and reported to the assessment funder.

The assessment process must be documented; in addition, the assessors must record the indicators of the performance or capability, used to justify the ratings.

The Assessment Process contains Roles and Responsibilities, Inputs and Outputs.

Input: The assessment inputs are the information required before a process assessment can star. They define the assessment sponsor, scope and limitations. The scope includes: which process will be assessed, the highest level of capability that will be assessed for each process and the department of the organization that will implement the assessment. The limitations of the assessment consider, for example, the availability of resources and the amount of time available to perform it.

Output: The output of the assessment should be compiled into a document that contain at least the date of the assessment, the inputs, the identification of the evidence, the assessed process and the set of the process profile and, optionally, a Capability level for each assessed process.

The process assessment is carried out by a team with at least one competent assessor who has the necessary competencies.

Roles and responsibilities: The assessment sponsor should verify that the person responsible for carrying out the assessment is competent, ensuring that the necessary resources are available. The assessor should ensure that the assessment will take place in accordance with ISO/IEC 15504 and that the participants are aware of the objective and scope of the assessment.

3.6 PROCESS ASSESSMENT METHOD

In the context of this research, it is understood that an “assessment method” is the set of all the elements necessary to carry out a process assessment, that is, the same elements called “framework” by ISO/IEC 15504. There are several assessment methods, such as MARES, an assessment method for small software companies conformant with 15504 (ANACLETO, 2004), SCAMPI is an appraisal method to provide benchmark-quality ratings relative to CMMI models (TEAM, 2010), ISO/IEC 15504-5 a software life cycle process assessment model (ISO/IEC, 2005) and ISO/IEC 29110-3 an assessment guide for very small entities (ISO/IEC, 2016b). As ISO/IEC TR 29110-3 is specific for SEs and most assessment methods are based on international standards (LACERDA; VON WANGENHEIM, 2017), it was considered more appropriate to support this work.

3.6.1 ISO/IEC 29110 - Lifecycle profiles for Very Small Entities

Small organizations constitute a large part of the global software development market (LARRUCEA et al., 2016). Given the characteristics and limitations typical of small software organizations (SE), this type of organization may need specific support, depending on their difficulties, in order to achieve quality in their products and services (ISO/IEC, 2016a). To help these organizations to use the concepts, practices and processes defined in international standards of software engineering, ISO/IEC 29110 was developed.

ISO/IEC 29110 is a series of International Standards and Technical Reports that aim at addressing the specific issues of the SE and encouraging them to assess and improve their software processes. This series recognizes the limitations of SE and therefore has as few processes and practices as possible (ISO/IEC 29110).

To address some difficulties of the SE, a set of guides has been developed. These guides are based on subsets of appropriate standards processes, activities, tasks, and outcomes, referred to as “profiles”. The purpose of a profile is to define a subset of International Standards relevant to the SE context (ISO/IEC, 2016a).

ISO/IEC 29110 profiles are grouped to be applicable to more than one SE category, accordingly with the aspects and characteristics of the organization. Thus, a generic profile group was developed, which may be applicable to most SEs that have typical situations and do not develop critical software (O’CONNOR; LAPORTE, 2011). The profiles of the generic profile group are identified in Table 5:

Table 5 - ISO/IEC Profile groups

Profile groups	Profile
Generic	Entry
	Basic
	Intermediate
	Advanced

The basic profile has project management processes and enterprise software development for small entities (organization, department or project up to 25 people). The basic profile targeted at organizations developing a single application by a single work team. The Entry, Intermediate, and Advanced profiles groups have not yet been published (ISO/IEC, 2016a).

ISO/IEC 29110 is composed of several parts grouped into three main categories: Description, Profiles and Guides. The Description category has the purpose of introducing the concepts necessary to understand the other parts. The categories Profiles and Guides, aim to define the profiles supported by the standard and to guide the implementation of the norm. In total series ISO/IEC 29110 is constituted of 5 parts:

• **Part 1 Overview:** This part of the standard is intended to introduce the main concepts used in the ISO/IEC 29110 series, as well as clarify the characteristics of the SEs and why this is the target profile defined.

• **Part 2 Framework and taxonomy:** presents the standardized profiles for SEs, establishes the coherence related to the definition and application of standardization of the standard, specifies the elements in common between the developed and introduces the catalog of the profiles of the standard.

• **Part 3 Assessment guide:** defines the process assessment guidelines needed to meet the purpose of defined SE profiles.

• **Part 4 Profile specification:** defines all profiles in a profile. It is aimed at authors of supporting materials such as guides and tools.

• **Part 5 Management and engineering guide:** provides an implementation guide for the profiles described in part 4 of the standard.

ISO/IEC 29110 presents an AM that contains 5 phases (Planning, Data collection, Data validation, Process attribute rating, Reporting) and defines each activity that must be performed in each one of them. This standard also defines how the assessment activities should to be carried out, as well as provide templates to aid in the assessment, by incorporating the assessment process described in the update version of ISO/IECC 15504 (ISO/IEC 33002).

4 STATE OF THE ART

This chapter presents an analysis of the state of the art on methods for self-assessing the usability process. In order to systematically analyze the state of the art related to usability process assessment and related to process self-assessment, two systematic mappings are performed. The first review aims at analyzing usability processes capability/maturity models, and the second, aims at analyzing software process self-assessment methods in general. Systematic mapping is a type of systematic literature review, which consists of a research methodology that aims to identify, evaluate and interpret as many relevant studies as possible and available the research question, research topic or phenomenon of interest (KITCHENHAM, 2007). This process basically consists of three stages: definition, execution of the research and analysis of the results found. Details of the results can also be found in (LACERDA; VON WANGENHEIM, 2017).

4.1 STATE OF ART: USABILITY CAPABILITY/MATURITY MODELS

This section presents the systematic review of the literature on usability capability/maturity models. In order to provide an overview on the current state of the art on usability capability/maturity models we performed a systematic mapping following the procedure proposed by Kitchenham (2010).

4.1.1 Definition of the systematic mapping on usability capability/maturity models

Definition of research questions. The objective of this review is to elicit the state of the art on usability capability/maturity models. In this respect, we aim to obtain an overview on the existing models answering the following research questions:

RQ1. What are the existing UCMMs and what are their characteristics?

RQ2. To what extent do the UCMMs support their use (providing a guide, suggesting methods or tools)?

RQ3. How the UCMMs were developed in terms of source and methodology? And how were they validated?

RQ4. Are there UCMMs with a specific focus for a given context especially related to recent trends in software development?

Data source and search strategy. We examined all published English-language articles on usability capability/maturity models that are available on SCOPUS with free access through the CAPES Portal[1]. To increase publication coverage including grey literature, we also used Google Scholar, which indexes a large set of data across several different sources.

Inclusion/exclusion criteria. In accordance to our research objective/questions, we only consider studies written in English, whose focus is to propose a usability maturity or capability model, published in the last 24 years between January 1993 (initial release of CMM (PAULK, 1995)) and March 2017). On the other hand, we excluded:

- Studies that developed/applied a UCMM, but do not present the model explicitly.
- Studies presenting usability models, which aim assessing software usability but not the usability process.

1 A portal for access to scientific knowledge worldwide, managed by the Brazilian Ministry on Education for authorized institutions, including universities, government agencies and private companies (www.periodicos.capes.gov.br).

- Studies presenting capability/maturity models containing usability processes, yet not focusing exclusively on usability.
- In case of duplicate reports of the same study, we consider for review the most recent complete report found.

Quality criteria. In addition to our inclusion/exclusion criteria, we also appraised the overall quality of the found studies. We considered only articles with substantial information on the UCMM detailing its components.

Definition of search string. The string use in this search contained combinations of expressions related to the research question, synonymous, related/broader concepts for each core concept synonyms, as well as abbreviations, as shown in Table 6. Due to a lack of agreement on an accepted definition of usability [6], we also include the terms “UX” and "user experience". The terms “UCD”, "user-centered design”, “HCD” and "human-centered design”, were included as they represent a system development process approach that aims to produce high usability systems (ISO/IEC, 2001b).

Table 6 - Keywords

Core concepts	**Synonyms**
Usability	UX, user experience, user-centred design, UCD, human-centred design, HCD, human computer interaction, HCI
Maturity, capability	Assessment
Model	Method

Using these keywords, the search string has been calibrated and adapted in conformance with the specific syntax of each of the data sources as presented in Table 7.

Table 7 - Search strings per repository

Repository	Search string
SCOPUS	((usability OR ux OR "user experience" OR ucd OR "user centred design" OR hcd OR "human centred design" OR hci OR "human computer interaction") AND (("maturity model" OR "capability model" OR "assessment model") OR ("maturity method" OR "capability method" OR "assessment method"))) AND PUBYEAR > 1992)))
Google Scholar	Search string 1: ((usability OR ux OR "user experience" OR ucd OR "user centred design" OR hcd OR "human centred design" OR hci OR "human computer interaction") AND (("maturity model" OR "capability model" OR "assessment model")) Search string 2: ((usability OR ux OR "user experience" OR ucd OR "user centred design" OR hcd OR "human centred design" OR hci OR "human computer interaction") AND (("maturity method" OR "capability method" OR "assessment method"))

4.1.2 Execution of the search

Search and selection. The search has been executed in March 2017. Initial searches were carried out and resulted in the selection of 78 articles. In the second selection step, the articles found were verified against the inclusion/exclusion criteria. Table 8 presents the number of articles found and selected per repository in each selection stage.

Table 8 - Number of articles per repository per selection stage

Repository	**Initial search results**	**Selected after 1º stage**	**Selected after 2º stage**
SCOPUS	417	27	8
Google Scholar	16.000 (search sting 1)	51	14

Total	26.817	78	15 (eliminating 7

In the first analysis stage, we quickly reviewed titles, abstracts and keywords to identify papers that matched the exclusion criteria, resulting in 78 potentially relevant articles. Considering the large amount of results with Google Scholar and the fact they are ordered by relevance; the first 150 results of both Google Scholar searches were analyzed. Then, we analyzed the complete text of the remaining articles in order to check their accordance regarding the inclusion/exclusion criteria. This step led to the exclusion of 56 articles that were either not written in English, or presented case studies not presenting a UCMM, or that present usability models to assess software products but not the usability process. We also applied the quality criteria excluding articles that did not provide sufficient information on the model. As result, 15 studies were considered relevant, as shown in Table 10.

We also checked if the articles reported by (JOKELA et al., 2006) (our control paper) and that meet our inclusion criteria were found. Except for the HCD-PCM Design and HCD-PCM Visioning models, all articles presented in (JOKELA et al., 2006) were found also in this literature review, however, only 4 of them were included in this review (as indicated in bold in Table 9). As the other articles identified in (JOKELA et al., 2006) do not fulfill our inclusion criteria they have not further considered here. We explicitly list the justification for their exclusion in Table 9.

Table 9 - Articles found by Jokela (2006) not included in this systematic mapping

Articles presented in (JOKELA et al., 2006) **not included in this review**	**Justification for exclusion**
Trillium	Although this model includes usability processes, it does not specifically focus on the usability process.

Usability Leadership Maturity Model	Full work not available online
Humanware Process Assessment	Does not present a capability/maturity dimension.
User Centred Design Maturity	Full work not available online
Procedures for Usability Engineering Process Assessment	Not written in English
Human Factors Integration Process Risk Assessment	Full work not available online

In addition to the articles analyzed by Jokela (2006) that fulfilled our inclusion and quality criteria (marked in bold in Table 10), we encountered eleven more UCMMs.

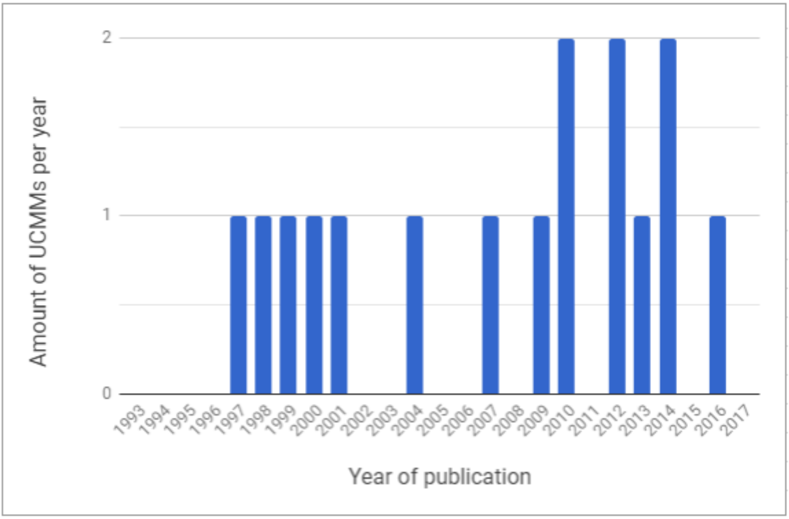
Table 10 - Articles found in the literature review (1- 15)

Reference ID	Author(s)	Article title	Repository
[7]	Raza et al., 2012	An open source usability maturity model (OS-UMM)	SCOPUS and Google Scholar
[8]	Earthy, 1998	**Usability Maturity Model: Human Centredness Scale**	Google Scholar
[9]	Marcus et al., 2009	Validating a Standardized Usability/User-Experience Maturity Model: A Progress Report	SCOPUS and Google Scholar
[10][11]	Staggers and Rodney, 2012 HIMSS, 2011	Promoting Usability in Organizations with a New Health Usability Model: Implications for Nursing Informatics	Google Scholar
[12]	Chapman and Plewes, 2014	A UX Maturity Model: Effective Introduction of UX into Organizations	SCOPUS and Google Scholar
[13]	Kieffer and Vanderdonckt, 2016	STRATUS: a questionnaire for strategic usability assessment	SCOPUS and Google Scholar
[14]	Earthy, 2000	Human Factors Integration Capability Maturity Model – Assessment model	Google Scholar
[1] [15]	Earthy, 1999; ISO, 2000	**Usability Maturity Model-Processes**	SCOPUS
[16]	Sward and Macarthur, 2007	Making User Experience a Business Strategy	Google Scholar

[17]	Van Tyne, 2009	Corporate User Experience Maturity Model	SCOPUS and Google Scholar
[18]	Mostafa, 2013	Maturity Models in the Context of Integrating Agile Development Processes and User Centred Design	Google Scholar
[4]	Jokela, 2001	**Assessment of user-centred design processes basis for improvement action**	Google Scholar
[19]	ISO, 2010	**ISO/IEC 18152 - A specification for the process assessment of human-system issues**	Google Scholar
[20]	Peres et al., 2014	AGILEUX Model – Towards a Reference Model on Integrating UX in Developing Software using Agile Methodologies	SCOPUS and Google Scholar
[21]	Vasmatzidis et al., 2001	Introducing usability engineering into the CMM model: an empirical approach	SCOPUS and Google Scholar

The first studies proposing UCMMs were published in 1998. Figure 13 illustrates that the amount of studies publishing new usability capability/maturity models is linear increasing, indicating that this topic is of continuous interest and is still being researched.

Figure 13 – Amount of UCMM studies per year



4.1.3 Data extraction

We systematically extracted data from the studies in order to answer the research questions as specified in Table 11.

Table 11 - Specification of the data extracted from the studies

Research question	**Data**		**Description**
	Study author and title		
RQ1	Capability or/and maturity		To indicate if the study presents a usability capability and/or maturity model.
	Measurement framework	Measurement structure definition	To indicate if the model developed or suggested a measurement structure.
		Scale levels	To indicate the capability and maturity models scale levels.

	Reference model	Reference model definition	To indicate if the model developed or suggested the use of a reference model.
		Process decomposition	To describe how the reference model is presented in terms of process, practices or attributes.
RQ2	Model usage support		To indicate if the model describes how to be used or provides support for its application.
RQ3	Model source, methodology and validation		To indicate which source and methodology were used to develop the UCMM and how it was validated.
RQ4	Specific development context		To indicate whether the model was developed for some specific context, such as agile, open source, small businesses, etc.

The extracted data are presented in the data analysis section.

4.1.4 Data analysis

This section presents an analysis of the data extracted from the studies in accordance to the defined research questions.

RQ1. What are the existing UCMMs and what are their characteristics?

In total, we encountered 15 UCMMs that focus on assessing usability engineering processes. To facilitate the reading of the dissertation the articles found in the literature review are identified by their number, according to Table 12. Most models represent research results (14 models), only one standard has been encountered, with the model [1] being later on transformed into the ISO18529 standard. An overview of the main characteristics of the models is presented in Table 12.

Table 12- Overview on general characteristics of the models

Model reference	Capability (C) or/and Maturity (M) model	Defines measurement framework	Explicitly defines reference model	Elements of process decomposition
[7]	M	Yes	No	--

[8]	M	Yes	No	Process attributes
[9]	M	Yes	No	--
[10][11]	M	Yes	No	--
[12]	M	Yes	No	Indicators
[13]	M	Yes	No	Indicators
[14]	C	Yes	Yes	Base practices and associated work products
[1][15]	M/C	Yes	Yes	Base practices and associated work products
[16]	M	Yes	No	--
[17]	M	Yes	No	--
[18]	M	Yes	Yes	Base practices
[4]	C	Yes	Yes	Process and outcomes
[19]	C	Yes	Yes	Practices and outcomes
[20]	M	Yes	No	Practices, recommendations, techniques and artifacts
[21]	M	Yes	No	Practices

Type of model. Analyzing the purpose of the models, we carefully classified them into "Maturity" or "Capability" model in accordance to the definition given in Section 2. Following this definition, one model [1] was classified contrary to the terminology used by the author. As a result, we can observe that the majority of the encountered articles propose maturity models, only 4 present capability models.

Measurement framework. Basically, all models present a scale for measuring process capability or organizational maturity. The only exception is model [4] by which processes are only assessed up to level 1 of the process capability scale of the ISO/IEC 15504 standard, since the assessment is concerned on whether a process truly implements user-centered design and not how the project or process are managed etc. (ISO/IEC, 2004). All encountered capability models use the same scale, adopting ISO/IEC 15504's scale with six levels ranging from level 0. Incomplete to level 5. Optimizing.

On the other hand, the maturity models use different scales as shown in Table 12. None of them corresponds with the maturity scale proposed by ISO/IEC 15504 (ISO/IEC, 2004). Some models [9], [16], [20] and [21] are based the maturity scale of the CMMI (TEAM, 2010). And, although the maturity models use different labels for each level, the meaning of the scales is quite similar. All the models subdivide maturity in more or less the same amount of levels. For example, all scales define a level at which no usability process [7], [8], [9], [18] (or few usability processes) are performed. In general, they define a level at which processes are performed, but in an unstructured way [9], [12], [16]. In addition, all scales represent a level of maturity at which usability processes are managed. On higher levels they vary more in terms of issues they deal with. Some scales define levels to represent the maturity of organizations, whose usability processes are integrated with other processes of the organization [8], [9], [10], [12], [17]. Other models use the higher levels to represent organizations in which processes are used not only in isolated projects, but in the organization as a whole [8], [10], [12], [13], [17]. Yet, the majority of the scales define that the highest level of maturity is achieved when its processes are being constantly assessed and improved [7], [8], [9], [10], [12], [13], [14], [1], [16], [17], [19]. Models [20] and [21] define their scales practices for level 2 and 3 respectively, stating that the other levels will be discussed in future work. An exception is model [13] as it defines only 3 maturity levels. Following this scale, at the first level, usability management is done in an ad-hoc way, without dedicated resources and only individual efforts to implement UE processes. On the second level processes are repeatable, there are allocated resources and usability benefits are recognized. At the highest level, usability is constantly managed, and the importance of users is recognized. This simplified scale makes it less precise to classify less mature organizations, since it does not provide the necessary levels to differentiate organizations that do not implement any process, from organizations that implement few processes, or that implement all processes but in an unstructured way.

Table 13 - Maturity scales of the encountered models

ID	Maturity Scales
[7]	1. Preliminary 2. Recognized 3. Defined 4. Streamlined 5. Institutionalized
[8]	X. Unrecognized A. Recognized B. Considered C. Implemented D. Integrate E. Institutionalized
[9]	1. Initial 2. Repeatable3. Defined 4. Managed 5. Optimized
[10]	1. Unrecognized 2. Preliminary 3. Implemented 4. Integrated 5. Strategic
[12]	1. Beginning 2. Awareness 3. Adopting 4. Realizing 5. Exceptional
[13]	1.Initial 2.Tactical 3.Strategical
[16]	1.Initial 2. Repeatable 3. Defined 4. Managed 5. Optimized
[17]	Initial, Professional Discipline, Managed Process, Integrated User experience, Customer-Driven Corporation
[18]	0. Not Possible 1. Possible 2. Encouraged 3. Enabled / Practiced 4. Managed 5. Continuous Improvement
[20]	1.Initial 2. Repeatable 3. Defined 4. Managed 5. Optimized
[21]	1.Initial 2. Repeatable 3. Defined 4. Managed 5. Optimized

Process reference model. Each usability capability/maturity model should implicitly or explicitly define a process reference model that presents the ideal UE model (JOKELA, 2004b). A process reference model defines the elements of the process that should be examined in an assessment. Some UCMMs (including [8], [12], [13], [20] and [21]) however, mix the concepts of measurement structure and reference model. Instead of presenting a separate process reference model, they present practices or indicators for each level of the measurement scale. Only 5 UCMMs present a conformant process reference model by decomposing the process into process attributes (such as, Process performance and Performance management attributes), base practices (such as, identify user attributes, and analyze user trends), work products (such as, Trend analysis and User Interaction Specification) and indicators (such as, use a User-driven methodology and Each key usability function is filled).

Regarding the content assessed by the models, it is observed that most of them focus on management issues, such as staff usability awareness, staff training, management of usability resources, management of the usability process and the integration of usability processes. Model [15] deals only with issues related to the performance of the UE processes, that is, the extent to which the UE processes are executed, such as UE methods are selected, UE requirements generated, user attributes identified, prototypes developed and assessed. Models [4], [19], [20], [21] assess both management and performance issues of UE processes.

The presented reference models differ also in respect to their structure as shown in Table 12. The majority of the process reference models decompose the processes into practices [14], [1], [18], [19], practices and working products [14], [1] or practices and outcomes [19]. Practices describe what has to be done in order to represent and include the users of the system in the lifecycle [1]. Work products describe artifacts, such as documents, pieces of information, products or other items that acts as input or output to the process [1]. Outcomes indicate significant assessable results of the achievement of the process [19].

RQ2. To what extent do the UCMMs support their use (guide, method or tool)?

Although SPCMMs are not intended to define an assessment method, the lack of further support for their application in practice, may entail the risk that the model is not applied as originally intended by authors (JOKELA et al., 2006). Thus, in order to increase the adoption of these models in practice, it is of great importance to provide an assessment method or at least to suggest a suitable method for its use.

Overall, one third of the articles does not present any kind of usage support for the application of the model proposed [9], [12], [16], [17], [20] and [21]. The other models provide different levels of use support, as shown in Table 14, providing guides or manuals, or any other information/guidance that makes it easier to use the UCMM.

In order to adequately use the models for the establishment of UE processes it is fundamental to define the most important assessment domain-related key terms. Examples of UE key terms are "context of use", "quick and dirty evaluation", or "task analysis". However, only a minority of the models ([8], [1], [4] and [19]) presents such a glossary. Some models present tools for data collection, to facilitate the recording of information during interviews, including recording forms [8], [1], [18] or questionnaires [7], [10], [13]. Some of them, however, do not provide detailed information on how to conduct the assessment, or how to use the provided tool [7], [10], [13], as shown in Table 14. Models [8], [14], [1], [18] and [19] recommend collecting data via interviewing selected persons, while model [4] proposes the use of workshops. Model [19], on the other hand, only recommends using workshops for informal assessments.

Model [4], differently from the others, explicitly defines a detailed assessment process composed of several steps: setting up PAM, planning, document examination, interview session, wrap-up session, results session and follow-up session. This model addresses in detail the process for data collection, through workshops, as well as the compilation of the results to provide the assessment feedback. Yet, it does not provide information on how the assessment should be planned or which stakeholders should be involved.

All UCMMs (except for [4]) provide assessment feedback through a maturity or capability scale. Model [4] presents a *process capability profile* and *process findings.* Process capability profile is a quantitative representation of the assessment results in form of a process diagram with different chart types to show the different performance dimensions. *Process findings* are presented in a table of qualitative findings and justifications for the rating.

Several models [8], [14], [1], [18] and [19] are derived from ISO/IEC 15504, and, therefore, have a similar content and structure. These models provide greater guidance defining how to plan, select stakeholders, collect data, rate the process and how to calculate the process capability/maturity level. The only exception among them is model [14], which defines just a way for data collection and which person should assume the role of assessor.

Table 14 - Overview on the application guidance provided by the models

Model reference	**Defines key terms**	**Provides tool for data collection/ questionnaire**	**UCMM defines process for:**				
			Planning	**Data collection**	**Process rating**	**Assessor/ Interviewee**	**Calculate final score / provide feedback**
[7]	-	x	-	-	-	-	x
[8]	x	x	-	x	x	x	x
[9]	-	-	-	-	-	-	-
[10]	-	x	-	-	-	-	-
[12]	-	-	-	-	-	-	-
[13]	-	x	-	-	-	-	-
[14]	-	-	-	x	-	x	-
[1]	x	x	x	x	x	x	x
[16]	-	-	-	-	-	-	-
[17]	-	-	-	-	-	-	-
[18]	x	x	x	x	x	x	x
[4]	-	-	x	x	x	-	x
[19]	x	-	-	x	x	x	x

RQ3. How were the UCMMs developed in terms of source, methodology and validation?

Source. Most of the articles present the source(s) used as basis to develop the UCMM. As shown in Table 15, most of the models are based on earlier UCMMs, usability process reference models or process assessment models.

Table 15 - Overview on the sources, development methodology and validation

ID	Source model(s)	Model definition	Model validation
[7]	Model's author previews studies.	No details about the model development are provided, the authors only state that: "The measuring instrument of the model contains factors that have been selected from four of our empirical studies."	Validation
[8]	(FLANAGAN, 1995), (ISO/IEC, 2004), (ISO/IEC, 1999), [29].	Not informed	Not informed
[9]	Not informed.	Not informed	Not informed
[10]	[30], (JOKELA et al., 2006), [8].	Not informed	Not informed
[12]	Not informed.	No details about the model development are provided, the authors only state that "6 indicators were defined based on the author's experience."	Not informed
[13]	[8], (JOKELA et al., 2006), [30].	Not informed	Validation
[14]	[15], (ISO/IEC, 1999), (ISO/IEC, 2004) and HF process models available up to 1998.	The model was developed / reviewed by a group of more than 50 experts. Earlier versions of the descriptions have been used and feedback has led to many of the changes between TR 18529. But it does not present details of the model development.	Assessment
[1]	(ISO/IEC, 1999), (ISO/IEC, 2004), [32], [33].	Not informed	Not informed
[16]	Studies found in a SLR.	Not informed	Not informed
[17]	(TEAM, 2010), [31].	Not informed	Not informed
[18]	(ISO/IEC, 1999), [4].	The development methodology was fully described and was based on Hevner et al (2004), DeBruin (2005) and Mettler (2011). The maturity dimensions are based on ISO/IEC 13407 (1999) and Jokela (2001). The model development occurred in 4 phases: scope, design, populate and test.	Assessment

[4]	(ISO/IEC, 1999)	The methodology used was adapted from frameworks proposed by March & Smith (1995) and Järvinen (2000) and consists of the steps: analyze existing models, artifact building and evaluation through experimental assessments, and theorizing.	Assessment
[19]	(ISO/IEC, 1999), (ISO/IEC, 2004).	Not informed	Not informed
[20]	(TEAM, 2010), [22], (SILVA et al., 2011), [23], [24], [25].	Not informed	Assessment
[21]	[5].	Not informed	Not informed

Development methodology. However, most articles do not describe the methodology used to develop the UCMM ([8], [9], [10], [13], [1], [16], [17], [19], [20] and [21]). Model [7] and [12] are reported to be developed based on the authors' experience. The author of model [13] mentions the sources used, but does not report how they were used in the development of the UCMM. Model [14] does not present in detail the methodology used, but the author mentions that initial versions of the model were applied and based on the received feedback improvements were made.

The articles proposing model [18] and [4] are the only ones that present in more details the methodology used. Model [18] has been developed by using a systematic methodology for the development of SPCMMs proposed by [2] and [26], including the phases: scope, design, populate and test. Model [4] has been developed by using a research framework adapted from March & Smith [27] and Järvinen [28]. The framework defines a series of activities necessary to produce new knowledge consisting of the steps: analyze existing models, artifact building and evaluation through experimental assessments, and theorizing.

Validation. Only 6 studies present some form of validation or evaluation of the developed UCMM, as presented in Table 15. Only two models ([7] and [13]) present the validation of the proposed questionnaires. The questionnaire of model [7] has been evaluated in terms of its reliability and construct validity. The validation has been conducted in several steps. First, a pilot test has been run in which 10 project managers answered the proposed questionnaire. Analyzing the alpha coefficient based on the collected data, the authors conclude that all questionnaire items are reliable. Then, two case studies were conducted, one with 4 and another with 6 respondents. Analyzing the inter-rater agreement by using both Kendall and Kappa statistics it was concluded that the questionnaire contributes to establish a comprehensive strategy for the usability maturity of open source software projects.

The authors of model [13] performed a validation of the questionnaire utility. It was conducted as a case-control study in order to compare the questionnaire responses of 28 persons divided in two groups, novices and experts in usability. The study highlighted some dependencies between experience in usability and usability awareness, together with some positive correlations between the presence of usability staff and strategic management of usability. However, the authors also point out that the questionnaire still needs to be validated by means of a longitudinal study.

A few models ([14], [18], [20] and [4]) have been evaluated using different approaches. Model [14] and [20] were assessed through expert panels. The author of UCMM [20] does not provide information about its assessment execution or results. The expert panel of model [14] involved more than 50 experts from 40 organizations worldwide. Yet, again the author concludes that at the present state of development, the validation for all objectives has not been achieved. Model [18] has been assessed in terms of both, the model and the data collection instrument. The assessment was performed in 2 phases: (1) author evaluation: in which the author examined the model development phases and the model as a product of design and phase and (2) domain expert evaluation, in which 3 experts were encouraged to elaborate their answers and to suggest any improvements related to maturity levels, processes, practices, scoring scheme or assessment guidelines. The result of both assessments led to the evolution of the original model into a number of subsequent versions. Model [4] was built based on five case studies in which it was used to perform assessments. The qualitative data gathered during the assessments from the stakeholders was used to guide the development of the model.

RQ4. Are there UCMMs with a specific focus for a given context?

Most UCMMs aim at assessing the UE process in general (or similar subjects such as HCD and UX), as shown in Table 16, so, they are intended be used for the development of any type of software in any context.

However, some of the models are customized to specific domains or development contexts. One model [10] focuses on UE in a specific application domain, nursing systems. Three models are specific tailored to particular development environment [7], [18] and [20]. Models [18] and [20] focus on the integration of the agile approach and usability engineering following this trend in software development. Model [18] presents practices regarding the communication, coordination, and collaboration between UE practitioners and agile developers in order to synchronize and complete their work. It also approached features and activities that should be played by some team roles in agile and UE. Model [20] presents specific practices, recommendations and techniques such as that user experience designers should always work one or more sprints ahead of other developers and that design and development teams must be physically near each other so as to enhance the communication and exchange of agile documents. Model [7] on the other hand, represents a UCMM in the open source development context aiming at the coordination of open source software with usability-related process activities.

Although, most of the UCMMs are supposed to be applicable in any kind of context, we observed that only two models [13] and [4] reported the models' application and validation in a specific context. Model [13] was evaluated by expert panel with experts from different domains (traffic control, health care and R & D), yet not presenting specific comments about the applicability of the model in these specific domains. And, although, UCMM [4] was not developed specifically for a particular domain, it was used to conduct an assessment in a company that develops software for mobile phones. According to the model's author, the assessment results showed that it is a challenge to develop a process model that covers different situations. In this assessment specifically, were observed challenges to understand the usage context process, since the environment of use of mobile applications is quite different from desktop applications.

Table 16 - UCMMs domain

ID	UCMM domain for which it was designed	Domain in which UCMM was assessed
[7]	Open source usability	Open source usability
[8]	Usability	-
[9]	Usability/user experience	-
[10][11]	Usability in healthcare	-
[12]	UX	-
[13]	Usability	Experts from traffic control, healthcare and R&D sectors
[14]	Human factors	-
[1][15]	Usability	-
[16]	UX	-
[17]	UX	-
[18]	Agile and Usability	-
[4]	Usability	Software for mobile phones and telecommunication industry.
[19] [15]	Human systems	-
[20]	Agile	-
[21]	Usability	-

4.1.5 Discussion

Based on our review, we found few (only 15) UCMMs being published during the last 24 years. Overall, most of them aim to assess the maturity of the organizations regarding the usability engineering process.

Half of the encountered models focus on usability exclusively without providing a potential integration of the model into commonly ones used, such as CMMI or ISO/IEC 15504 [4], [7], [9], [10], [12], [13], [16], [17]. Such an integration could facilitate their adoption by organizations using already one of the prominent ones, as we observed also that the proposed UCMMs do not seem to be largely adopted in practice so far.

In terms of the measurement framework most of the models basically follow the capability/maturity level definition of ISO/IEC 15504 and/or CMMI. Differences with respect to these generic well-accepted models are mostly related to the process reference model. Yet, in this regards it seems contradictory that half of the models ([7], [9], [10], [16], [17], [20] and [21]) do not present a specific process reference model. Most of the UCMMs ([1], [4], [14], [18] and [19]) that provide a process reference model define base practices, work products or outcomes.

Despite the importance of supporting the use of UCMMs, analyzing the model's guidance to perform assessments, we can classify the models in three groups. Some models do not present any information on the intended use of the model ([9], [12], [16], [20] and [21]).A second group of models present little guidance, e.g. presenting only a data collection instrument ([7], [10], [13], [14]). Such a lack of information on how to apply these models in practice may significantly hinder their larger scale application. Only a few UCMMs ([8], [1], [18], [4], [19]) provide also information on the assessment process in more detail, defining each step of the assessment, as well as defining data collection instruments. Apart from model [7] and [8], the encountered models require the conduction of the assessment by an experienced usability professional. In addition, these models require that assessors are familiar with standards documentation, such as the ISO/IEC 15504, or ISO/IEC 13407. Although models [7] and [8] were tailored to be used in self-assessments, they do not explicitly present the factors that make them different from the other models.

Analyzing how the UCMMs have been developed and validated, we observed that most studies mention the sources used, but only two studies ([13] and [18]) justify their choices. In general, the studies do not present the methodology used to develop the model. Only few models have been validated or assessed ([7], [13], [14], [18], [4], [20] and [21]) and most of them present only an initial attempt using non-experimental methods with very small samples. Thus, the lack of scientific rigor of the validation of the proposed models may leave their validity, reliability and generalizability questionable.

Regarding the focus of UCMMS on specific contexts, we observed that most of the encountered models are defined in a generic way supposed to be applicable to any type of software or context. Yet, in general, no large-scale evaluations of the proposed models across different contexts have been encountered, thus leaving their general applicability questionable. In addition, such generic UCMMs may miss domain specific problems [3]. Only models [18], [20] and [7] take into consideration specific development contexts. The first two models integrate agile principles and usability engineering, while [7] developed a UCMM for open source development. Especially the research integrating UE and agile practices indicates a need for customizing such models to current software development trends. However, no further customizations, for example, taking into consideration software development for mobile devices have been encountered.

In comparison to the review presented by JOKELA et al. (2006) more than ten years ago, we encountered further UCMMs thus pointing out a continuous interest in this area, as well as through the few customizations also the need for tailoring such models to current development trends. However, we observed that several of the original findings identified by Jokela, seem not to have been improved over time, as we observed that still most of the UCMMs do not report how they have been developed and only a few present very little empirical evidences with respect to their validity, reliability and generalizability.

4.2 STATE OF ART: SELF-ASSESSMENT METHODS FOR ASSESSING SOFTWARE PROCESS

This section presents the systematic review of the literature on software process self-assessment methods. In order to provide an overview on the current state of the art on software process self-assessment methods, we perform a mapping study. The mapping study aims at providing a broad review on existing reports on software process self-assessment methods, classifying them and describing their methodology and results. The research questions of this study focus on which software process self-assessment methods exist and what are its characteristics, especially with respect to their process reference models and measurement frameworks. We also analyze how these methods were evaluated. As mapping studies use the same basic methodology as systematic mappings, this study follows an adaptation of the procedure proposed by (KITCHENHAM et al., 2010; PETERSEN et al., 2008).

4.2.1 Definition of the mapping study

The main research question driving this study is: what is the state of the art on software process self-assessment methods? Furthermore, we want to obtain an overview on the existing methods answering the following sub-questions:

RQ1. Which software process self-assessment methods exist?

RQ2. What are the software process self-assessment methods characteristics in terms of assessment process, techniques and stakeholders?

RQ3. What are the characteristics of the process reference models?

RQ4. What are the characteristics of the measurement frameworks?

RQ5. Have the methods been developed and evaluated systematically?

Data source and search strategy: We examined all published English-language articles on software process self-assessment methods that are available on the Web via major digital libraries and databases in the computing field (IEEE Xplore, ACM Digital Library, Wiley, Springer and SCOPUS) with free access through the CAPES Portal[2]. To increase publication coverage, we also used Google Scholar, which indexes a large set of data across several different sources (HADDAWAY et al., 2015).

Inclusion/exclusion criteria: We only include studies, which present a software process self-assessment method, published in the last 24 years (starting at the initial release date of the CMM (PAULK, 1995)) between January 1993 and June 2017. On the other hand, we excluded:

---

[2]A web portal for access to scientific knowledge worldwide, managed by the Brazilian Ministry on Education for authorized institutions, including universities, government agencies and private companies (www.periodicos.capes.gov.br).

- Studies that present methods for the self-assessment of objects that are not software processes, such as products etc.
- Studies that developed a software process self-assessment method, but do not present the AM.
- Studies that present self-assessment methods for other contexts not related to software processes.
- Studies not written in English.

In case of duplicate reports of the same study, we consider the most recent complete report found.

Quality criteria: In addition to our inclusion/exclusion criteria, we also appraised the overall quality of the found studies. We considered only articles with substantial information on the process assessment method regarding our research questions.

Definition of search string: In order to calibrate the search string, we conducted informal searches in the aforementioned repositories. The string use in these searches contained combinations of expressions related to the research question, synonymous, related/broader concepts for each core concept synonyms, as well as abbreviations. The test strings contained combinations of the expressions self-assessment, “internal assessment”, software, “software process”, method, methodology, capability, maturity, guide and framework. After the calibration, the string selected was “(self-assessment OR "internal assessment") AND process AND software AND ("maturity" OR "capability") NOT (students AND education)”. Table 17 presents the search string used to perform the search in each repository.

Table 17 - Search string per repository

Repository	Search string
Springer Link	(self-assessment OR "internal assessment") AND process AND software AND ("maturity" OR "capability") NOT (students AND education)' within Engineering Computer Science English

Wiley Online Library	self-assessment OR "internal assessment" in All Fields AND process in All Fields AND software in All Fields AND "maturity" OR "capability" in All Fields NOT students AND education in All Fields
ACM Digital Library	(self-assessment OR "internal assessment") AND (process) AND (software) AND ("maturity" OR "capability") NOT (students AND education)
IEEE Xplore	(((self-assessment OR "internal assessment") AND (process) AND (software) AND ("maturity" OR "capability") NOT (students AND education)))
SCOPUS	ALL ((self-assessment OR "internal assessment") AND (process) AND (software) AND ("maturity" OR "capability") AND NOT (students AND education)) AND PUBYEAR > 1992
Google Scholar	"internal assessment" "software" "process" maturity capability -students -education
	"self-assessment" "software" "process" maturity capability -students -education

4.2.2 Execution

The search has been realized in June 2017. Table 18 presents the number of articles found and selected per repository in each selection stage.

Table 18 - Number of identified articles per repository per selection stage

Repositories	No. of initial search results	No. of articles analyzed	No. of articles selected after 1º stage	No. of articles selected after 2º stage
Springer Link	189	189	6	0
Wiley Online Library	664	664	27	2
ACM Digital Library	337	337	28	0
IEEE Xplore	604	604	39	9
Google Scholar	222 (search string 1)	900	142	27

	2040 (search string 2)			
SCOPUS	467	467	46	14
Total	4523	3383	287	33 (discounting 19 duplicates)

In the first analysis stage, we quickly reviewed titles and abstracts to identify papers that matched the inclusion criteria, resulting in 287 articles potentially relevant. In the second stage, the articles were fully read with the objective to check their relevance with respect to our inclusion/exclusion criteria. In this step, 254 articles were excluded, most of them due to the fact that they address other forms of assessment than self-assessment, or deal with self-assessment of processes not related to software development. In this step, we also evaluated the articles with respect to the quality criteria. Some of the studies found (6) present case studies in which self-assessments were carried out in software companies or present the development of an assessment method, but do not provide enough details to analyze the AM, so they were also excluded. As result, 33 studies were considered relevant, as shown in Table 19.

Table 19 - Self-assessment methods found in the mapping study

ID	Reference	Title
1	Garcia et al., 2010	Adopting an RIA-Based Tool for Supporting Assessment, Implementation and Learning in Software Process Improvement under the NMX-I-059/02-NYCE-2005 Standard in Small Software Enterprises
2	Graden and Nipper, 2000	An Innovative Adaptation of the EIA/IS 731.2 Systems Engineering Capability Model Appraisal Method
3	Muladi and Surendro, 2014	The readiness self-assessment model for green IT implementation in organizations
4	Glanzner & Audy, 2012[3]	2DAM WAVE An Evaluation Method for the WAVE Capability Model

[3] This study presents two assessment methods, however only the mini assessment version is considered in this research, as the extended

5	Widergren et al., 2010	Smart grid interoperability maturity model
6	Burnstein et al., 1998	A Model to Assess Testing Process Maturity
7	Grceva, 2012	Software Process Self-Assessment Methodology
8	Amaral & Faria, 2010	A Gap Analysis Methodology for the Team Software Process
9	Serrano et al., 2003	An experience on using the team software process for implementing the Capability Maturity Model for software in a small organization
10	Bollinger & Miller, 2001	Internal capability assessments
11	Shrestha et al., 2015	Evaluation of Software Mediated Process Assessments for IT Service Management Building a Software Tool for Transparent and Efficient Process Assessments in IT Service Management
12	Blanchette & Keeler, 2005	Self Assessment and the CMMI-AM – A Guide for Government Program Managers
13	Wiegers & Sturzenberger, 2000	A Modular Software Process MiniAssessment Method
14	Yucalar & Erdogan, 2009	A Questionnaire Based Method for CMMI Level 2 Maturity Assessment
15	Kasurinen et al., 2011	A Self-assessment Framework for Finding Improvement Objectives with ISO/IEC 29119 Test Standard
16	Karvonen et al., 2012	Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain
17	Varkoi, 2010	Process Assessment In Very Small Entities - An ISO/IEC 29110 Based Method
18	Shrestha et al., 2014	Software-mediated process assessment for IT service capability management
19	Böcking et al., 2005	A Lightweight Supplier Evaluation based on CMMI

---

assessment version requires the participation of an external assessor, and therefore cannot be performed by the organization in a totally autonomous way.

20	Patel & Ramachandran, 2009	Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices
21	Pino et al., 2010	Assessment methodology for software process improvement in small organizations
22	Timalsina & Thapa, 2016	Assessment of software process improvement
23	Göbel et al., 2013	Towards an agile method for ITSM self-assessment
24	Homchuenchom et al, 2011	SPIALS: A light-weight Software Process Improvement Self-Assessment Tool
25	Orci & Laryd, 2000	Dynamic CMM for Small Organisations - Implementation Aspects
26	Daily & Dresner, 2004	Towards Software Excellence
27	Coallier et al., 1994	Trillium - Model for Telecom Product Development & Support Process Capability
28	MacMahon et al., 2015	Development and validation of the MedITNet assessment framework: improving risk management of medical IT networks
29	Kar et al., 2012	Self-assessment Model and Review Technique
30	Raza et al., 2012	An open source usability maturity model (OS-UMM)
31	Rapp et al., 2014	Lightweight Requirements Engineering Assessments in Software Projects
32	Abushama, 2016	PAM SMEs process assessment method for small to medium enterprises
33	Kuvaja et al., 1999	TAPISTRY—A Software Process Improvement Approach Tailored for Small Enterprises

4.2.3 Data extraction

We systematically extracted data from the articles in order to answer the research questions. Secondary sources (e.g., academic works) were also used to complete the information of the primary articles. In accordance to the research questions, we extracted the data described in Table 20.

Table 20 - Data extracted from the studies

Research question	**Data extracted**
RQ1. Which software self-assessment methods exist?	• Author(s) and title of the AM
R2. What are the method characteristics in terms of assessment process, techniques and stakeholders?	• Activities (planning, data collection, data validation, process attribute rating and reporting) • Technique used in each activity (interview, questionnaire, focus group, etc.) • Tool support (templates, checklists, tables, etc.) • Necessity for participants to have specific knowledge of SPI • Customization for a specific domain • Effort to perform a self-assessment with the method
RQ3.What are the characteristics of the process reference models?	• SPCMM the reference model is based on • Scope (process areas)
RQ4. What are the characteristics of the measurement frameworks?	• Measurement scale • Calculation of result • Questionnaire/checklist: ○ Amount of items ○ Item format and response scale ○ Examples/explanations ○ Respondents
RQ5. Have the methods been developed and evaluated systematically?	• Research design: ○ Development methodology ○ Evaluation methodology ○ Amount of data points ○ Evaluated characteristics ○ Evaluation context(s)

Table 20 presents the data extracted from the studies in order to answer the research questions. The extracted data is presented as part of the analysis of each of the research questions in the next section.

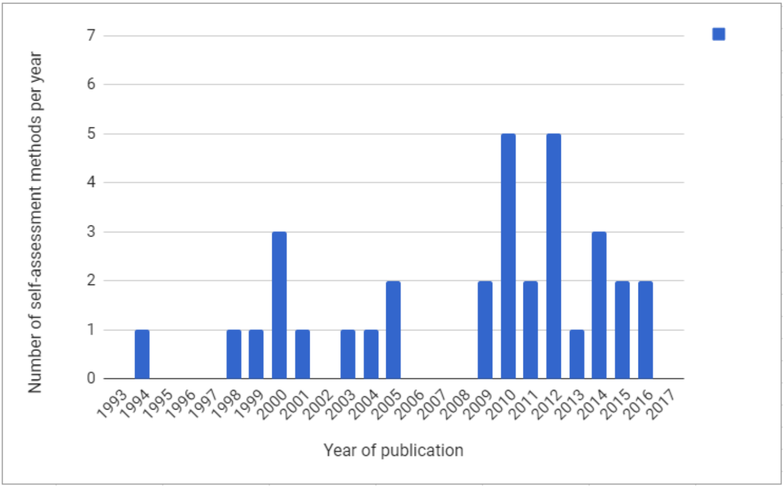
4.2.4 Data analysis

This section presents an analysis of the data extracted from the studies in accordance to the defined research questions.

RQ1. Which software process self-assessment methods exist?

Thirty-three process self-assessment methods were found as listed in Table 19. The first study found proposing a software process self-assessment method was published in 1994. Figure 14 illustrates that the amount of studies publishing new software process self-assessment methods is increasing, which indicates that this topic is of continuous interest still being researched.

Figure 14 - Number of self-assessment methods per year of publication



RQ2. What are their characteristics in terms of assessment process, techniques and stakeholders?

Analyzing this question, we observed that basically none of the published methods explicitly defines the process/steps to be performed in the self-assessment. An exception is data collection that in some way is described by almost all methods. Some articles implicitly describe the assessment activities, for example, presenting tools to carry out data collection or explaining how data validation should be performed.

More specifically we encountered the following information on the assessment process:

Planning: Fourteen-four articles discuss the planning activity, but most of them do not mention the techniques used to perform it. Three articles propose to hold meetings with the assessors and those responsible for the processes and four mention the development of documents with pertinent information about the assessment.

Data collecting: Except for one method [27], all methods present a data collection activity. Most of them use questionnaires (30 methods), 4 of which are used as script for interviews and 4 are used for performing workshops. Ten methods use interviews and five held workshops or focus group sessions. Only five articles mention performing data collection by gathering documents and other artifacts. Seven methods combine multiple ways for data collection.

Data validation: Only 8 articles mentioned the validation of the collected data. Some assessment methods ([4]; [18]; [2]), used a software system to group the collected information and automatically indicate the validity of the information. Yet, in some cases no further information has been encountered on this issue. [4] and [18] do not present how the tool identifies inconsistency in the data. The tool used by Graden et al. [2] checks for conflicting responses. Amaral et al., [8] propose the accomplishment of interviews to validate the collected information (artifacts) and an intermediate presentation of the results to management. On the other hand, [21] suggests that the assessor, in parallel to the interviews and questionnaires, collects information in order to validate participants' responses. In a similar way, [31] cross-check the initial interview results with the documents collected.

Process attributes rating: Half of the methods (16) perform the process attribute rating as suggested by the standards on which they are based, such as ISO/IEC 15504, CMM/CMMI, among others. The other methods define their own process rating, which can be performed manually or automated [4]; [6]; [7]; [18]; [22]; [26].

Reporting: Sixteen articles mention an activity regarding the report of the assessment results, of which 3 provide a template for reporting. One method [4] provides a software tool to record the lessons learned during the assessment. Other methods, besides generating a report mention the presentation of the results to the managers [4]; [28]; [31] and the realization of feedback sessions [17]. However, in general, the articles do not describe in detail the content and format of the reports. Table 21 provides an overview on the activities that are defined/mentioned in the studies by "*" and indicate the techniques proposed to carry out them. Information that is not provided in the articles is indicated by "-".

Table 21 - Self-assessment activities and techniques

Article	Assessment activities					
		Planning	**Data collection**	**Data validation**	**Process attribute rating**	**Reporting**
[1]	**Activity**	-	*	-	*	-
	Technique	-	Questionnaire	-	ISO/IEC 15504	-
[2]	**Activity**	*	*	*	*	*
	Technique	Discussions with the consultant and with the target group leadership	Questionnaire (in a spreadsheet) and interview	The questionnaire was imported into the database for validation. Charts served to provide insight about internal differences	EIA/IS 731.2	-

				within the target group.		
[3]	**Activity**	*	*	-	*	-
	Technique	Determination of readiness factors, interview questions and questionnaire.	Interview and questionnaire (software)	-	ISO/IEC 15504	-
[4]	**Activity**	*	*	*	*	*
	Technique	Requirements Analysis, Selection and Preparation of the Team and Development of the Evaluation Plan (Document)	Questionnaire (software)	The tool calculates if the data collected are valid, or if there was any relevant discrepancy between the results through a heuristic,	Mini: The responses indicate the level of each attribute, and at the end of the questionnaire, the unit capability level. Extensive: Preparing Participants, Evidence and Affirmations Collection, Evidence and Affirmations Documentation, Evidence and Affirmations Verification, Validate the First Discoveries.	All relevant artifacts are included in the WAVE's database of historical Data. The leader of the assessment presents to all stakeholders.
[5]	**Activity**	*	*	-	*	-
	Technique	Define target, domain, goals and identify practices. (Assessment and planning process diagrams).	Gather evidence and others and use maturity model tools	-	CMMI	-
[6]	**Activity**	*	*	-	*	*
	Technique	A statement of assessment purpose, scope,	Interviews, presentations, questionnaires	-	The ranking algorithm requires a	The profile can be presented as a graphical display or in the form of

		and constraints is prepared to guide the development of the assessment plan.	(soft. tool), and relevant documents.		rating of the maturity subgoals, then the maturity goals, and finally the maturity level.	a matrix that indicates maturity goals that are satisfied or not, the TMM level, a summary of test process strengths and weaknesses, and recommendations for improvements.
[7]	**Activity**	-	*	-	*	-
		-	Document-based	-	Documents are inserted in a soft. tool that statistically generates the analysis of the results.	-
[8]	**Activity**	*	*	*	*	*
	Technique	-	Interviews (based on questionnaires).	Perform interviews to validate the collected information (artifacts)	ISO/IEC 15504	Report template.
[9]	**Activity**	-	*	-	*	-
	Technique		Questionnaire		CMM	
[10]	**Activity**	-	*	-	*	-
	Technique		Questionnaire		EIA/IS-731	
[11]	**Activity**	-	*	-	*	-
	Technique	-	Questionnaire (Developed tool)		ISO/IEC 15504	
[12]	**Activity**	-	*	-	Does not apply	-
	Technique	-	Questionnaire	-	-	-
[13]	**Activity**	*	*	-	*	*
	Technique	The assessors meet with the project's software leader to plan the activities.	Questionnaire and Participant discussion	-	The assessors analyze the questionnaire responses using a spreadsheet tool.	1. Assessors present findings to project team 2. Project team presents findings to their management.
[14]	**Activity**	-	*	-	-	-
	Technique	-	Interview (based on questionnaire).	-	-	-
[15]	**Activity**	-	*	-	-	-
	Technique		Rounds of Interviews			

			(questionnaire based)			
[16]	**Activity**	-	*	-	-	-
	Technique	-	Checklist	-	-	-
[17]	**Activity**	*	*	*	*	*
	Technique	Face-to-face meetings.	Interview	-	ISO/IEC 15504	Feedback sessions
[18]	**Activity**	-	*	*	*	*
	Technique	-	Questionnaire (Developed tool)	Developed software tool calculates the coefficient of variation score.	Developed software tool.	The tool extracts a recommendation item from the knowledge base and the items are compiled into an assessment report.
[19]	**Activity**	-	*	-	*	*
	Technique	-	Questionnaire (Excel application)	-	CMMI	Delivers the supplier the results of the evaluation and plans steps for improvement
[20]	**Activity**	-	*	-	-	-
	Technique	-	Questionnaire	-	-	-
[21]		*	*	*	*	*
	Technique	-	Interview and survey (EvalTool).	The assessor gathers information separately from the person responsible for the process to be assessed so the documentation of the organization's processes is inspected.	ISO/IEC 15504	Assessment report template.
[22]	**Activity**	-	*	-	*	-
	Technique	-	Questionnaires (soft. tool) and interviews.	-	Using a software tool to determine the final score for each capability level by calculating the mean value of all the	-

					responses for that level by all the respondents.	
[23]	**Activity**	*	*	*	-	*
	Technique	-	Workshop (based on checklist)	-	When all participants understand the meaning of the statement the group discusses the different ways they work and agree on a "rating". Additional metrics (optional) are selected as complement to the self-assessment rating.	Notes and graphs
[24]	**Activity**	*	*	-	*	*
	Technique	-	Questionnaire	-	CMMI	-
[25]	**Activity**	*	*	-	*	*
	Technique	Select - Appoint - Train: involves both selection of the appropriate model, appointment of people to I-Roles, and training.	Workshop (based on checklist)	-	A defined and documented process must be approved by the working group and SEPG. If an approved status cannot be directly reached, a new workshop should be arranged.	Documented D-process
[26]	**Activity**	-	*	-	*	*
	Technique	-	Questionnaire (web based tool)	-	Responses are combined using a weighting scheme. This weighting takes into account the	A standard format of Assessment Report is available.

					number of questions on the form and the importance of each one.	
[27]	**Activity**	-	-	-	*	-
	Technique	-	-	-	-	-
[28]	**Activity**	-	*	-	*	*
	Technique	-	Focus group interviews (based on questionnaires)	-	ISO/IEC 15504	A report document is generated and presented.
[29]	**Activity**	-	*	-	*	-
	Technique	-	Questionnaire	-	SMART SPICE	-
[30]	**Activity**	-	*	-	*	-
	Technique	-	Questionnaire	-	The maturity is determined by the extent to which the project managers and developers agree with each statement in the questionnaire.	-
[31]	**Activity**	*	*	*	*	*
	Technique	Kick-off meeting inclusive handing over the requirements engineering artifacts.	Initial Document Analysis and/or Interviews (based on questionnaire in a soft. tool)	Cross-checking initial interview results and documentation.	Do not apply	Final report with diagrams and presentation to the process representatives.
[32]	**Activity**	*	*	-	*	*
	Technique	-	Questionnaire	-	SCAMPI C	List of improvement areas.
[33]	**Activity**	-	*	-	*	-
	Technique	-	Workshop (BootCheck tool)	-	ISO/IEC 15504	-

Specific methods for certain contexts of use: Twenty-four assessment methods aim at supporting the assessment of software processes in specific contexts, 10 of them in domains such as open source, usability, IT service, system engineering, Green IT, Smart grid, etc. Three methods are specific for agile development and 14 are specific for small businesses (two of which are customized to both contexts), as presented in Table 22.

SPI Knowledge: Several articles do not mention the SPI expertise required for assessors to carry out the assessment. Eight articles mention that the assessment method can be applied even by a team without specific knowledge on process assessment. On the other hand, eight articles state the need for the assessor to have knowledge on SPI. Other studies report the possibility of conducting trainings if necessary.

Table 22 - Contexts of use and SPI knowledge requirements

Article	Context of use	Knowledge in SPI
[1]	-	Requires knowledge in SPI
[2]	SME enterprises	Does not require.
[3]	Green IT	-
[4]	Global software development	-
[5]	Smart grid	Requires knowledge in SPI
[6]	Software testing	Requires knowledge in SPI
[7]	-	-
[8]	-	Requires knowledge in SPI
[9]	SME enterprises	-
[10]	-	-
[11]	IT service	-
[12]	Acquisition	Does not require.
[13]	-	Does not require.
[14]	SME enterprises	-
[15]	Software testing	-
[16]	Lean	-

[17]	SME enterprises	Does not require.
[18]	IT service	-
[19]	SME enterprises	Does not require (only a basic understanding of CMMI)
[20]	SME enterprises and agile	-
[21]	SME enterprises	Requires knowledge of the methodology and assessment of the application process, and analysis of the data collected.
[22]	-	-
[23]	Services	-
[24]	SME enterprises and agile	-
[25]	SME enterprises	Requires at least a leader knowledge of both SPI and software development.
[26]	SME enterprises	-
[27]	-	Requires knowledge in SPI
[28]	Risk management	-
[29]	SME enterprises	Requires knowledge in SPI
[30]	Open source usability	Does not require.
[31]	-	Does not require knowledge in SPI, but in RE.
[32]	SME enterprises	Does not require.
[33]	SME enterprises	Requires knowledge in SPI

Effort to perform assessment: Half of the assessment methods analyzed (15 methods) address the efficiency of the self-assessment method. However, only four of them present the average time to perform an assessment (Table 23). Except for Abushama [32], the effort to perform the whole assessment is around 300 hours. Six articles present the average time for data collecting only. As shown in Table 24, data collection is expected to be a quick activity, performed in one or two work days. The other articles only comment that the developed method should require little effort to be applied [6]; [12];[16]; [23].

Table 23 - Effort to perform assessment

Article	Effort to perform self-assessment
[1]	Average of 385 hours
[2]	About 4-5 weeks (~160-200 hours)
[15]	Lasts 390 hours
[32]	About 24-34 hours

Table 24 - Effort for data collection

Article	Effort for data collection
[17]	Average of 1 hour per questionnaire.
[18]	Average of 4 hours per participant in software project activities. Assessors spent a total of 48 hours.
[21]	Average of 16 hours for small organizations (11 hours for the assessor advisor and 5 hours for the organization).
[28]	Focus group lasting approximately 2 hours.
[32]	Average of 2 hours for one interview (recommends interview a total of 1- 5 persons). An average of 12 hours for documents analysis.
[33]	2 days (~16 hours).

RQ3. What are the characteristics of process reference models?

Most of the assessment methods are based on an already consolidated process reference model (28 methods), while some are based on models being developed as part of the specific assessment method (5 methods). Seven methods are based on ISO standards, 12 methods are based on for CMM/CMMI, and 6 are based on CMM/CMMI-based models (PSP, TSP, TMM, TIM, MoProsoft) as summarized in Table 25. Most of the self-assessment methods do not focus on a specific software process area scope, except for [29] and [19], and Kuvaja [33] that include only processes considered important for any SMEs.

Table 25 - Characteristics of the reference models

Article	SPCMM based on	Scope
[1]	NMX-I-059/02-NYCE-2005 (Moprosoft)	All process areas
[2]	EIA/IS 731.2 Systems Engineering Capability Model Appraisal Method	All process areas
[3]	Green IT implementation (developed by the author)	All process areas
[4]	WAVE capability model	All areas: people, projects, unit and portfolio
[5]	Smart grid interoperability maturity model (developed by the author)	All areas Configuration & Evolution, Operation, Security & Safety
[6]	TMM	16 process areas
[7]	CMM	All process areas
[8]	Team Software Process (TSP)	All process areas
[9]	SW-CMM,TSP, PSP.	All process areas
[10]	EIA/IS-731	All process areas
[11]	ISO 20000 and IT Infrastructure Library (ITIL)	All process areas
[12]	CMMI-AM	All process areas
[13]	CMM	All process areas
[14]	CMMI	All process areas
[15]	ISO/IEC 29119 and TIM	Similar to ISO/IEC 29119 processes organization is conceptually close to organizational management process (OTP), planning and tracking to test management process (TMP) and TMCP, test cases to test plan process (TPP), test ware to STP and DTP, and reviews to TCP.
[16]	Lean Enterprise Model (LEM), the enterprise Transition-To-Lean (TTL) roadmap.	All process areas
[17]	ISO/IEC 29110	All process areas
[18]	ISO/IEC TR 20000-4:2010	All process areas

[19]	CMMI	Process areas from level 2 and 3
[20]	AMM model (developed by the author)	All process areas
[21]	PmCOMPETISOFT	All process areas
[22]	CMMI	All process areas
[23]	CMMI-SVC 1.3 e ARC	All process areas: Strategic Service Management (STSM), Service System Development (SSD), Service System Transition (SST), Service Delivery (SD), Incident Resolution and Prevention (IRP), Capacity and Availability Management (CAM), Service Continuity (SCON)
[24]	CMMI + SCRUM	Process areas Project Planning (PP), Project Monitoring and Control (PMC) and Integrated Project Management (IPM)
[25]	CMM	All process areas
[26]	TSE Model (developed by the author)	All process areas: Customer - Supplier, Engineering, Support, Management, Organisation, Legal
[27]	Trillium (developed by the author)	All process areas
[28]	IEC 80001-1	All process areas
[29]	ISO/IEC 12207	Basic dimension based on ISO/IEC 12207
[30]	OS-UMM (developed by the author)	All process areas
[31]	Requirements Engineering Reference Model (REM) and Requirements Capability Maturity Model (R-CMM)	All process areas
[32]	CMMI	All process areas
[33]	BOOTSTRAP 3.0	19 processes were considered to include the most important processes for any SMEs

RQ4. What are the characteristics of the measurement framework?

Analyzing this question, we observed that most AMs perform the process attribute rating and use measurement scales based on well-established standards such as ISO/IEC 15504 and CMMI. Almost all methods used some type of questionnaire to assist in data collection. In general, the questionnaires are not extensive (having less than 50 items) using closed answers questions. One third of them offers some kind of help, such as examples and explanations of the items.

Analyzing this question in more detail:

Process attribute rating: Most assessment methods do not define a specific activity for process attribute rating, however almost all of them present the way this should be done. Half of them use CMMI and ISO/IEC 15504 measurement scales and also calculate the assessment result the same way as suggested by these models. Four methods do not mention how their results are calculated and others propose different ways of calculating the maturity/capacity level, as shown in Table 26.

Measurement scale: Half of the assessment methods use consolidated assessment models scales such as CMM/CMMI (10 models) or ISO/IEC 15504 (5 models). Two methods use the scale of the TMM model and two the EIA 731 scale. Seven methods use different scales proposed specifically for the method. Six methods do not mention the use of measurement scales. Instead of presenting a level on a maturity or capability scale as a result of the assessment, some articles propose different way for providing feedback. For example, [29] presents a "profiled set of guidance and, therefore, an overall view of the capability of their software/IT practices". Daily & Dresner [26], in turn, presents opportunities for process improvement. Blanchette [12] provides as assessment result a graph representing the processes and their respectively scores, where the bars depict the range of scores for each process area. The self-assessment method proposed by Shrestha et al. [18] aims to provide information that can drive improvement of IT service processes, rather than providing a capability level. Wiegers et al. [13] is also more concerned with identifying appropriate improvement opportunities rather than maturity level ratings.

In total, 30 methods utilize questionnaires for performing the data collection:

Amount of questionnaire items: Most articles do not present the questionnaire itself (as indicated by "-" in Table 26) and only 16 inform the amount of questionnaire items. Among the articles that provide this information, most use instruments with up to 50 items (4 methods) or only one item for each attribute/process indicator (4 methods). Only 7 instruments have more than 50 items. In one instrument, the number of items depends on the input provided by the organization and another proposes a questionnaire for each key process area. Methods that do not use questionnaires are indicated by "N/A" at Table 26.

Item format/response scale: Among the methods that use questionnaires, 4 do not mention the format of the items. Among the articles that provide this information only 2 use open questions (of which 1 also used a questionnaire with closed questions), 22 methods use closed response items, of which 6 are affirmations and not questions. In general, closed response instruments have response scales with an average of 4 categories (ranging from 2 to 7 categories). Few questionnaires use a Likert scale (Likert, 1932), or a dichotomous scale (satisfying or not the respective item).

Example/explanation: Among the methods that use questionnaires and checklists, 11 provide information to aid in the interpretation of the questions/items as part of the questionnaire. Some methods provide explanations for each PA or just examples for the items, if necessary.

Table 26 presents the characteristics of the methods regarding their measurement framework and their data collection instruments, such as, process attribute rating, measurement scale, amount of items and response scale.

Table 26 - Characteristics of the measurement frameworks

Article	Process attribute rating	Measurement scale	Amount of items	Items format	Response scale	Provides example/explanations	Respondent
[1]	Adopting ISO/IEC 15504	ISO/IEC 15504 capability levels	-	Closed	7 point ordinal scale: Always, Usually, Sometimes, Rarely if ever, and Never. Don't Know and Not Apply. (Comments)	-	Project managers
[2]	Adopting EIA/IS 731.2	EIA/IS 731.2 capability levels	More than 600 items	Closed	3 point nominal scale: Yes, no, don't know, not apply.	Explanation for each question session.	Program/project management
[3]	Adopting ISO/IEC 15504	Readiness levels 0-50 Not ready 51-85 Ready 85-100 Prepared	-	-	-	-	Organization's manager
[4]	The levels of implementation are defined based on the number of evidence e=and weak points found.	Capability levels 2, 3, 4	One item for each model attribute (26 attributes).	-	-	-	Two groups of professionals, three with technical responsibilities and three with management responsibilities.

	"Fully Implemented" and "Largely Implemented" indicates that the practice was considered implemented.						
[5]	Adopting CMMI	CMMI maturity Levels	N/A	N/A	N/A	N/A	N/A
[6]	The ranking algorithm requires a rating of the maturity subgoals, then the maturity goals, and finally the maturity level.	TMM scale: Level 1: Initial Level 2: Phase definition Level 3: Integration Level 4: Management and Measurement Level 5:Optimization, defect, prevention and Quality control.	-	-	-	Instructions for use. Recommendations for questionnaire improvement. A glossary of testing terms	-
[7]	Adopting CMM (soft. tool)	CMM Levels	N/A	N/A	N/A	N/A	N/A
[8]	-	-	55 for managers 46 for testers	Open	-	-	Managers, developers and testers.

			61 for developers				
[9]	Adopting CMM	CMM Levels	124 items (mean of 6-7 per KPA)	Closed	4 point nominal scale: Yes, No, Does Not Apply, and Don't Know.	Instructions for use and explanation for each KPA and main concepts	-
[10]	Adopting EIA/IS-731	EIA/IS-731 capability Levels	-	Closed	3 point nominal scale: Yes, no, not apply. A "yes" answer required a brief comment that cited an example of the type of activity that was carried out that met the practice. A "NA" answer required a brief comment to justify why it was not applicable.	Examples for some items.	A mix of participants that gave sufficient coverage of the various engineering disciplines as well as a mix of practitioners and leaders.
[11]	Adopting ISO/IEC 15504	ISO/IEC 15504 capability levels	-	Closed	5 point ordinal scale: No, Partially, Largely, Fully and Not Applicable	-	Process stakeholders
[12]	-	-	30 affirmations. Questions	Closed	10 point ordinal scale. Score each statement from 1 to 10.	-	Program manager and deputy program manager, chief engineer, chief software engineer,

			covering all the process areas described in the CMMI-AM.		Statement could be positive or negative.		contracts specialist, business manager, and leads of integrated product teams.
[13]	-	-	-	Closed	7 point ordinal scale: Always, Usually, Sometimes, Rarely, Never, Don’t Know, Not Applicable	No. The assessors facilitate the questionnaire administration session, using standard slides to describe the intent of each KPA before the participants answer the questions for that KPA.	Organization representatives
[14]	A score of 80 or better, most likely indicates having achieved the maturity level 2.	CMMI maturity level	39 items.	Closed	5 point ordinal scale: definitely yes, usually, planned but not applied, not sure, definitely no	-	A responsible and knowledgeable person.
[15]	Compare the observations made	TIM levels; Level 0, Initial, Level 1, Baseline,	-	-	-	-	Software designer, test manager, manager.

	with a profile that indicates the maturity level.	Level 2, Cost-effectiveness, Level 3, Risk-lowering, Level 4, Optimizing.					
[16][16]	The capability level is decided upon by using lean indicators and capability-level descriptions for the specific practice.	Least capable (Level 1) to world class (Level 5).	54 affirmations (one for each practice)	Closed	2 point nominal scale: C (current) and D (desired).	Examples of indicators for each practice.	Leadership of the enterprise
[17]	Adopting ISO/IEC 15504	ISO/IEC 15504 capability levels	N/A	N/A	N/A	N/A	N/A
[18]	-	-	One for each indicator.	Closed	5 point ordinal scale: Not, Partially, Largely, Fully and Not Applicable	-	Process performers, process managers and other process stakeholders
[19]	Adopting CMMI	CMMI Levels	One sheet (questionnaire) for each of the seven process areas,	Statement	3 point ordinal scale: Yes, Partially, No	Each questionnaire begins with a short description of the process area.	-

			one sheet for the assessment details, and one sheet for the presentation of results.				
[20]	The answer of the questionnaires is used as input in a formula that calculates a percentage of achievement for each PAs. The degrees of achievement are: Fully Achieved, Largely Achieved, Partially Achieved and Not Achieved. KPA identifies the issues that must be addressed to achieve a	Maturity levels: Initial Explored, Defined, Improved. Sustained.	94 affirmations (mean of 7 items per PA)	Closed	4 point nominal scale: Yes, Partially, No, Not Applicable (N/A)	-	Developers, coach, testers with collaboration of on-site customer.

	maturity level.						
[21]	Adopting ISO/IEC 15504	ISO/IEC 15504 capability up to level 2.	-	Statement	Ordinal scale. Assigned a numeric value of 0 (never), 0.5 (sometimes) or 1 (always).	Each Process attribute begins with a short description.	-
[22]	Adopting CMMI. The software tool determines the final score for each capability level by calculating the mean value of all the responses for that level by all the respondents.	CMMI capability levels	36 items.	Open and closed	4 point nominal scale: Never, Sometimes, Almost & Always.	-	Key resource person of the company.
[23]	Adopting CMMI-SVC	CMMI-SVC 1.3 Levels	-	-	-	-	-

[24]	Practices are scores in strength, weak and not rated. The practices indicator determines the practice characteristic that in turn indicates the goal satisfaction that indicates the process area satisfaction.	-	Questionnaire generated based on the organization's input	Statement	3 point nominal scale: (a) Use, (b) Do Not Use, and (c) Not Available to Use	-	Each organization's project role provides evidences, related with its role.
[25]	Adopting CMM	CMM up to level 2.	One item for each activity from each PA.	Statement	2 point nominal scale: Checked, not checked	-	Senior Manager, Project Manager, SoftWare Manager, Software Engineering group, SQA group
[26]	-	-	Sets of questions for each practice that is going to be assessed.	Closed	6 point ordinal scale: always, usually, sometimes, rarely or never (or is not applicable) or 3 options: yes or no (or is not applicable).	Explanation about the process area. Examples for some items.	-

[27]	To achieve a level, an organization must satisfy a minimum of 90% of the criteria in each of the 8 Capability Areas at that level. Levels 3, 4 and 5 require the achievement of all lower levels	Capability levels: 1. Unstructured, 2 Repeatable and Project Oriented, 3 Defined and Process Oriented 4 Managed and Integrated, 5 Fully Integrated.	-	-	-	-	-
[28]	Adopting ISO/IEC 15504	ISO/IEC 15504 capability levels	-	-	-	-	Risk management stakeholders
[29]	Each question is awarded with score range 0-5, so the maximum score is “125” point and the minimum score is “0”.	Capability levels: 0 to 50% - Poor, 51% to 65% - Fair, 66% to 80% - Average & manageable, 81% to 90% - Established above 90% - Well established	125 items, 5 per process area	Closed	6 point ordinal scale: 0-Not attained at all, 1-poorly attained, 2-fairly attained, 3-attained averagely 4-Largely attained, 5-completely attained	-	-
[30]	Presents a formula to calculate	Maturity levels: 1: Preliminary	111 affirmations	Statement	5 point ordinal scale: Fulfilled,	-	Project managers or developers.

	the Usability Maturity Level based on the questionnaires responses.	2: Recognized 3: Defined 4: Streamlined 5: Institutionalized			Largely Fulfilled, Partially Fulfilled, Not Fulfilled and Not Applicable.		
[31]	-	-	-	Closed	4 point ordinal scale: Definitely yes, Rather yes, Rather no, no.	-	A company representative who have been involved in the RE activities.
[32]	KPA satisfaction level (not achieved, partially achieved, largely achieved and fully achieved) is calculated through a formula.	CMMI levels	A questionnaire for each KPA	Closed	4 point ordinal scale: Yes–Partially–No–Does Not Apply	-	Certified/experienced assessor by the SEI.
[33]	Adopting ISO/IEC 15504	ISO/IEC 15504 capability levels	-	-	-	-	-

RQ5. Have the assessment methods been developed and evaluated systematically?

Extracting information, we analyze the methodology used to develop the AMs, as well as the methodology used for their evaluation/validation.

Development methodology: Among the encountered articles, only 6 mention how the assessment methods were developed. Yet, 2 do not use a formal development methodology [16]; [25]. Karvonen et al. [16] states that *LESAT for Software* was developed from an adaptation of the LESAT (Lean Enterprise Self-Assessment Tool). Key concepts of the model were initially identified and were also valid in the context of software development. These concepts were adapted in order to use terminology more appropriate to the new context of use of the model. In addition, comments and examples relevant to the software development domain were included. Orci & Laryd [25] used an approach that starts with proposing a new model, followed by its application in case studies, measuring, analyzing, and validating it. The guidelines to implement the model were developed by common sense and based on the experience of the author.

On the other hand, [23], [18] and [11] used the formal DRS (Design Science Research) method (or an adaptation). DSR is a method for developing artifacts that consist of 7 steps: problem identification, objectives of solution, design & development, demonstration, evaluation and communication. In addition to DSR other development methodologies were used. Shrestha et al. [18] used the Goal-Question-Metric (GQM) approach to ensure that the measurement follows a transparent workflow of assessment activities, since this approach defines a measurement model for software metrics on three levels: goal (conceptual level), question (operational level) and metric (quantitative level). On the other hand, Kasurinen et al. [15] present only the methodology used to develop the assessment model, but not the assessment method, and, consequently, are not included in Table 27, which lists the methodologies used for the development of the self-assessment methods.

Table 27 - Articles reporting the development methodology used

Article	Method development methodology
[16]	"Adaptation of LESAT"
[25]	“Proposing a new model, followed by its application in case studies, measuring, analyzing, and validating it.”

[23]	DSR
[18]	DSR and GQM
[11]	DSR

Evaluation methodology: Although most articles do not present the development methodology, most of them (24) present some form of evaluation of the developed assessment method (Table 28). Among these, 7 articles do not present evaluation results related to the self-assessment method, but only regarding to the process reference model or the result of the assessment performed in some companies.

Seventeen articles report the evaluation of the assessment method. The most evaluated factors are efficiency, effectiveness and comfort (also represented by the term "usability" in some studies). Among these, 14 articles carried out case studies applying the methods in companies or with groups of professionals, who belong to the target audience of the assessment. The sample size varies between 1 and 24 companies; yet, nine studies were conducted with three companies or less. In addition to the conduction of case studies, Burnstein et al. [6] also conducted a review of the questionnaire by a panel of software engineering experts.

Expert panels have also used by other studies as the only way to evaluate the method. MacMahon et al. [28] carried out a review with 5 experts. Karvonen et al. [16], on the other hand, compare the elements of the method with an analogue one proposed by Ericsson. Göbel et al. [23] states that the method was tested iteratively, but do not inform how the tests were performed.

Table 28 - Articles that report the evaluation methodology

Research design	Articles	Evaluated characteristics	Evaluation context	Amount of data points
Case study	[11]	Effectiveness, Efficiency,	Customer contact management, spatial information	1 company, 9 participants

		Usefulness, Trust, Comfort	services for improved web mapping services, including mobile solutions, etc.	
	[2]	Effectiveness	Department of Energy field office	14 people
	[7]	-	-	1 company
	[13]	Cost, Efficiency, user satisfaction	Small projects. Average project team size was 12.	24 projects
	[15]	Accuracy and usability.	A large, internationally operating software company. A small-sized company, producing solutions for customer organizations. A large company producing software used for computer-assisted design. A medium-sized company, producing embedded software.	3 companies
	[18]	Transparency and efficiency (the degree of economy with which any assessment consumes resources, especially time and money).	Large public-sector IT organizations.	2 companies
	[21]	Reliability, construct, internal and external validity.	Small software organizations	8 companies
	[8]	Content validity.	-	1 company
	[29]	Effectiveness and comfort	Small software organization.	1 company

	[31]	Flexibility, efficiency, questions understandability, repeatability of results and meaningfulness of results.	-	10 industrial projects and 27 projects in a semi-industrial environment.
	[33]	Participants feedback	-	-
	[32]	The use of the PAM-SMEs to guide process improvement with orientation to business objectives. The applicability and suitability of the PAM-SMEs within SMEs.	ERP software packages, E-Learning software, E-Banking software	3 companies
	[30]	Reliability and validity	Open source projects	2 projects
Case study and Expert panel	[6]	-	-	2 experts (3 three development groups)
Expert panel	[28]	Utility, usability, scalability and generalizability, coverage of the requirements	-	5 experts
Comparison with other assessment method	[16]	Efficacy	-	-
Not informed	[23]	Functionality, usability, fit with the organization (the method is tailored for SME's within ITSM area), performance.	-	-

4.2.5 Discussion

A significant number of 33 software process self-assessment methods was encountered. The majority of them (24 methods) intents to assess the software processes in general (without focusing on a specified domain) and uses a version of the CMM/CMMI model or ISO/IEC 15504 as process reference model. Other self-assessment methods focus on a variety of specific domains such as IT service, Green IT, software testing, with a considerable number of 24 methods for specific contexts of use, including 15 self-assessment methods customized to SMEs and agile enterprises.

Almost all AMs use measurement scales (27 methods), most of them adopting the CMM/CMMI or ISO/IEC 15504 scale as is, calculating the maturity/capability level of the process in the same way as suggested by these models. Most AMs provide numerical results, such as a score or assign a level on a scale. However, considering the primary objective of self-assessment on process improvement, some methods rather focus exclusively on providing improvement feedback on the strengths and weaknesses of the assessed process [26]; [13]; [18]. Others propose to at least present the scores for each assessed process, in order to identify "weak" processes [12].

With respect to the AM process, most of the articles focus on discussing data collection and do not approach the other activities in detail. However, the lack of guidance on how to plan the assessment, validate the collected data, calculate and generate the results may result in an inaccurate assessment result and/or difficult the application of these methods in practice. Regarding the planning activity, 3 AMs suggest holding a meeting with stakeholders to plan the following activities [2]; [13]; [17]. Some AMs develop documents in which the assessment roles, as well as, which processes will be assessed are defined [6]; [5]; [4]; [3]. The most used technique for data collection is via questionnaires, followed by interviews. To perform this activity, the self-assessment methods often provide software tools [3]; [18]; [21]; [26]; Malanga et al., 2015; [4]; [6]; [22]; [33]; [31] or electronic spreadsheets [2]; [19].The approach of asking questions directly in an digital environment is considered a faster and more efficient data collection method compared with other methods such as interviews . Also, the fact that assessment activities can be automated, such as the generation of results, offers an efficiency gain, which can be translated into significant cost savings [18]. The questionnaires are answered by different participants' roles (such as tester, developer, manager, area leader). Some questionnaires, on the other hand, are used to guide interviews and workshops [8]; [14]; [23]; [25]; [31]; Kuvaja et al., [33]. Considering the concern to carry out assessments as efficiently as possible, these data collection instruments are kept succinct in general. Several AMs use only 1 item for each practice, for example, with a total of up to 50 items; only 7 instruments are composed of more than 50 items. Some AMs are concerned in collecting data using more than one technique, for example, combining the use of questionnaires and interviews [2]; [3]; [21]; [22]; [6], questionnaires with participants' discussion [6] or interviews with documents analysis [31]. Such a triangulation may be important in order to draw valid conclusions.

A very small number of articles (8) mention the validation of the collected data activity. Among them, 3 perform validation by comparing the data collected through different methods [8]; [21]; [31] and 3 used software to analyze respondents' responses [4]; [18] and [2]. These situations might make sense in a context in which the organization is not seeking a certification, but rather an informal assessment in order to understand the capacity of its processes. On the other hand, it is essential that the assessment result be reliable and effective in order to correctly guide software process improvement actions. In addition, the fact that data collection does not result in a large amount of data may justify the scarcity of works that present this activity. Regarding the process attribute rating, half of the methods perform the process attribute rating following standards, such as ISO/IEC 15504 and CMM/CMMI. Half of the articles mention the reporting of the assessment results, although most of them do not report how it is performed. Overall, the articles do not mention how the results are reported. Three methods provide a template to guide the development of a document report, 3 suggest the presentation of results to stakeholders [4]; [28]; [31] and one suggests a feedback session [17].

A self-assessment method should not require specific knowledge of process assessment standards or process reference models as being conducted by internal staff, which might not have specialized knowledge nor experience in software process assessment (VON WANGENHEIM; C. ANACLETO, 2006). Some of the analyzed AMs do not indicate this requirement [2]; [5]; [12]; [13]; [17]; [19]; [30]; [31]; [32]. However, almost all the other AMs do not require the necessity of the assessment participants to have specific subject knowledge. This situation allows the applicability of these AMs in companies with few resources, either without staff specialized in SPI or not being able to invest in SPI training for their staff. Working around this situation, about half of the AMs provide examples or tips on how to use the data collection instrument and how to analyze the collected data. Shrestha et al. [11] states that in order to clarify survey questions, relevant examples should be provided when necessary. The lack of aid may also make it difficult to use AMs based on standards such as CMMI or ISO/IEC 15504, as they require a certain level of SPI knowledge in order to be used. In general, we observed that, although, the lack of SPI knowledge and experience of the internal assessors poses a significant threat to the validity of self-assessment, this issue is not studied in detail by the reported methods.

The great amount of effort required to carry out a process assessment is often mentioned as a disincentive to companies that wish to have insight into their processes. In this respect, one of the main advantages in carrying out a self-assessment is the minimization of time and effort. Nevertheless, few articles mention this benefit. In general, the articles that evaluated the performance of the AMs in relation to its effort report that assessment lasts between 160 and 390 hours [1]; [2]; [15], except for Abushama's method [32], that requires only 2 days. Other articles, on the other hand, presented only the effort to perform data collection. Some considered only the time for responding the data collection instrument, others include the time for data analysis. As a result, the reported efforts are not comparable. Yet, in general, the total duration for data collection does not vary by more than 3 working days (22 hours), except for [18]. The need for leaner data collection instruments makes their development more complex, as it requires the evaluation of the relevance and necessity of each item (KAZI; KHALID, 2012). Regarding the format of the questionnaire items, we observed that most of them use closed questions or affirmations. Using open-ended questions, on the other hand, might increase the burden on work, since the provision of qualitative data involves a demanding process (SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009). They also increase the complexity for the respondents, often non-experts on the subject being assessed, which may not know exactly how to answer the questions. For these reasons open-ended questions should not be seen as an ‘easy option’ (SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009). Closed-ended questions in contrast ask the respondents to make choices among a set of answers for a given question. The response can be mutually exclusive or may select more than one option. For measuring dichotomous variables closed-ended questions are preferred because possible answers can be easily precoded (KAZI; KHALID, 2012). In general, closed question instruments have response scales with an average of 4 ordinal points (ranging from 2 to 7 points). Few questionnaires used a Likert scale. Despite the impact on the complexity of responding and the validity of the result the format of the response scale may have, none of the articles justified the scale used nor discussed on whether the respondents would be able to distinguish between the categories used, for example, between partially or largely achieved.

As assessment methods should be valid, reliable, and cost efficient, they need to be developed systematically (SIMONSSON; JOHNSON; WIJKSTRÖM, 2007; VON WANGENHEIM et al., 2010), and in a similar way, in order to acquire data properly, data collection tools need to be designed in such a way that they can measure things of interest (KAZI; KHALID, 2012). However, we observed that most AMs analyzed do not report information on their development. Only 3 articles present the use of systematic methods for the development of the AM, using DSR and GQM. However, even these articles do not present a systematic method for the development of the measurement framework (in the case of articles that were not based on existing measurements) or of the development of the data collection instruments. However, systematic development is important in order to assure the validity of the obtained results, especially, when mostly using questionnaires that need to be designed carefully in order to comprehensively cover the object to be measured, while at the same time minimizing the number of items. Considering also the possible inexperience of the respondents with respect to SPI, it also becomes essential to take great care on the wording of the items (KAZI; KHALID, 2012; SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009).

Although 17 articles have reported some form of evaluation of the AM, most of them do so using small samples and/or without using a systematic method. Again, especially, as most AMs use of some kind of questionnaire, it is important to assure their reliability and construct validity (KAZI; KHALID, 2012; SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009).

However, we observed that only four studies evaluated the questionnaire/checklist used. Amaral et al. [8] evaluated the questionnaire with respect to content validity, [30] evaluated reliability and construct validity. Rapp et al. [31] evaluated the reliability of the questionnaire and [21] evaluated reliability, construct, internal and external validity. Thus, the lack of scientific rigor of the validation of the majority of the proposed methods may leave their validity questionable.

4.3 THREATS TO VALIDITY

As with all research, there exist several threats to the validity of the results presented (ZHOU et al., 2016). We, therefore, identified potential threats and applied mitigation strategies in order to minimize their impact on our research. A main risk in a systematic mapping is the omission of relevant studies. To mitigate this risk, the search string was carefully constructed. Different strings (containing the core concepts and their synonyms) were tested in order to identify the one that returned the most relevant results. The risk of excluding relevant studies is further mitigated by the use of Google Scholar, a repository that covers a large number of repositories. Based on the suggestion of Haddaway et al. [34] that the first 200-300 results on Google Scholar should be analyzed, we reviewed the first 200 results of both searches performed in this repository.

Threats to study selection and data extraction have been mitigated through a detailed definition of the inclusion/exclusion criteria. We defined and documented a rigid protocol for the study selection. In the first mapping study, it was also checked, if all articles reported by (JOKELA et al., 2006) (our control paper) and that meet our inclusion criteria, were found and included in our review.

Recognizing the lack of consistent use of terminology, the information of the encountered articles has been carefully extracted and revised interpreting the presented models in relation to the theory presented in the background section. As this paper presents an interpretative analysis, the findings are based on the author's subjective interpretations.

4.4 CONCLUSION

As result of the first systematic mapping we encountered a total of 15 UCMMs. We observed that most of the models are in conformance or are based on consolidated SPCMMs such as CMMI and ISO/IEC 15504. On the other hand, only 5 UCMMs define a proper process reference model, indicating in general a lack of the definition of requirements to the usability engineering process. We also observed that most UCMMs do not provide sufficient support to be applied in practice. Thus, it is necessary to seek other sources and/or to make an arrangement of different models and methods. Furthermore, it seems that with very few exceptions, no systematic methods have been used to develop the UCMMs, and only a very small number of models have been validated in non-experimental ways based on very small samples. This reflects a lack of methodological rigor, which leaves the validity of the proposed models questionable. Despite the importance to customize UCMMs for specifics context of use especially considering recent trends in software development, we found only 3 models for specific contexts, including two models integrating usability engineering and agile principles.

In the second systematic mapping we encountered a considerable amount of 33 software process self-assessment methods. Most methods are based on traditional and consolidated process reference models and measurement frameworks, such as CMMI and ISO/IEC 15504. On the other hand, in relation to these traditional models, the assessment process, in general, is simplified. Half of the AMs are customized for SEs, which may be due to their characteristics, requiring less costly, more efficient and less bureaucratic assessment methods. Most of the AMs perform data collection through the application of (closed-question) questionnaires. Benefits of the data collection technique are a reduction of time and effort and the ease to analyze the data as well as the possibility of automation of the result generation. However, although several studies mention concerns about the duration and effort of the assessment, only very few analyzed these factors of the proposed AM. Another shortcoming observed is that with respect to the assessment process, most methods focus exclusively on the data collection not detailing other stages. This may further complicate the application of the proposed AMs in practice. We also observed that with few exceptions the methods for developing these AMs have not been reported. And, only very few articles report the validation of the methods and/or the data collection instruments. Another issue observed is that most proposed AMs do not discuss the mitigation of significant threats to the validity of the results obtained through self-assessment due to the potential inexperience of the assessors and the lack of triangulation of data. Therefore, it seems to be questionable to which regard the majority of the proposed AMs can be used to perform valid self-assessments.

The literature review indicates the scarcity of research on usability process assessment for SEs, since no maturity/capability usability model was found that focused on SEs, and none self-assessment method was found that focused on usability process.

5 REQUIREMENTS TO A SELF-ASSESSMENT METHOD FOR ASSESSING THE USABILITY PROCESS IN SE

In order to develop an effective and efficient method for self-assessing the usability processes on SE, a set of requirements is elicited to guide the development of an method to self-assess the capability of the usability process in SEs (Table 29). The proposed requirements are based on:

- Requirements specific for:
  - Software process assessment methods,
  - Usability process assessment methods,
  - Process self-assessment;
- Needs and characteristics of SE found in literature.

In addition, it is desirable for UPCASE to provide an assessment process in accordance with ISO/IEC 29110-3, as an internationally recognized series of technical reports on conducting process assessment in SE. The ISO/IEC 29110-3 assessment process is based on ISO/IEC 15504, a series that provides a framework to perform assessment of processes. Thus, UPCASE includes requirements for the assessment method as a whole, and requirements specifically for each of the method's elements, as defined by ISO/IEC 15504 (ISO/IEC, 2004): process assessment, measurement framework and process reference model.

Table 29 - Requirements to a self-assessment method for assessing the usability process in small enterprises

No.	Requirement	Element	Source(s)
1	The method should allow a fast-internal assessment.	Method	(M. Mirna et al., 2012) (Pino et al., 2010) (Hering et al., 2015) (Sánchez-Gordón et al., 2016)
2	The method should allow getting a snapshot of actual processes.	Method	(M. Mirna et al., 2012)

3	The method should be of low cost.	Method	(M. Mirna et al., 2012) (Sánchez-Gordón et al., 2016) (Anacleto et al., 2004)(Caffery, 2007) (Pino et al., 2010) (Sulayman et al., 2012)
4	The method should provide the necessary tools (including tools for (partial) automation, eliminating laborious manual work and extensive documentation).	Method	(M. Mirna et al., 2012) (Anacleto et al., 2004)(Caffery, 2007) Pino et al., 2010) (Sulayman et al., 2012)
5	The method should be based on already established SPI standards that are widely recognized.	Method	(Mishra et al., 2009) (Pino et al., 2008) (Anacleto et al., 2004)(Caffery, 2007) (Pino et al., 2010)
6	The method should not require staff to have prior SPI experience, specific software engineering knowledge nor require the involvement of external experts.	Method	(Mishra et al., 2009) (Anacleto et al., 2004)(Caffery, 2007) (Pino et al., 2010) (ISO/IEC 29110-4)
7	The method should provide accesses to a detailed definition of the process reference model and the assessment model, with descriptions of process purpose, process outcomes provided by the PRM and capability levels and process attributes. The rating scale needs to be supported by a comprehensive set of indicators of process performance.	Method	(Anacleto et al., 2004)(Caffery, 2007) (Pino et al., 2010) ISO/IEC 29110-3-1
8	The method should be public available.	Method	(Anacleto et al., 2004)(Caffery, 2007) (Pino et al., 2010)
9	The method should support the identification of improvement suggestions.	Method	(Anacleto et al., 2004)(Caffery, 2007) (Pino et al., 2010)
10	The process assessment should guide the activities that need to be performed in an assessment. It should provide a clear definition of roles and their responsibilities and a detailed description of the assessment process, with recommendations that are easy to understand.	Assessment Process	(M. Mirna et al., 2012) (Anacleto et al., 2004)(Caffery, 2007) Pino et al., 2010) (Sulayman et al., 2012) (Fuchs et al.,2012) (ISO/IEC 29110-4)

11	The process assessment should require few resources.	Assessment Process	(Sulayman et al., 2012) (Sánchez-Gordón et al., 2016)
12	The process assessment should consider the views of the team while indicating what needs to be improved.	Assessment Process	(Mishra et al., 2009)
13	The process assessment and measurement framework should facilitate self-assessment.	Assessment Process / Measurement framework	ISO/IEC29110-3-1 (Mishra et al., 2009)
14	The measurement framework should provide a basis for use in process improvement and capability determination.	Measurement framework	ISO/IEC29110-3-1
15	The measurement framework should take into account the context in which the assessed process is implemented.	Measurement framework	ISO/IEC29110-3-1
16	The measurement framework should contain a process capability scale.	Measurement framework	ISO/IEC29110-3-1
17	The measurement framework should be applicable across all application domains mainly for very small entities.	Measurement framework	ISO/IEC29110-3-1
18	The measurement framework should provide an objective benchmark between organizations.	Measurement framework	ISO/IEC29110-3-1
19	PRM processes should be light, easily implementable, representing well-focused life cycle profiles, not requiring processes that do not make sense.	Process Reference Model	(Sulayman et al., 2012) (Laporte et al., 2008) (Mirna et al., 2012)
20	PRM processes should avoid complex nomenclature, concepts and practices (SEs have little awareness of usability concepts and usability standards. Their definition of usability may be limited and inconsistent).	Process Reference Model	(O'Connor, 2009) (Renzi et al., 2015) (Hering et al., 2015) (Fuchs et al.,2012) (ISO/IEC 29110-4)
21	PRM processes should be strongly human oriented and emphasizing communication (most of communication is performed face to face).	Process Reference Model	(O'Connor et al., 2014) (ISO/IEC 29110-4)
22	PRM processes should focus on the Engineering Process group (SE are less interested in the Management Process Group and the Support Process Group.	Process Reference Model	(Pino et al., 2009) (ISO/IEC 29110-4)

23	PRM processes should aim at involving user in the usability lifecycle. (Understanding users is considered important and greater integration with user interferes positively in usability capability).	Process Reference Model	(Hokkanen et al., 2016) (Scheiber et al., 2012)
24	PRM processes should not impose rigorous and inflexible methods and practices.	Process Reference Model	(Hokkanen et al., 2016)
25	Practices should be simple (nowadays SE are generally immature in relation to the use of usability processes.)	Process Reference Model	(Scheiber et al., 2012)
26	PRM processes should be flexible and allow iteration.	Process Reference Model	(Hering et al., 2015)

The requirements presented in Table 29 were used to guide the development of UPCASE, a method for self-assessing the capability of the usability process in SEs, described in the following chapter.

6 A METHOD FOR SELF-ASSESSING THE CAPABILITY OF THE USABILITY PROCESS OF SE

This chapter presents UPCASE, a method for self-assessing the capability of the usability process in SEs, in agreement with ISO 29110-3 (ISO/IEC, 2016b). UPCASE is presented in the following sections, according to its structure: process reference model, measurement framework and assessment process.

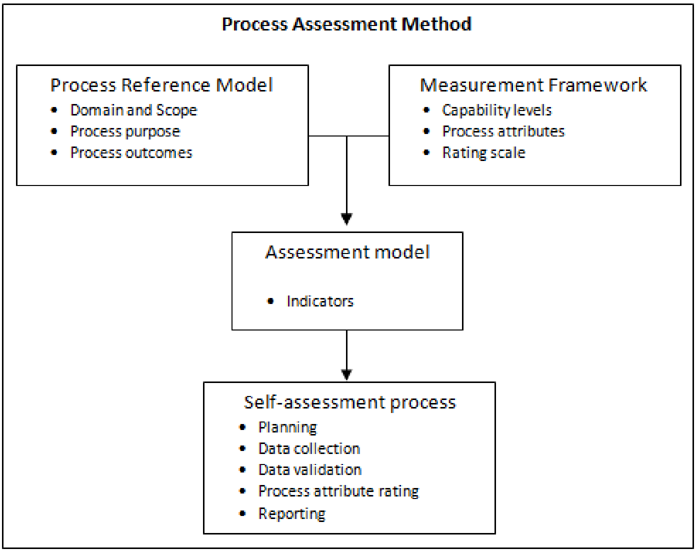
In the context of this work, it is understood that a method is a systematic approach to achieve a specific objective or result and that describes the characteristics of an ordered process or procedure used in the engineering of a product or in the performance of a service (ISO/IEC, 1990). Based on this definition, it is proposed a method that aims at providing a systematic support for the self-assessment of usability processes in small software organizations.

The method, called UPCSE, is based on the technical report ISO/IEC TR 29110-3-1 Assessment Guide, which is part of the series ISO/IEC 29110 Systems and software engineering - Lifecycle profiles for very small entities (VSEs) (ISO/IEC, 2016b). This series aims at guiding small software organizations in the development and/or maintenance of their products, as well as in the management of their projects. The assessment guide for ISO/IEC 29110-3 intents to assess the process capability based on a two-dimensional assessment model, containing a process dimension and a quality dimension of the process. The choice of using ISO/IEC TR 29110-3 was based on the finding that most assessment methods are based on international standards such as CMMI and ISO/IEC 15504 (LACERDA; VON WANGENHEIM, 2017). In this context, ISO/IEC TR 29110-3-1 was considered more appropriate as it is specific for SEs.

The scope of UPCASE method covers the first objective of ISO/IEC TR 29110-3-1, focusing on "Assessing the process capability based on a two-dimensional evaluation model, containing a process dimension and a quality dimension of the process". As UPCASE intends to assess the usability processes of SE of all profiles of the generic profile group, thus, it does not include the assessment on whether an organization meets a desired profile.

In this context, the original ISO/IEC TR 29110-1 elements have been simplified, so that UPCASE uses only the elements necessary to achieve the first objective of the standard. Figure 15 shows the basic elements of UPCASE, adapted from ISO/IEC TR 29110-3-1.

Figure 15 - Elements of the UPCASE assessment method (adapted from ISO/IEC 29110-3-1 (2015))



6.1 UPCASE MEASUREMENT FRAMEWORK

The measurement framework provides a schema to be used to characterize the capability of a process in relation to a reference model. The measurement framework of the UPCASE

Method is based on ISO/IEC 29110 and is composed of three elements: capability levels, process attributes and a rating scale.

Capability levels are used to determine the process capability. Capability levels group the process attributes and define an ordinal scale of capability that is applicable across all processes. Considering the typical capability of SE, two capability levels are considered by UPCASE, as shown in Table 30.

Table 30 Process capability level description

Capability level	Description
Level 0: Incomplete process	The process is not implemented or fails to achieve its process purpose. At this level there is little or no evidence of any systematic achievement of the process purpose.
Level 1: Performed process	The implemented process achieves its process purpose.

(Source: ISO/IEC TR 29110)

Process attributes (PA) are measurable characteristics of the process capability that are applicable to any process. Capability level 0 has no process attributes. At this level the process is not implemented, or do not achieve its purpose. Capability level 1 contains only one process attribute **PA 1 Performance**. The process Performance attribute is a measure of the extent to which the process purpose is achieved. As result of full achievement of this attribute: The process achieves its defined outcomes.

The **rating scale** is a defined ordinal scale of measurement used to measure the extent of achievement of a process attribute, adapted from ISO/IEC 29110. Originally ISO/IEC 29110 uses four rating scales (including L - Largelly achieved). In UPCASE method this rating was removed in order to simplify the rating process, thus making it easier for assessors to better discern between ratings, as extensive scale, when using use by lay respondents may be confusing (SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009). For example, would lay respondents be able to distinguish between the categories used, for example, between partially or largely achieved. The UPCASE's rating scale is composed of the ratings.

N- Not achieved 0 to 15% achievement.
P- Partially achieved >16% to 85% achievement.
F- Fully achieved >86% to 100% achievement.

The process profile (to determine the process capability level) is generated as defined by ISO/IEC 15504 (and used by ISO/IEC TR 29110). The percentage of the process attribute achievement (PPAA) is calculated based on the process indicators rating, as follows:

$$PPAA = (\sum process\ indicators\ rating \ / \ n^{o}\ indicators * 2)*100.$$

6.2 UPCASE PROCESS REFERENCE MODEL

The purpose of the Process Reference Model (PRM) is to define the usability processes and sub-process, describing them in terms of purpose and outcomes. As the other elements of UPCASE, the PRM is also compliant with the structural definition of PRMs given by ISO/IEC 29110-3 and specifies the following components (ISO/IEC, 2016a):

Process purpose: The process purpose consists of a single paragraph stating the purpose of the process describing the overall objectives of performing the process. It is supplemented by an enumeration of the principal process outcomes associated with that process.

Process outcomes: A process outcome is an observable result of the successful implementation of a process. Process outcomes are normally worded as descriptive statements.

There exists several usability PRMs (LACERDA; VON WANGENHEIM, 2017). The most comprehensive ones, which cover a wide range of human activities related to software development, are described by ISO/IEC 18152 and ISO/IEC 18529 (LACERDA; VON WANGENHEIM, 2017). The PRM described by ISO/IEC 18152 presents a wider scope and may be used in larger or more complex projects, covering the whole range of Human Centered Design activities (HCD) involved in systems engineering. In addition to the HCD technical processes, it also defines 13 processes related to planning and management. This standard is compliant with ISO/IEC 15504 and provides detailed guidance on how to perform process assessment. However, no research results on its use in practice are reported. On the other hand, the PRM described by ISO/IEC 18529 is more focused on technical processes, including only one additional one dedicated to the planning and management of HCD. This standard is also compliant with ISO/IEC 15504 and provides detailed guidance to perform process assessments. Considering that SE generally need simpler processes (REQ 19) and, typically, do not develop large and complex projects (MAJCHROWSKI et al., 2016), it was chosen to base UPCASE's PRM on ISO/IEC 18529, since it focuses on technical process (in agreement with REQ 22). As result, the UPCASE PRM includes four categories as defined in ISO/IEC 18529, as shown in Table 29.

Taking into consideration the requirements identified in Section 2 further adaptations have been done to customize the PRM to the specific needs and characteristics of SE. Considering REQ 22, 3 processes from ISO/IEC 18529 were not included in UPCASE's PRM: HCD 1, HCD2 and HCD 7. HCD 1 and HCD 2 were excluded, as they deal mainly with management and business strategy practices, not focusing on technical practices. On the other hand, the HCD 7 process was excluded, as it deals with the implementation and support of the system practices that are generally not responsibility of SE. Table 31 details the selection of processes in accordance with the identified requirements in Section 2, presenting a justification for each excluded process (marked in red).

Table 31 - UPCASE Usability process

Process (based on ISO/IEC 18529)		Justification for exclusion
HCD 1	Ensure HCD content in systems strategy	REQ 22 (Process should focus on engineering process)
HCD 2	Plan and manage the HCD process	REQ 22 (Process should focus on engineering process)
HCD 3 (UP 1)	Specify stakeholder and organisational requirements	
HCD 4 (UP 2)	Understand and specify the context of use	
HCD 5 (UP 3)	Produce design solutions	
HCD 6 (UP 4)	Evaluate designs against requirements	
HCD 7	Introduce and operate the system	REQ 22 (Process should focus on engineering process)

For each of the 4 processes selected from ISO/IEC 18529 a purpose and outcomes was defined, as presented in Table 32 - Table 35. Again, some outcomes have been excluded in accordance to the identified requirements in relation to the characteristics specific to SE and the process defined by ISO/IEC 29110-4. ISO/IEC 29110-4 defines minimum software engineering processes for SEs that develop a single application by a single work team. Thus, for example, as ISO/IEC 29110-4 does not consider acquisition as a mandatory software engineering process in SE, inferring that it also should not be mandatory with respect to the usability process.

In total, 6 outcomes from ISO/IEC 18529 were excluded. These outcomes have been removed as they are typically not in the scope of the processes of small software organizations (such as the responsibility of installing and operating the system) or require more advanced usability knowledge than the staff of small organizations usually have. The justification for excluding the outcomes is presented in the column Justification (marked in red) in Table 32 – Table 35, presenting separately the outcomes with respect to each of the processes.

Table 32 - UP1 Specify stakeholder and organizational requirements

UP1	Specify stakeholder and organizational requirements	Justification
Purpose	To establish the requirements of the organization and other interested parties for the system. This process takes full account of the needs, competencies and working environment of each relevant stakeholder in the system.	
Outcomes	Required performance of the new system regarding its operational and functional objectives.	
	Relevant statutory or legislative usability requirements, depending on the system domain.	
	Co-operation and communication between users and other relevant parties	
	The users’ jobs (including the allocation of tasks, users’ comfort, safety, health and motivation)	This outcome overlaps with the outcomes "Definition of the characteristics of the intended users" and "Definition and characterization of the tasks the users are to perform" from UP2.

	Task performance of the user when supported by the system	
	Work design, and social practices and structure	This outcome overlaps with the outcome "Definition and characterization of the tasks the users are to perform" from UP2.
	Feasibility of operation and maintenance	REQ 22 (Process should focus on engineering process)
	Objectives for the operation and/or use of the software and hardware components of the system.	

Table 33 - UP2 Understand and specify the context of use

UP2	Understand and specify the context of use	Justification
Purpose	To identify, clarify and record the characteristics of the stakeholders, their tasks and the social and physical environment in which the system will operate.	
Outcomes	Definition of the characteristics of the intended users.	
	Definition and characterization of the tasks the users are to perform.	
	Definition and characterization of the social and environment in which the system is used.	
	Definition and characterization of the technical environment in which the system is used.	
	The use of context analysis results in requirements to the interface design.	
	The context of use is available and used at all relevant points in the system development.	

Table 34 - UP3 Produce design solutions

UP3	Produce design solutions	Justification

Purpose	To create potential design solutions by drawing on established state-of-the-art practice, the experience and knowledge of the users and the results of the context of use analysis.	
Outcomes	Results of socio-technical context of use analysis are considered in the design.	
	User characteristics and needs will be taken into account in the purchasing of system components.	There are no mandatory or optional requirements related to the Acquisition processes in the Basic Profile of the ISO/IEC 29110 series; Nor are they expected to be defined in the next ISO/IEC 29110 series profiles (SEBRAE, 2013).
	Results of the user analysis are taken into account in the design of the system.	
	Existing knowledge of best practice from socio-technical systems engineering, ergonomics, psychology.	REQ 19 (Little awareness on usability) Small organizations typically does not have HR with expertise in usability
	Cognitive science and other relevant disciplines will be integrated into the system.	REQ 19 (Little awareness on usability) Small organizations typically does not have HR with expertise in usability
	Communication between stakeholders is improved because the design decisions are more explicit.	
	The development team is able to explore several design concepts before they settle on one.	
	Feedback from end users and other stakeholders is incorporated in the design early in the development process.	
	It is possible to evaluate several iterations of a design and alternative designs.	

	The user's tasks are analyzed in relation to their, navigation, hierarchy and information architecture.	
	The design of all the user-related components of the system is specified, in terms of "look and feel".	
	The interface between the user and the software, hardware and organizational components of the system are designed.	
	User training and support will be developed.	ISO/IEC 29110-4 (Small enterprises generally are not responsible for the management, operation, integration and installation of the system.)

Table 35 - UP4 Evaluate designs against requirements

UP4	**Evaluate designs against requirements**	**Justification**
Purpose	To collect feedback on the developing design. This feedback will be collected from end users and other representative sources.	
Outcomes	Feedback is provided to improve the design.	
	There is an assessment of whether stakeholder and organizational usability objectives have been achieved or not.	
	Long-term use of the system will be monitored	ISO/IEC 29110-4 (Small enterprises generally are not responsible for the management, operation, integration and installation of the system.)
	Potential problems and scope for improvements in: the technology, supporting material and social or physical environment.	

	Which design option best fits the functional and stakeholder and organizational requirements.	
	Feedback and further requirements from the users.	This outcome overlaps with the outcome "Feedback is provided to improve the design" from UP4.
	How well the system meets its organizational goals.	This outcome overlaps with the outcome "There is an assessment of whether stakeholder and organizational usability objectives have been achieved or not" from UP4.
	Guarantee that a particular design meets the human-centred requirements.	
	Conformity to international, national and/or statutory requirements, depending the system domain.	

6.3 UPCASE PROCESS ASSESSMENT MODEL

The UPCASE Process Assessment Model (PAM) is compliant with ISO/IEC 29110-3 (ISO/IEC, 2016a) and contains the basis for collecting evidence and rating process capability. It contains two dimensions: The Process Dimension, which defines the set of processes that will be assessed (they are defined in the PRM) and the Capability Dimension, which defines the capabilities related to each process capability level and each process attribute. UPCASE PAM contains a scope, indicators and a mapping for a Process Reference Model and a Measurement Framework (ISO/IEC, 2016a):

Scope: determines which processes will be assessed (at least one of the PRM) and which scale levels will be used to assess them.

Indicators: provide guidance on the interpretation of the process purposes and outcomes as defined in the PRM. They are sources of objective evidence used to support the assessors' judgment in rating process attributes and demonstrate the achievement of the process attributes within a capability level. There are two types of process performance indicators: Base Practice (BP) and Work Product (WP) indicators. Evidence of performance of the base practices and the presence of work products provide objective evidence of the achievement of the purpose of the process. A **base practice** is an activity that addresses the purpose of a particular process. A set of base practices is associated with each process in the process dimension. The base practices are described at an abstract level, identifying “what” should be done without specifying “how”. The performance of a process produces **work products** that are identifiable and usable in achieving the purpose of the process.

Again, in accordance to the identified requirements and the process defined by ISO/IEC 29110-4, the practices of ISO/IEC 18529 have been excluded or adapted to meet the requirements of the self-assessment method in this specific context. The adaptation of the practices aims at meeting requirements 6, 19, 20 and 25 identified in Section 2. Therefore, the practices are written in such a way that staff without SPI or usability knowledge can understand them and, thus, eliminating the need for external experts. To accomplish this, the use of complex nomenclature and concepts and jargons from the usability domain was avoided. Furthermore, for each of the work products an example is provided, illustrating the expected result. Aiming at a "light" process, practices that overlap each other or that seem to complex in the context of small organizations were removed.

The customization of the practices is presented in Table 36. In total, eight practices were excluded ISO/IEC 18529 (marked in red). The 16 practices selected from ISO/IEC 18529 were rewritten with the aim of making their understanding easier for assessors, who are not experts in usability processes (REQ 20).

Table 36 - UPCASE process practices

		ISO/IEC 18529 practices	**Customized UPCASE practices**	**Justification for exclusion**
UP1	1	Clarify system goals	Identify system purpose.	
	-	Analyze stakeholders	--	This practice overlaps with "Identify and document significant user attributes" practice. In addition, the basic profile of ISO/IEC 29110-4 does not have any practice related to the analysis of the roles of each stakeholder group besides the users. Characterization of the users is covered through UP2-Practice 6.
	-	Assess H&S risk	--	This practice has been removed in order to keep the process simple (REQ 19), and because it is contained in practice 6.
	2	Define system	Define system performance and behavior requirements desired by the user.	
	-	Generate requirements	--	This practice is performed in the context of the software engineering process (ISO/IEC 12207). Its output, however, should be used as input in the usability process, being necessary for the execution of practices 3 and 4.
	3	Set quality in use objectives	Define usability requirements.	
UP2	4	Identify and document user's tasks	Identify and describe the user's tasks of the system.	
	5	Identify and document	Identify user characteristics.	

		significant user attributes		
	6	Identify and document organizational environment	Identify social environment characteristics.	
	7	Identify and document technical environment	Identify device characteristics.	
	8	Identify and document physical environment	Identify physical environment characteristics.	
	-	Allocate functions	--	This practice has been removed in order to keep the process simple (REQ 19), and because it is contained in practice 10.
UP3	9	Produce composite task model	Analyze user's tasks.	
	10	Explore system design	Develop and analyze design options during interface development.	
	11	Use existing knowledge to develop design solutions	Develop design solutions using existing knowledge.	
	12	Specify system and use	Specify all user-related elements of the system.	
	13	Develop prototypes	Prototype all user-related elements of the system.	
	-	Develop user training	--	ISO/IEC 29110-4 (Small enterprises generally are not responsible for the management, operation, integration and installation of the system.)
	-	Develop user support	--	ISO/IEC 29110-4 (Small enterprises generally are not responsible for the management,

				operation, integration and installation of the system.)
UP4	14	Specify and validate context of evaluation	Prepare prototype/system evaluation.	
	-	Evaluate early prototypes in order to define the requirements for the system	--	This practice has been removed in order to keep the process simple (REQ 19), and because it might be contained in practice 15.
	15	Evaluate prototypes and in order to improve the design	Evaluate prototypes and system to find usability problems.	
	16	Evaluate the system in order to check that the stakeholder and organizational requirements have been met	Evaluate system against usability requirements	
	-	Evaluate the system in order to check that the required practice has been followed	Evaluate system to find usability problems.	This practice has been removed in order to keep the process simple (REQ 19), and because it might be contained in practice 15.
	-	Evaluate the system in use in order to ensure that it continues to meet organizational and user needs	--	ISO/IEC 29110-4 (Small enterprises generally are not responsible for the management, operation, integration and installation of the system)

In order to facilitate the understanding of its practices, ISO/IEC 18529 provides a description for each of them (Table 37). These descriptions have been adapted in order to attend REQ 20, helping assessors to better understand the UPCASE practices.

In order to better assist in the correct implementation of the usability process assessment, as well as to help the assessment team to verify if the usability practices of the SE are in accordance with the UPCASE assessment method, the UPCASE method contains other artifacts, such as:

Description of the assessment process: describes how each step of the assessment process should be performed. The description of the assessment process is presented in APPENDIX A .

Assessment questionnaire: the assessment questionnaire should be used during the assessment as a “roadmap”, which allows the assessment team to judge each practice of the UPCASE PRM. Therefore, the questionnaire has an item to assess each of the 16 practices of the usability process.

To support the judgment of the performance of each practice of the usability process, the questionnaire presents an indicator for each of them (APPENDIX D - Table 55, column Indicators). The indicators objectively demonstrated characteristics of the practices of the assessed process. Each indicator is written in such a way that it is easy to understand, even by person with a poor knowledge on usability process, in order to meet REQ 11.

The measurement scale used in the assessment questionnaire is defined in section 6.1 UPCASE MEASUREMENT FRAMEWORK.

Suggestions of techniques and work-products: a set of examples of work products and techniques for each indicator (each questionnaire item). The work products are based on ISO/IEC 18529.

Glossary: In order to facilitate the accurate understanding of all the elements of the Assessment Method, a glossary is provided with the definition of the more technical wording in the area of usability (APPENDIX E). In addition to the definition, when necessary the glossary also presents examples of the explained concepts.

Examples of work products: To make it easier for the assessor to correctly identify practices within his/her organization, when necessary, examples of work products for each practice were elaborated based on literature (KLEIN, 2013; MAYHEW, 1999; ROGERS; SHARP; PREECE, 2011; USABILITY NET, 2006) and ISO/IEC 18529. ISO/IEC 18529 contains a list of the typical work products originated from the implementation of the usability processes. These work products were adapted to suit the characteristics of SE, i.e. work products considered too complex for this type of organization were omitted and others were adapted in order to facilitate their understanding by lay people in usability. This adaptation resulted in a compact list of work products that typically are expected to be generated even by SE, (Table 54).

Table 37 - Description of UPCASE practices and example of techniques

N Practice	UPCASE Practices	ISO/IEC 18529 Practices Description	Customized Practices Description	Example of Techniques
1	Identify system purpose	Describe the objectives which the user or user organisation wants to achieve through use of the system.	Identify and describe the purpose of the system, this is, the objective(s) that the user wants to achieve using the system.	Survey, brainstorming, interview, observation.
2	Define system performance and behavior requirements desired by the user.	Set and agree the required behavior and performance of the system in terms of the total experience of the relevant stakeholders and/or the user organisation with the system. The total experience covers each aspect of a relevant stakeholder's relationship with the system and its context of use from its	Identify the stakeholder's requirements regarding the behavior and performance of the system. The requirements cover each aspect of the system related to its use and its interface in a context of use.	Survey, brainstorming, interview, observation.

		commissioning to its decommissioning.		
3	Define usability requirements.	Generate and agree on measurable criteria for the required quality in use of the system	Define an explicitly statement for each usability requirements, regarding its effectiveness, efficiency and user satisfaction based on the context of use analysis. The statements should be measurable objectives.	Benchmarking with concurrent systems, synchronic analyzes, formal work analyses.
4	Identify and describe the user's tasks of the system	Describe the activities which users perform to achieve system goals.	Describe the tasks the users need perform in the system in order to achieve their goals.	Survey, interview, observation, formal work analysis, brainstorming with user's task, user's tasks modeling.
5	Identify user characteristics	Describe the relevant characteristics of the end-users of the system. This will include knowledge, language, physical capabilities, level of experience with job tasks and with relevant systems equipment, motivations in using the system, priorities, etc.	Identify relevant characteristics of the users, such as knowledge about the system domain, degree of literacy, physical capabilities, level of experience with the tasks and with the device he will use to interact with the system, motivations in using the system, etc	Survey, interview, observation.
6	Identify social environment characteristics	Describe the relevant social and organizational milieu, management structure, communications and organizational practices, etc.	Identify relevant social and organizational milieu, management structure, communications and organizational practices in the	Survey, observation, interview.

			environment in which the system will be used.	
7	Identify device characteristics	Describe the relevant characteristics of any equipment to be used in the system or the context of use. Particular attention should be paid to the equipment with which the users will directly interact.	Identify relevant characteristics of the device with which the users will directly interact, such as memory and process capacity, ways of input and output data, screen size, etc.	Review platform documentation.
8	Identify physical environment characteristics	Describe the location, workplace equipment and ambient conditions and the implications for design. For example, lighting, noise levels, vibration, etc.	Identify relevant characteristics of the location, workplace equipment and ambient conditions and its implications for the system design, such as lighting, noise levels, vibration, heat, hazards, dimensions of working and living space.	Survey, observation, interview.
9	Analyze user's tasks	Develop a feasible model of the user’s new tasks from existing knowledge of best practice, the requirements, context of use, allocation of function and design constraints for the system.	Analyze the user's tasks in terms of alternative navigation pathways and flowcharts and identifying the main system screens and constraints.	Conceptual model design, wireframes development, navigation definition, task hierarchy analysis, information architecture definition, card sorting.
10	Develop and analyze design options during interface development	Generate and analyze a range of design options for each aspect of the system related to its use and its effect on stakeholders.	Analyze a range of design options for each aspect of the system related to its use and its effect on stakeholders, such as definition of system controls, location and format of display	Wireframe development, sketches development, storyboarding, use case analysis.

			components, use of colors, terminology, fonts, and wording of messages.	
11	Develop design solutions using existing knowledge	Apply relevant human science information to the design of the system. Include the stakeholder and organizational requirements, context of use, international standards, legislative requirements, existing patents, good practice, style guides and project standards etc. in the design.	Apply existing usability knowledge, such as stakeholder requirements, information about the context of use, international standards, usability good practice and style guides to the design of the system and is used to select the appropriate alternatives of design.	--
12	Specify all user-related elements of the system	Produce a design for the user-related components of the system. Produce description(s) of how the system will be used. Change design in the light of feedback from evaluations.	Specify the design of all the user-related components of the system. This specification is a description of how the components and the system will be used, such as the kind of systems controls will be used, location and format of display components, colors, terminology, fonts, wording of messages).	Wireframe and sketch development, storyboarding.
13	Prototype all user-related elements of the system	Make design solution(s) more concrete using simulations, models, mock-ups etc. Develop simulation or trial implementation of key aspects of the system for the purposes of testing with users or user representatives.	Refine design through the development of high-fidelity prototypes of all aspect of the system related to its use and its interface.	Prototype development.

14	Prepare prototype/system evaluation	Describe and check the conditions under which a system is tested or otherwise evaluated. Describe the relationship, and especially discrepancies, between the context of evaluation and the context of use.	Prepare and defined all arrangements necessary to evaluate the prototype or the system, such as definition of which evaluation method will be used, who will be the assessor, place, scripts, questionnaires, cameras and etc.	Prepare usability evaluation necessary materials, select participants.
15	Evaluate prototypes and system to find usability problems	Collect user input on the quality in use of the developing system. Present the results to the design team(s) in the most appropriate format.	Prototypes are evaluated against usability knowledge, style guides, standards, guidelines in order to find usability problems and verify if the required good practices have been followed.	Usability test with users, heuristic evaluation, cognitive walkthrough, key level stroke model evaluation.
16	Evaluate system against usability requirements	Test the developing or final system to ensure that it meets the requirements of the users, the tasks and the environment, as defined in its specification.	System is evaluated to ensure that it meets the requirements of the users, the tasks and the environment, as defined in its specification.	Usability test with users.

6.4 UPCASE ASSESSMENT PROCESS

The purpose of the assessment process is to systematically guide the process assessment activities. The assessment process of the UPCASE method is based on the assessment process defined by ISO/IEC TR29110 and is composed by three components:

Phase: is a set of activities grouped in steps, presenting a logical or structured sequence.

Activity: is a stage of the process assessment that produces visible changes in the state of the product. The activity may have inputs, outputs, intermediate results, generically called work products. The activity implements procedures, rules and objectives to transform a product.

Work products: are the inputs and outputs of a process activity. They can be produced and consumed throughout the process and may have long life cycles, being created, accessed and modified.

6.4.1 Work products of the UPCASE Method

The UPCASE method uses an adaptation of the work products defined by ISO/IEC TR 29110:

Input: The process assessment input defines the basic elements necessary to carry out the process assessment:

- **Purpose**: defines the reason for performing the assessment.
- **Scope**: defines the boundaries of the assessment, provided as part of the assessment input, encompassing the organizational limits of the assessment, the processes to be included, and the context within which the processes operate.
- **Constraints**: are restrictions placed on the freedom of choice of the assessment team regarding the conduct

of the assessment and the use of the assessment outputs.

- **Identities**: the identity of the Process Assessment Model and the identity of the Process Reference Model used in the assessment.
- **Approaches**: establishes the assessment approach: self-assessment or external assessment.
- **Assessor competencies**: are the criteria for competence of the assessor who is responsible by the assessment.
- **Questionnaire template**: defines indicators, examples of techniques and work products that support the judgment of the capability of an implemented process, as well as a form to register the assessment results. UPCASE's questionnaire template v0.2 is presented in APPENDIX F .

Any assessment input information need to be defined by the assessor, as it is provided by UPCASE.

Output: The process assessment output consists of information about the performance of the assessment and its results, such as:

- **Date**: on which the assessment was carried out.
- **Assessment input**: the information used as input in the assessment process.
- **Identification of evidence**: the completed UPCASE assessment questionnaire, the document that presents examples of work products that show evidence of the accomplishment of the indicators.
- **Assessment process used**: identifies which assessment process was used to perform the process assessment.
- **Process rating**: contains the set of the process attribute ratings for each assessed process. Each

attribute rating represents a judgment by the assessor regarding the extent to which the attribute is achieved.

All output information is documented in UPCASE's process assessment report phase.

Roles and Responsibilities:

Another input for conducting a process assessment is the definition of roles and responsibilities. UPCASE defines three main roles for conducting an assessment:

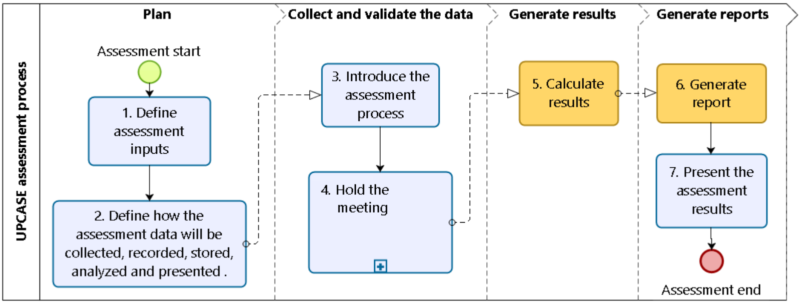
- **Sponsor:** the representative of the enterprise that is being assessed.
- **Moderator:** responsible for ensuring that the assessment is performed in accordance with the UPCASE assessment process. This role is also responsible for conducting the assessment meeting, acting as a moderator.
- **Assessors that compose the assessment team:** along with the moderator are responsible for performing the assessment.

Considering the context of SE, in which a person may be responsible for various roles and responsibilities, when performing a process assessment, the same professional can assume the role of sponsor, moderator and assessor.

6.4.2 Phases and Activities of the UPCASE Method

Based on ISO/IEC 29110-3, the UPCASE assessment process is composed by four phases: Plan the assessment, Collect and validate the data, Generate results and Report the assessment, as presented in Figure 16. Activities that may be automated by UPCASE Tool are presented in yellow. The definition of the techniques and tools to perform each of these phases is based on good practices identified in literature.

Figure 16 - Assessment process



6.4.2.1 Phase 1 - Plan the assessment

The assessment phase "Plan" contains at least to two activities. When using UPCASE, some of the activities typically needed to plan the assessment no longer need to be performed by the assessment team, but only revised, as shows Table 38. The column "Outputs defined by UPCASE" presents the outputs of the activity that are provided by the UPCASE method, and therefore do not need to be generated by the assessment team. The column " Outputs defined by the assessment team" presents the outputs of the activity that must be developed by the team.

Table 38 Plan the assessment – activities

Activities	Outputs defined by UPCASE	Outputs defined by the assessment team
1. Define the assessment inputs.	**Purpose**: assess de capability of the usability process. **Scope**: Includes the usability process 1, 2, 3 and 4 defined in UPCASE's PRM. **Approach**: Self-assessment. **Process Assessment Model**: UPCASE's PAM. **Assessment activities**: Defined by UPCASE. **Constraints** (the quantity and type of objective evidence to be examined in the assessment): One evidence (example of work product	Revised by the assessor

	or technique) that the practice is performed. **Criteria for competence of the assessor:** Criteria are defined by UPCASE, as shown in Table 39. **Roles and responsibilities** are defined by UPCASE, as shown in Table 39.	
		Constraints: a) Availability of key resources and b) The maximum duration of the assessment. **Identities of moderator and participants.**
2. Define how the assessment data will be collected, recorded, stored, analyzed and presented with reference to the assessment tool.	As presented in Section 6.4 UPCASE assessment process.	Revised by the assessor

Table 39 UPCASE's Roles and Responsibilities

UPCASE Roles	Responsibilities	Role description
Sponsor	a) verify that the individual who is to take responsibility for conformity of the assessment is a	Some leadership position of the organization that

	competent assessor (following the definition of "moderator" as given by the Inputs of UPCASE); b) ensure that resources are made available to conduct the assessment; c) ensure that the assessment team has access to the relevant resources.	realizes the need to assess the usability process, such as: - Project manager, - Development leader, - UI Design leader.
Moderator	a) confirm the sponsor's commitment to proceed with the assessment; b) ensure that the assessment is conducted in accordance with the assessment method; c) ensure that participants in the assessment are briefed on the purpose, scope and approach of the assessment; d) ensure that all members of the assessment team have knowledge and skills appropriate to their roles; e) ensure that all members of the assessment team have access to appropriate documented guidance on how to perform the defined assessment activities; f) ensure that all assessors are able to participate in the assessment meeting. g) carry out assigned activities associated with the assessment, e.g. detailed planning, data collection &validation and reporting;	Should be chosen by the sponsor. Preferably should be the professional with more knowledge about process assessment or usability.
Assessors	a) provide examples of work products and techniques as evidence of the execution of the process. b) rate the processes attributes.	Assessors may be any employee who perform activities related to the usability process, such as:

		- Designers, - System analysts, - Testers.

Activities of Phase 1 - Plan the assessment:

The activities of this phase can be carried out during a meeting with some members of the organization, who are responsible for the usability process, as defined in the input Roles and Responsibilities. During this meeting the following items should be defined:

1) Resources and schedule: all resources necessary for carrying out the assessment, and any constraints, should be defined, including:

The availability of key resources: such as location of the meeting, computer and projector to present the assessment tools, deck of cards to for consensus finding, members of the organization who have knowledge about the process being assessed.

The maximum duration of the assessment: the maximum time that the enterprise may expend to perform the assessment meeting, for example, a meeting of 2 hours or an entire afternoon.

2) Roles and responsibilities: It is necessary to define which professionals will participate in the assessment and which role each one will take.

The identity of the moderator: it should be defined who among the assessor will be the moderator.

The identity of the assessors: it should be defined who among the employees will participate in the assessment meetings.

6.4.2.2 Phase 2 - Collect and validate the data

The assessment phase "Collect and validate the data" contains four activities (Table 40) that need to be performed by the assessment team.

Table 40 Collect and validate the data - activities

Activities	Outputs defined by the assessment team
Brief the assessment team	1) Briefing the assessment
Collect evidence of process capability for each process within the scope.	2) Assessment meeting with assessment poker.
Record and maintain references to the evidence	2.2.7 The moderator completes the response of the items in the questionnaire.
Validate the data	2.2.5 The moderator requests examples of work products that demonstrate the accomplishment of the practice indicator.

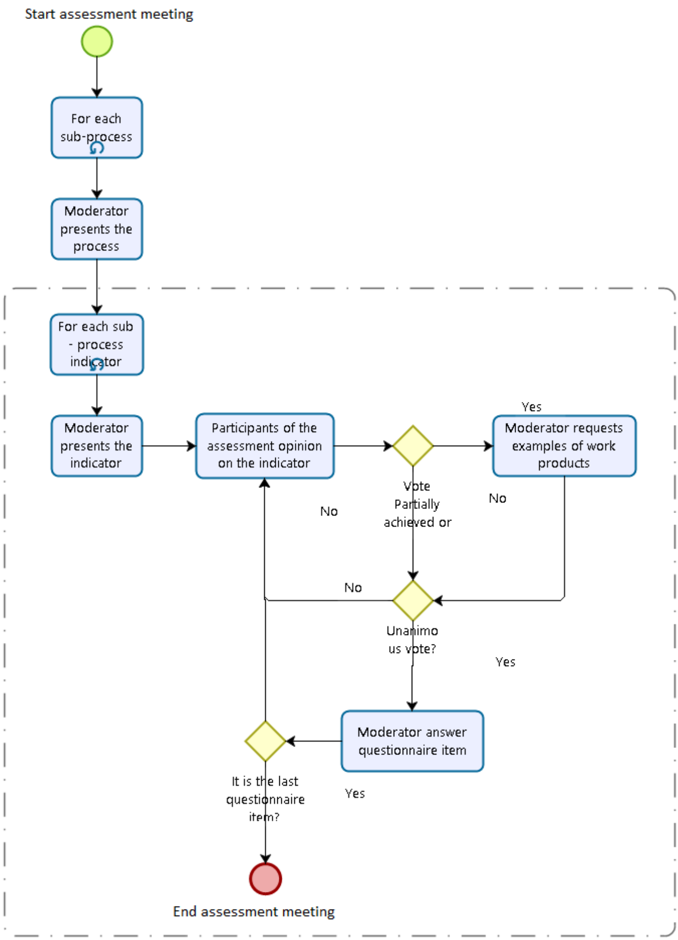
Activities of Phase 2 - Data collection and validation:

The four activities that must be performed by the assessor are carried out during a briefing and a focus group meeting are:

1) Briefing the assessment: The moderator presents the purpose of the process assessment that will be carry out. She/he presents the focus group methodology and the assessment poker techniques, as well as the inputs and outputs of the assessment. At the end of the briefing the moderator should ensure that the assessment team understood the proposed assessment approach, the inputs and outputs, as well as, how to use UPCASE. UPCASE provides a script for the briefing in APPENDIX A .

2) Assessment meeting with assessment poker: The activity "Data collection and validation" is performed during an assessment meeting, as defined in the planning phase. The assessment meeting is divided in sub-activities, as defined in Figure 17. To conduct the assessment meeting, UPCASE provides a questionnaire that contains the items that should be assessed for each usability sub-process, as well as the description of each of them with examples of work products and techniques. The questionnaire is presented in APPENDIX F and in English and Brazilian Portuguese. UPCASE also provides a deck to be used in the assessment poker (available in APPENDIX B).

Figure 17 - Assessment meeting activities



1. Start assessment meeting (Assessment poker):

In an Assessment Poker, similarly to Planning Poker, the meting the moderator reads a story (or indicator). There is a discussion clarifying the story as necessary. Each participant chooses a card that represents their estimate. Once all participants have chosen their estimate, they turn over all the cards. If there is agreement, no discussion is necessary, and the estimate is recorded. If there is disagreement in the estimates, the team discusses their different estimates and tries to get to consensus (GRENNING, 2002). Each step of assessment poker is detailed below:

1.1 The moderator presents the purpose of the process assessment and how it will carried out.

2. Perform the assessment using the Assessment poker with the UPCASE questionnaire:

2.1 The moderator presents the usability sub-process, along with its purpose and outcomes.

2.2 For each questionnaire item:

2.2.1 The moderator reads the questionnaire item, along with its description, work products and techniques.

2.2.2 If any concept is not understood, the moderator reads the concept in the glossary.

2.2.3 After the questionnaire item is clearly understood by all participants, the moderator asks them to give their opinion about the item that was read with respect to its achievement in the organization.

2.2.4 Participants express their opinion through one of the assessment poker cards, "Not achieved", "Partially achieved" or "Fully achieved". All participants show the selected card at the same moment in order to prevent any influence on the individual opinions. The participant must inform "Not achieved" if he considers that the organization does not perform the practice, "Partially achieved", if he considers that the organization accomplishes the practice, but it is not always performed; and "Fully achieved", if he agrees that the organized perform the practice described in the item.

2.2.5 If any participant opines with "Partially achieved" or "Fully achieved", the moderator should request an example of work product that demonstrate the achievement of the indicator. If an example is provided, then the indicator should be rated according to the consensual opinion of the participants, if no example is provided then the indicator is considered unachieved.

2.2.6 As long as the participants' opinions on the item do not achieve a unanimous result, step 2.2.4 is repeated.

2.2.7 If consensus is achieved, the moderator completes the response of the item in the UPCASE questionnaire.

Why hold an assessment meeting, in the form of a focus group, and use "Assessment" poker for collecting data?

Most self-assessment methods use questionnaires for data collection, only a few use interviews, workshops or focus group meetings (LACERDA; VON WANGENHEIM, 2017). Despite the wide adoption of questionnaires as a method for data collection, it does not come without shortcomings. The use of questionnaires may lead to unreliable responses (if the subject misinterprets a question) or/and to a lack of completeness. Furthermore, questionnaires are typically answered individually which makes it difficult to interact with the respondents in order to ask for further explanations on a given answer, and/or to confirm that respondent understood the questions correctly.

On the other hand, the use of interviews for collecting data may solve these issues. Interviews, however, also present some disadvantages, such as high cost (requiring people to conduct the interviews) and the collection of a small sample of data (as the size of the sample is limited to the size of interviewing staff) (SAUNDERS M. N. K.; LEWIS P.; THORNHILL, 2009). Furthermore, if conducted individually, inconsistencies and conflicts between the collected responses may have to be resolved later on.

In this context, for UPCASE's assessment process was chosen to use the Focus Group method for data collection. Focus groups are group interview that focus on a particular issue, product, service or topic and encompass the need for interactive discussion amongst participants (CARSON et al., 2001). The persons chosen to participate in the focus group meeting need to have certain characteristics in common that relate to the topic being discussed. During the meeting, the participants are encouraged to discuss and share their points of view (KRUEGER; CASEY, 2000). Focus groups have some advantages over other methods for collecting data:

- It allows the discussion of each one of the questions among a group of people. In this way it is possible to reach a consensual conclusion about each question discussed, and thus, increase the accuracy of the responses collected.
- Eliminates the need for a later step to validate participants' response, as this occurs during the meeting.
- It is more efficient as it allows obtaining the opinion of a larger number of people in a shorter period of time.
- Prevents the moderator from having to interview the same person again to confirm information provided by another respondent.
- A single participant may not know how to answer questions about all processes, so the focus group avoids wasting time in interviewing him about issues he does not know how to answer.
- Enable to resolve conflicting responses already during the meeting obtaining a consensus.

On the other hand, the realization of a focus group meeting is not without shortcomings, as group interactions may lead to a highly productive discussion as interviewees respond to the questions and evaluate points made by the group. Thus, there may emerge a group effect with certain participants trying to dominate the interview whilst others may feel inhibited. This may result in some participants publicly agreeing with the views of others, whilst privately disagreeing. As a consequence, a reported consensus may, be an idea that nobody really endorses or disagrees with (STOKES; BERGIN, 2006). To mitigate the risk that some participants do not opine or be embarrassed to give their true opinion in front of colleagues or boss, UPCASE proposes to perform the focus group in conjunction with an adaptation of the Planning poker, a consensus-based technique for estimating effort. Planning poker (GRENNING, 2002) is a popular technique based on the Wideband Delphi (BOEHM, 2006), which aims at increasing the precision of estimating the effort of tasks. This technique was created to solve two common problems during the estimation process in project management: estimates were taking a long time, and not the whole team in the decision-making (GRENNING, 2002). Thus, performing an assessment poker during the assessment process is expected to increase the accuracy of the responses by relying not only on the opinion of the moderator but also on the experiences of the whole assessment team. Furthermore, it allows the assessor team to reach a consensus about the achievement of each indicator.

6.4.2.3 Phase 3 - Generate results

The assessment phase "Generate Results" contains one activity (Table 41), “Derivation of assessment results”, that may be performed by the assessment team or may be automatically generated by the UPCASE Tool.

Table 41 Generate results - activities

Activities	**Outputs defined by UPCASE**	**Outputs defined by the assessment team**
1)Derivation of the assessment results	X	X

Activity of Phase 3 - Generate results activities:

The activity of the "Generate results" phase must be carried after the assessment meeting, by any of the participants. In this phase the following activity are expected to be performed:

1) Derivation of assessment results: The derivation of the process assessment results is based on the answers of the completed UPCASE questionnaire. The result of the assessment determines the organization usability process profile. The process profile contains a set of attribute ratings for each assessed sub-process, as well as the capability level of the usability process as a whole. APPENDIX H presents an example of a hypothetical assessment results derivation. The capability level and the profile of the usability process are derived from the process attribute ratings:

1) Calculate usability sub-process percentage of achievement (USPA) based on the indicator ratings:

Formula 1:

$$\textbf{USPA} = ((\sum sub - process\ indicator\ ratings)/\text{n}^{\text{o}}\text{indicators} * 2) * 100.$$

1.2) Calculate sub-process attribute capability rating based on its achievement percentage, as defined in ISO/IEC TR 29110.

Table 42 - Attribute rating according to the achievement percentage

Attribute rating	Achievement percentage
N- Not achieved	>0 and <=15% achievement
P- Partially achieved	>16% to 85% achievement
F- Fully achieved	>86% to 100% achievement

1.3) Calculate usability process percentage of achievement (UPPA) based on the indicators ratings: The usability process capability level is calculated based on its sub-process capability rating:

Formula 2:

$$\textbf{UPPA} = ((\sum usability\ process\ indicators\ ratings)/\text{n}^{\text{o}}\text{indicators} * 2) * 100.$$

Usability process attribute rating is calculated based on its percentage of achievement, as defined in step 1.2. An UPPA calculation example is presented in APPENDIX H

6.4.2.4 Phase 4 – Report assessment results

The assessment phase "Report assessment results" contains one activity (Table 43), "Report assessment results", that should be performed by the assessment team using the template provided by the UPCASE.

Table 43- Report the assessment results - activities

Activities	Outputs defined by UPCASE	Outputs defined by the assessment team
1)Report the assessment results	UPCASE defines a report template.	a) Prepare the assessment report. b) Present the assessment results to the sponsor.

Activity from Phase 4 - Report the assessment results activities:

1) Report the assessment results: In this activity a report with the information of the assessment is developed, as well as the results are presented for the interested parties:

1.1) Prepare the assessment report: The assessment findings are summarized, highlighting the process profile, observed strengths and weaknesses and potential improvement actions. The report can be generated automatically using the online UPCASE tool. Otherwise, the report may be developed by any member of the organization that participated in the assessment. In this case, the report template presented in APPENDIX G can be used.

The report shall contain at least the following information:

- Assessment date;
- Assessment moderator;
- Assessment participants;
- Usability process rating;

- Rating of each usability sub-process;
- Legend of the ratings;
- Strengths found in the assessment (e.g.: practices of the usability process that the organization already performs and that must continue being carried out);
- Points that can be improved (e.g.: practices of the usability process that the organization does not yet perform or that it does not perform consistently or adequately).

1.2) Present the assessment results to the sponsor: The assessment results are presented to the interested parties (e.g. management, practitioners, etc.) during a meeting. The presentation is typically made by the lead assessor. All the information contained in the assessment report can be presented. The emphasis of the presentation should not be on the process rating, but rather on the items identified as opportunities for improvement. During the meeting improvements actions may be discussed based on the assessment results.

6.5 UPCASE TOOL

To help the realization of the usability process self-assessment, it is interesting to use a tool that offers functionalities that support the assessment process. The use of a software tool may make the assessment more efficient, especially the phases of data collection and generation of the assessment results. Consequently, the use of an assessment tool can further reduce the overall assessment time, such as minimizing the possibility of human errors in generating the results. In order to facilitate the realization of self-assessment using the UPCASE method, a web–based application, called UPCASE Tool, was developed.

1) Based on the specification of the UPCASE process, we identified requirements for the UPCASE Tool intended to support the application of the method.
2) The functional and non-functional requirements defined are shown in Table 44.

Table 44 - UPCASE Tool requirements

Number	**Functional requirement**
1	Should present the assessment guide
2	Should allow the online responding of the assessment questionnaire
3	Should generate the assessment results calculating the usability process capability rating.
4	Should present a glossary
5	Should present examples of work-products
6	Should be available in English and Portuguese
7	Should present the terms of use and information about the tool
-	**Non-functional requirement**
8	Should follow the visual identity of GQS

3) The UPCASE Tool has one use case: Perform assessment:
   1. User accesses the UPCASE Tool
   2. UPCASE Tool opens on-screen assessment guide
   3. User starts assessing the usability process
   4. UPCASE Tool presents the screens with the assessment questionnaire.
   5. User responds and submits the assessment questionnaire.
   6. UPCASE Tool displays the result of the assessment.

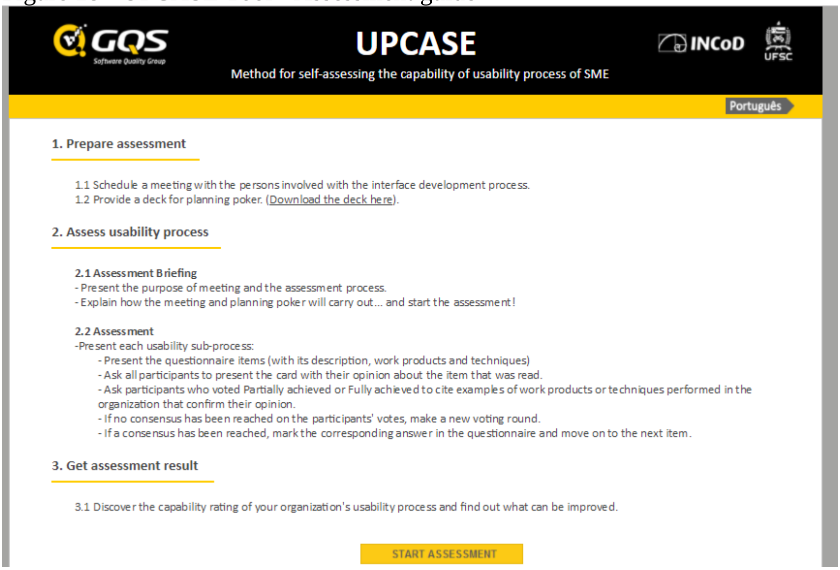
   4.a User views glossary

4.b User views examples
6.a UPCASE displays an error message of unfilled items.

The different functionalities of UPCASE are described in the order of its use case:

Assessment Guide: The UPCASE Tool home page, presented in step "2. UPCASE Tool opens on-screen assessment guide" contains the assessment guide, Figure 18. The assessment guide summarizes the self-assessment process, listing each of the assessment phases, and what activity should be performed by the user in each one of them. In addition, this page provides a link, so the user can download the deck of cards for the assessment poker.

Figure 18 - UPCASE Tool - Assessment guide

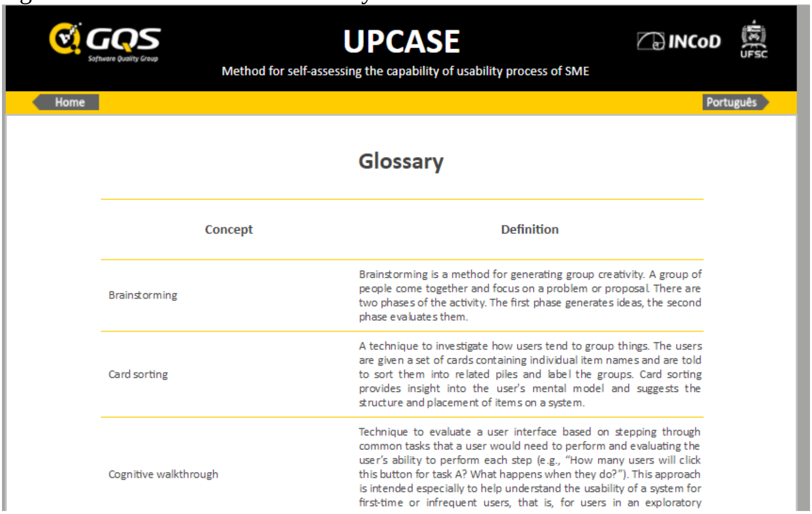


Assessment Questionnaire

When the assessment starts, “3. User starts assessing the usability process”, is presented the assessment questionnaire, Figure 21. These screens contain:

- Instructions on how the questionnaire should be filled.
- Access to a glossary that presents a list of the main concepts related to usability process assessment on SEs, Figure 19. The glossary presented in the UPCASE Tool is available on APPENDIX E
- Access to a page with examples of work products to elucidate doubts regarding concepts used in the questionnaire, Figure 20. The Examples of Work-Products screen presents a list with examples of the main work-products that may be generated when performing the activities of usability process. The examples presented in the UPCASE Tool are available on APPENDIX C .
- Assessment questionnaire, contains 16 items grouped in 4 sub-processes. Each questionnaire item presents a description, examples of work-products that can be obtained from the practice, and examples of techniques that can be used to carry out the practices. For each item the following response options are displayed: “Fully achieved”, “Partially achieved” or “Not achieved”.

Figure 19- UPCASE Tool - Glossary



UPCASE

Method for self-assessing the capability of usability process of SME

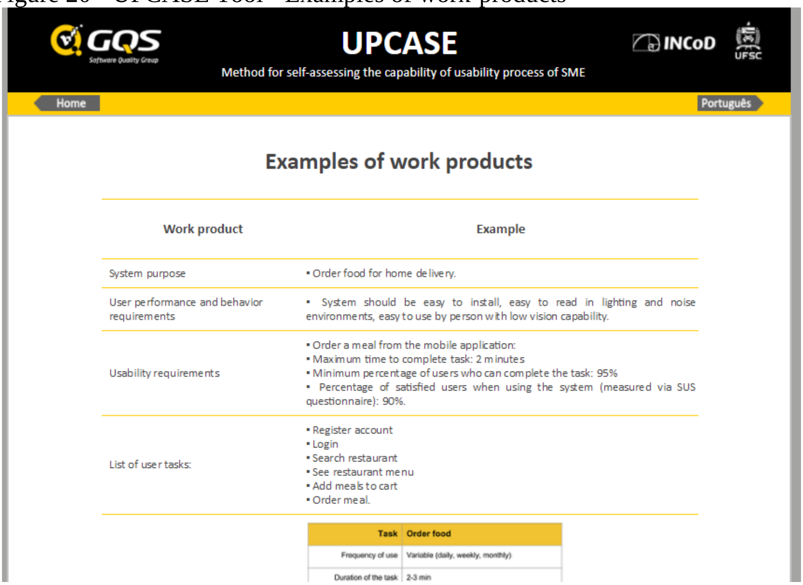
Home

Português

Glossary

Concept	Definition
Brainstorming	Brainstorming is a method for generating group creativity. A group of people come together and focus on a problem or proposal. There are two phases of the activity. The first phase generates ideas, the second phase evaluates them.
Card sorting	A technique to investigate how users tend to group things. The users are given a set of cards containing individual item names and are told to sort them into related piles and label the groups. Card sorting provides insight into the user's mental model and suggests the structure and placement of items on a system.
Cognitive walkthrough	Technique to evaluate a user interface based on stepping through common tasks that a user would need to perform and evaluating the user's ability to perform each step (e.g., "How many users will click this button for task A? What happens when they do?"). This approach is intended especially to help understand the usability of a system for first-time or infrequent users, that is, for users in an exploratory

Figure 20 - UPCASE Tool - Examples of work-products



UPCASE

Method for self-assessing the capability of usability process of SME

Home

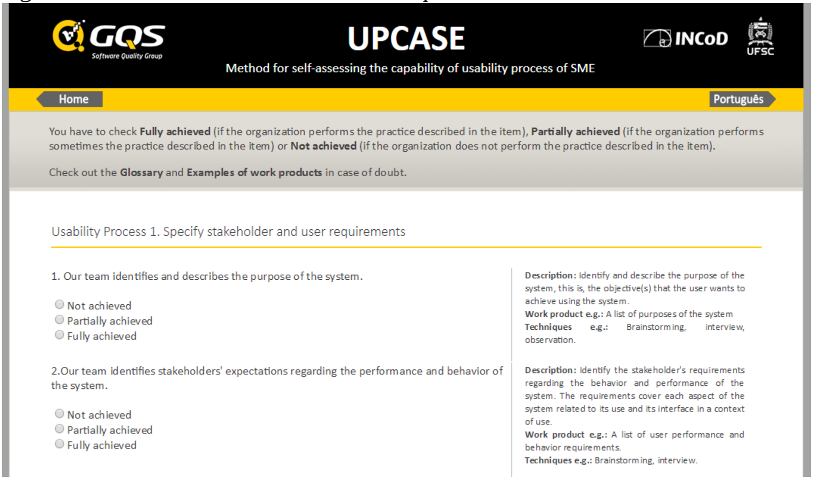
Português

Examples of work products

Work product	Example
System purpose	• Order food for home delivery.
User performance and behavior requirements	• System should be easy to install, easy to read in lighting and noise environments, easy to use by person with low vision capability.
Usability requirements	• Order a meal from the mobile application: • Maximum time to complete task: 2 minutes • Minimum percentage of users who can complete the task: 95% • Percentage of satisfied users when using the system (measured via SUS questionnaire): 90%.
List of user tasks:	• Register account • Login • Search restaurant • See restaurant menu • Add meals to cart • Order meal.

Task	Order food
Frequency of use	Variable (daily, weekly, monthly)
Duration of the task	2-3 min

Figure 21 - UPCASE Tool - Assessment questionnaire



Generation of Assessment Report

Once the user has responded the 16 items of the assessment questionnaire, he can generate the assessment results. The assessment results screen contains:

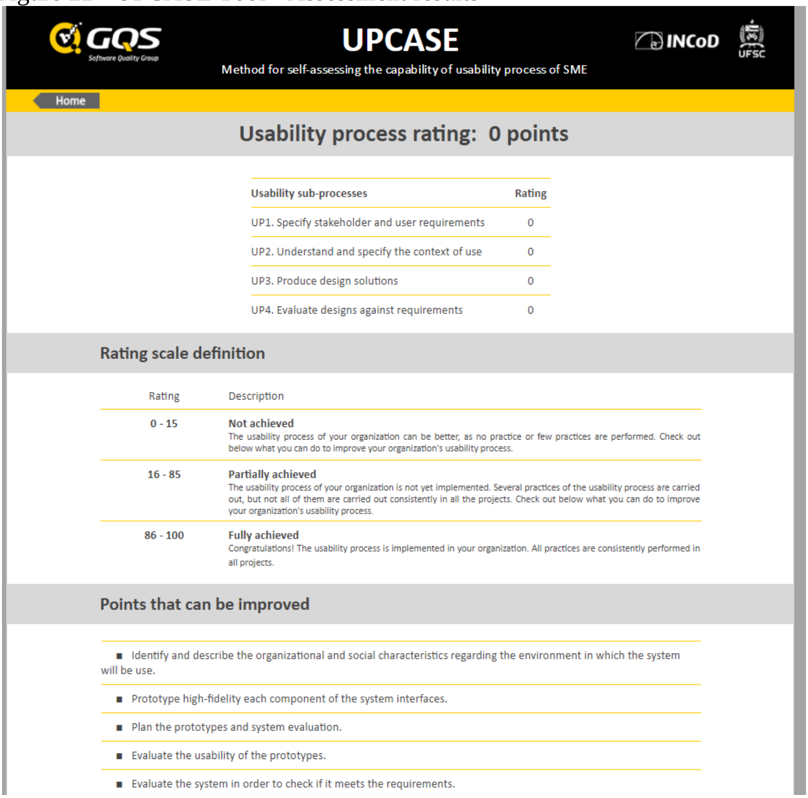
- The usability process rating obtained by the SE.
- The rating for each one of usability sub processes.
- A rating scale definition, which presents a description of the usability process rating, according to its usability range in accordance to the usability process (Formula 2):
  - 0 to 15 points: The usability process of your organization has much to be improved, as no practice or only few practices are performed.
  - 16 to 84 points: The usability process of your organization is only partially implemented. Several practices of the

usability process are carried out, but not all of them are performed consistently in the projects.

  - 85 to 100 points: Congratulations! The usability process is well implemented in your organization. Practices are consistently performed in the projects

- List of points that can be improved by the SE. This list contains suggestions on what the SE should do in order to improve the usability process. The suggestions are based on the items the user marked as “Partially achieved” or “Not achieved”.

Figure 22 - UPCASE Tool - Assessment results



4) The tool has been implemented to work with the PostgreSQL Database Manager System (DBMS), and was programmed in PHP and JavaScript languages. The UPCASE Tool is available online at http://match.inf.ufsc.br:90/upcase/index_en.html.

5) At the end of the implementation stage, the assessment tool was tested through an exploratory test.

7 UPCASE APPLICATION AND EVALUATION

It is essential to assess the quality of an assessment method in terms of reliability and validity in order to ensure that in fact the assessment results point out a correct picture of the capability of the assessed process and valid improvement opportunities, and if the items of the measurement instrument measure the different aspects of the same construct. Furthermore, in the context of SE that are not expected to have employees with expertise in SPI or the usability process, it is fundamental that the assessment method has good usability and can be correctly interpreted by the assessors. This is especially important when considering that the lack of guidance on how to plan the assessment, validate the collected data, calculate and generate the assessment results, may result in an inaccurate assessment and/or may difficult the application of the process assessment in practice. And, in accordance to the identified requirements it is also important that the assessment provides valuable results with low cost/effort.

Thus, in order to evaluate the quality of the UPCASE method we formed a series of evaluations throughout its development:

- *Face validity* by an expert panel
- Usability and comprehensibility by a series of case studies applying UPCASE in a moderated way in SEs.
- Reliability by comparing results from case studies applying UPCASE in a moderated way in SEs with the results of assessments performed by the author.
- Internal consistency by a series of case studies applying UPCASE in an unobserved way.

7.1 EVALUATION OF FACE VALIDITY VIA EXPERT PANEL

At the end of the first development cycle of the UPCASE method, to evaluate the face validity of the UPCASE method, we performed an evaluation via expert panel. The experts evaluated the UPCASE method with the objective to assess whether the method appears to be valid and if it covers all the required aspects of the domain being assessed.

The expert panel was performed in the period from 02/08/2017 to 14/09/2017. Four experts were invited to participate in the panel. Three of them have background in process assessment and one of them have background in usability engineering. All experts are postgraduate and have years of experience in their respective areas of knowledge. Each participant was invited by email to evaluate the UPCASE method. Attached to the invitation, the participants received the reference model, measurement framework and assessment questionnaire of UPCASE, in the form of a technical report (LACERDA et al., 2017). All experts responded to the invitation to evaluate the documentation and send their comments regarding the UPCASE method by e-mail.

The experts considered UPCASE an adequate method for assessing the capability of usability process in SEs. They observed that UPCASE contemplates the basic practices of a usability process for SE and that it presents the necessary elements of an assessment method. They also considered the examples of techniques and work products appropriate. However, few issues have been pointed out, regarding the comprehension of some of the practices (which consequently have been revised), regarding to the completeness of the PRM. They also suggested the addition of some outcomes, which were associated with practices, but had not been added. The experts also suggested to add some complementary examples of techniques and work products.

Two experts expressed concerns about the achievement of Requirement 6 (“The method should not require staff to have prior SPI experience, specific software engineering knowledge nor external experts") as it means that the assessment team does not need to have previous experience in SPI. These two experts believe that the presence of a specialist should be necessary, as they have greater competence to carry out process assessment. However, in order to be accessible to any SE, it is desirable that UPCASE meets Requirement 6, and so ensure that even professionals who do not have SPI (or even usability) expertise can assess their usability process and thus start improving their process.

In addition, suggestions were made to add examples of techniques and work products, as well as to unify practices of usability sub-processes. These suggestions were implemented and resulted in the latest version of the UPCASE method.

7.2 EVALUATION THROUGH A SERIES OF CASE STUDIES

The evaluation goal was decomposed into several research questions presented in Table 45.

Table 45 - Research questions and methods

Research questions	Method for collecting data	Method for data analysis
RQ1 Is the assessment method reliable?	Observed case studies	Intraclass correlation analysis Weighted Kappa analysis
RQ2 Does the method have good usability?	Observed case studies	Analysis of the correct realization, the duration of the assessment and satisfaction of the participants.
RQ3 Does the method allow comprehension of its content?	Observed case studies	Analysis of examples provided for each item of the questionnaire and

		correct execution of the assessment process.
RQ4 Is there evidence of internal consistency in UPCASE questionnaire?	Remote unobserved case studies	Cronbach' alpha analysis

The characteristics evaluated by these research questions are:

- Reliability is the overall consistency of a measure, that is, if the same measuring process provides the same results. A measure is said to have a high reliability if it produces similar results under consistent conditions (TROCHIM, 2006).

- Usability is the extent to which a product can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a specific context of use (ISO/IEC, 1998). In the context of this work, that is, whether the assessment process and the supplementary material (glossary and examples) may be used with efficiency, efficacy and satisfy the users.

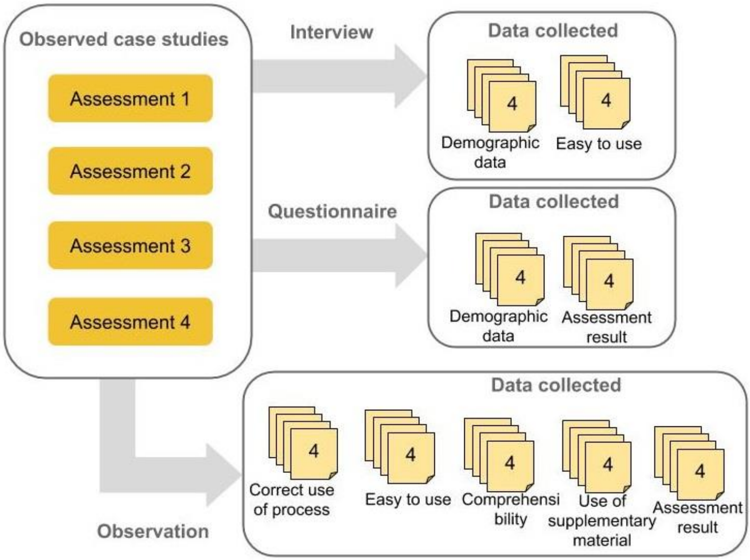
- Comprehensibility is the extent to which a text as a whole is easy to understand (ISSCO, 2016). In the context of this work, that is, the extent to which the items from the assessment questionnaire can be correctly understood.

- Internal consistency: is related to the degree in which a set of items are measuring a single quality factor, i.e., the capability of a usability process (CRONBACH, 1951).

7.2.1 Definition of the observed case studies

The evaluation of the reliability, usability and comprehensibility of the UPCASE method was made through the conduction of case studies presently observed by a researcher (WOHLIN et al., 2016), as shown Figure 23. The case studies consist in the application of the UPCASE method in different small software organizations/projects, under the observation of a researcher. The case studies allowed the observation of the use of the UPCASE method in the realization of usability process self-assessments in several SEs.

Figure 23 - Definition of the observed case studies



The SEs received, by email, an invitation to participate in the case study. The email presented the research motivation and benefits to participants, who would receive feedback on their usability processes. The invited SEs were asked to carry out a self-assessment of its usability process using UPCASE. For this, they were asked to invite employees, who perform tasks related to the design of the systems user interface to participate in an assessment meeting. The self-assessment was supposed be performed as defined by UPCASE. Information on the UPCASE method as well as the assessment questionnaire were made available to assessors by providing the link for the UPCASE Tool (participants could choose between using the tool version in English or Portuguese). During the assessment, a moderator presented the items of the assessment questionnaire. After obtaining a consensual response regarding each item, the moderator answered the respective question of the online questionnaire. During the assessment the observer records information about the usability and comprehensibility of the questionnaire and assessment process. The following information has been recorded:

- Correct execution of the assessment process;
- Duration of the assessment meeting;
- Whether the assessment process was considered easy to use;
- Whether the supplemental materials were used;
- Whether the questionnaire items were interpreted correctly.

After the assessment meeting the researcher asked the participants to present the work products provided as examples during the meeting, and then fill out his own assessment sheet.

7.2.2 Execution of the observed case studies

The observed case studies were carried out in 4 small software organizations/projects, in Brazil, in October 2017, as shown in Table 46. The purpose of the process assessment was clarified to the organizations by email and the assessment meeting was scheduled. The assessment meeting was supported using the UPCASE Tool. The participants performed the assessment autonomously, without interference of the researcher regarding the assessment process or the interpretation of the items of the questionnaire. At the end of the assessment, the observer filled out his own assessment questionnaire, based on the meeting discussion and the examples provided. When the assessment was over, the observer asked that one of the members of the SE to present the work products mentioned as examples during the meeting.

The size of the assessment teams that participated in the case studies ranged from 12 to 47 persons. All SEs develop web systems, two healthcare systems, one governmental and one for marketing. Except for one, all participants of the assessment teams are either UI designer or system analyst.

The duration of the assessment varied between 20 minutes and 80 minutes. In general, the duration of the assessments was considered appropriate (about one hour). The only assessment that lasted more than an hour was the one with a larger number of participants (seven participants). As expected, the number of participants in the assessment meeting influences the duration of the assessment. However, the factor that mostly influences the duration of the assessment is how critical the participants are and how much they discuss each questionnaire item and the number of examples they provide. Another reason that led to the prolongation of some assessment meetings was the attitude of the participants in trying to discuss how some of the items could be achieved, thus initiating already an improvement in the usability process.

Table 46 - Characteristics of the participating SEs

SE	**1**	**2**	**3**	**4**
Number of employees	30	12	47	18
Domain / Platform	Governmental/Web	Health/Web	Health/Web	Other/Web
Number of participants	4 (2 designers, 2 system analysts)	2 (1 designer,1 system analyst)	7 (2 designers, 5 system analysts)	2 (front-end developers)
Assessment duration (minutes)	20	60	80	43

The result of the process assessment carried out in the case studies shows, that despite the variation of ratings reached by which SE, all of them implement the usability process partially (attribute process = “P”) (Table 47). In addition, it can be highlighted that in all SEs, the highest rated sub-process was UP3 (Produce design solutions) and the lowest rated was the sub-process UP4 (Evaluate designs against requirements), indicating that the assessed SEs have a greater capability in the development of design solutions process, but do not yet adequately implement activities for evaluating the developed UI designs.

Table 47 - Assessment score and attribute rating of the assessed SEs

SE	**1**		**2**		**3**		**4**	
	Assessment score	Attribute rating	Assessment score	Attribute rating	Assessment score	Attribute rating	Assessment score	Attribute rating
Usability process	**35**	**P**	**69**	**P**	**63**	**P**	**50**	**P**
UP1	35	P	67	P	50	P	50	P
UP2	30	P	70	P	60	P	50	P
UP3	60	P	90	F	100	F	60	P
UP4	0	N	34	P	50	P	33,33	P

7.2.3 Data analysis of the observed case studies

As result of the observed case studies, the following data was collected on each SE (Table 48): assessment duration, whether the assessment process was implemented correctly or not, if they encountered difficulties in the understanding of the items and if the supplementary material was used. The analysis of the collected data is presented in the following sections.

Table 48 - Data collected per SE

SE	1	2	3	4
Correct use of process	No (Do not use the cards for voting)	Yes	Yes	Yes
Found the assessment easy to do	Yes	Yes	Yes	Yes
Correct understanding of items	1	2	2	0
Use of supplementary material	Read the item description and examples of work products and techniques. Did not access the Glossary and Example pages.	Read the item description and examples of work products and techniques. Did not access the Glossary and Example pages.	Read the item description and examples of work products and techniques. Did not access the Glossary and Example pages.	Read the item description and examples of work products and techniques. Did not access the Glossary and Example pages.

7.2.3.1 RQ1 Is the assessment method reliable?

Table 49 presents the answer to the UPCASE questionnaire filled out by the assessment team of the SE (T) and the observer (O) during the observed case studies. The results of both assessments were compared with the objective of evaluating the reliability of the method. The reliability was analyzed based on a an intra-class correlation analysis (FISHER, 1925) and a Weighted Kappa concordance analysis (COHEN, 1960).

Table 49 - Assessment responses from SEs teams and observers

N	Items	SE 1		SE 2		SE 4		SE 4	
		T	O	T	O	T	O	T	O

1	Our team identifies and describes the purpose of the system.	2	2	2	2	1	2	2	2
2	Our team identifies stakeholders' expectations regarding the performance and behavior of the system.	0	0	1	0	1	0	1	0
3	Our team defines explicit statements of usability requirements based on the context analysis.	0	0	1	0	1	0	0	0
4	Our team identifies and describes the characteristics of the tasks the user performs in the system.	1	1	1	1	1	1	0	0
5	Our team identifies and describes the characteristics of the users.	0	1	1	1	0	1	1	1
6	Our team identifies and describes the organizational and social characteristics regarding the environment in which the system will be use.	1	0	2	1	2	1	1	1
7	Our team identifies and describes the characteristics of the device with which the users will interact to use the system.	1	2	2	2	2	2	1	2
8	Our team describes the physical environment characteristics in which the system will be use.	0	2	1	2	1	2	2	2
9	Our team analyzes the use cases in terms of its flow, navigation, main screens and constraints.	1	1	2	1	2	1	1	0
10	Our team analyzes a range of design options for each	1	1	1	2	2	1	1	1

	aspect of the system related to its use and its interface.								
11	Our team applies existing usability knowledge (such as stakeholder requirements, usability guidelines) in the system design.	1	1	2	2	2	2	2	1
12	Our team specifies each aspect of the system related to its use and its interface.	1	2	2	2	2	2	2	1
13	Our team prototypes high-fidelity each component of the system interfaces.	2	2	2	2	2	2	0	0
14	Our team plans the prototypes and system evaluation.	0	0	0	0	0	0	0	0
15	Our team evaluates the usability of the prototypes.	0	0	0	0	1	0	2	0
16	Our team evaluates the system in order to check if it meets the usability requirements.	0	0	2	1	0	0	0	0
	Total	34,375	46,875	68,75	59,375	62,5	53,125	50	34,375

Intraclass correlation coefficient analysis

Intraclass analysis is an estimate of the fraction of the total variability of measures due to variations between individuals. Intraclass correlation values in the range [0.4; 0.75] are considered satisfactory, and values greater than 0.75 are considered excellent (CICCHETTI, 1994). Weighted Kappa measures the agreement between two raters who each classify N items into C mutually exclusive categories. Weighted kappa allows to verify disagreements especially useful when codes are ordered (COHEN, 1960). Kappa over 0.75 as excellent, 0.40 to

0.75 as fair to good, and below 0.40 as poor (FLEISS; LEVIN; PAIK, 2013).

Analyzing the intra-class correlation regarding each section of the UPCASE assessment questionnaire (presented in APPENDIX J), the following coefficient were obtained, as presented in Table 50.

Table 50 - Intraclass correlation coefficient per questionnaire section

Questionnaire section	**Intraclass correlation coefficient**
Items of usability sub-process 1	--
Items of usability sub-process 2	0,5128
Items of usability sub-process 3	0,7014
Items of usability sub-process 4	0,4285
All sub-process items	**0,5579**

The questionnaire section regarding the usability sub-process 1 did not show variability in the results, which resulted in an undefined result. The ICC is calculated by the variance ratio. Since such values may be zero or negative, the use of this technique may lead to inconclusive results.

On the other hand, the analysis of the questionnaire sections regarding the usability sub-process 2, 3, 4 and regarding the usability process as a whole, presented a coefficient between 0.4 and 0.75, considered satisfactory. This provides a first indication that the UPCASE assessment questionnaire presents a fair to good reliability when used in different moments to assess the same objects, that is, it is reliable to assess the same object with different assessors.

Concordance analysis between assessors

Analyzing the concordance between assessors regarding each question of the UPCASE assessment questionnaire (presented in APPENDIX J), the Weighted Kappa coefficient were obtained, as presented in Table 51.

Table 51 - Weighted Kappa coefficient per SE

SE	Weighted Kappa coefficient
1	0,67033
2	0,351351
3	0,394958
4	0,461538

SEs 1 and 4 obtained coefficients higher than 0,4 and SEs 2 and 3 obtained a coefficient close to 0.4 (coefficients between 0,4 and 0,75 are considered fair to good). Therefore, the Weighted Kappa coefficients obtained through the SEs' assessment, indicate that the use of the UPCASE questionnaire allows a reasonable agreement between different assessors.

7.2.3.2 RQ2 Does the method have good usability?

The assessment method is considered to have good usability if it is effective, efficient and if satisfies the users, in this context the assessment team. All the assessment teams were able to complete the assessment process using the UPCASE method without external assistance, which demonstrates that the method is **effective** to be used in the conduction of a usability process self-assessment. In general, all SEs implemented the assessment process accordingly to UPCASE proposal, except for SE "2". The team from organization 2 did not understand that all assessment participants should present their cards at the same time. In this way, for the first questionnaire item, each participant presented his card once. When using the cards in this way, the purpose of the "Assessment poker" was lost, so in the following items, the cards were no longer used. As a result, it was observed that 2 participants predominated the assessment, while on the other hand one participant practically did not made any comments. SE 3, which had the largest number of participants, found the greatest difficulty in reaching consensus on the items. For items that would require more than 3 voting rounds, was choose the response given by most of the members, even without reaching a consensus. Regarding **efficacy**, it can be considered that the method allows an efficient self-assessment of the usability process, as the assessment and generation of the results took in average 50 minutes, and at most 80 minutes, which is a viable period of time a meeting. Regarding **satisfaction** of the assessment teams, when questioned in the interview after the assessment meeting, all of them answered that were satisfied with the way the assessment was conducted. One of the interviewees claimed that the reason for having liked it was the simplicity of the process. Considering that most SEs implemented the assessment process properly, in an adequate period of time, and that all participants found the assessment process easy to perform, we can consider that UPCASE has good **usability**.

7.2.3.3 RQ3 Does the method allow comprehension of its content?

As the participants discussed each item of the questionnaire, the researcher observed at whether they were understanding the items properly. This was perceived by the examples they provided to justify whether the item was attended or not. In this way some items have been identified that can generate doubts and items that need more examples in order to prevent misinterpretations. The items that demonstrated to be difficult to understand are presented in Table 52.

Table 52 - Questionnaire items that were misunderstood

Item	Difficulty	SE
2.Our team identifies and describes system performance and behavior	They did not correctly understand the concept of performance	1,3

requirements desired by the user.		
3.Our team defines explicit statements of usability requirements based on the context analysis.	They did not correctly understand the concept of Usability Requirement	2,3
4. Our team identifies and describes the characteristics of the tasks the user performs in the system.	They did not correctly understand the concept of Task Characteristics	2

In addition, the use of the term "describe", in items 1, 4, 5, 6, 7 and 8 was considered confusing for participants of SEs 1 and 3, which understood that "describe" implies having to document the activity. That is, an SE that identifies the performance and behavior requirements desired by the user, but that does not document this information may respond that the item is not “not achieved”. Therefore, as the use of the term “describe” can lead to a misinterpretation of the items, it was chosen to remove it from the items APPENDIX H . Yet, Considering the correct interpretation of the assessment process and most of the 13 items of the assessment questionnaire, we can consider that UPCASE has good comprehensibility.

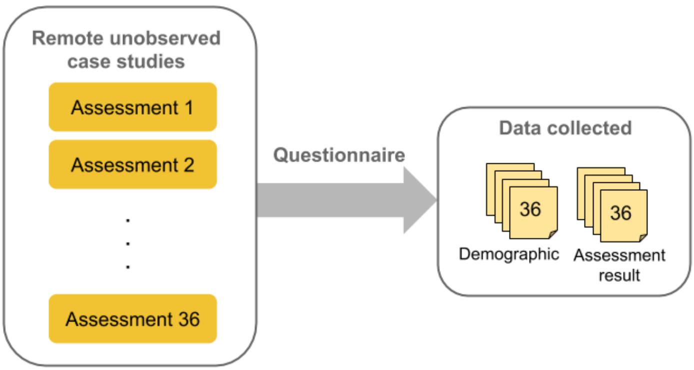
7.2.4 Definition of unobserved case studies

The evaluation of the internal consistency of the UPCASE method was made by conducting a series of unobserved case studies (WOHLIN et al., 2016), as shown in Figure 24. To carry out these unobserved case studies, invitations were sent via email and social networks to various SEs and discussion groups in the area of process assessment, usability and interface development. The invitation disclosed the purpose of UPCASE and its target audience. The invitation presented, as well, the benefits to participants, who would ultimately receive feedback on their usability processes. The SEs are asked to carry out a self-assessment of their usability process using UPCASE Tool (participants could choose between using the tool version in English or Portuguese). We collected the following information on each SE that participated in the case study:

- Assessment questionnaire responses;
- Platform for which the SE develops SW;
- Domain for which the SE develops SW.

The responses obtained in the case studies are used as input to evaluate the internal consistency of the UPCASE assessment questionnaire. To analyze the construct, we performed a Cronbach's alpha analysis. Cronbach α values greater than 0.7 are considered acceptable (DEVELLIS, 2016), thus, indicating internal consistency of the measurement instrument.

Figure 24 - Definition of the remote unobserved case studies



7.2.5 Execution of unobserved case studies

The invitation to carry out the usability process self-assessment was sent to 85 IT organizations/associations and 43 discussion groups related to the development of interface/software systems and usability. Data was collected in the period between October 29 and January 17, 2018. At the end, 36 responses were obtained. As result of the remote case studies, the following data was collected on each questionnaire answer (APPENDIX I): domain and platform of the software developed, number of assessment participants, assessment duration and the assessment scores.

In general, it was observed that the respondents work with software development in a variety of domains, especially in information technology, health and government sectors. The main development platform used is the Web.

Figure 25- System domain

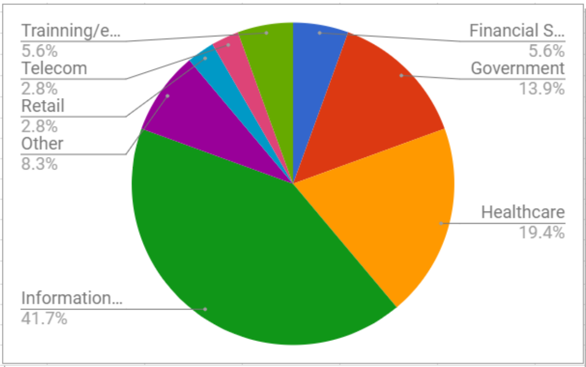
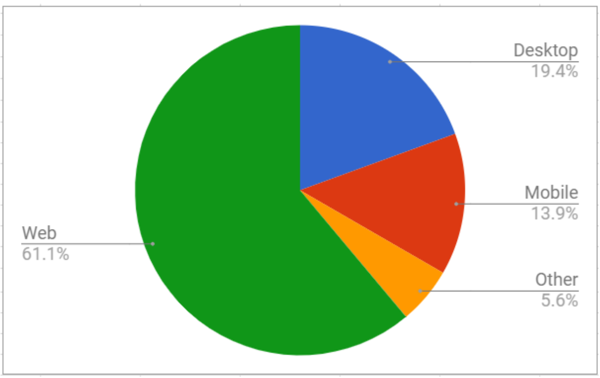


Figure 26 - System platform



The results of the usability process assessments showed that most of the assessed SEs partially implement the usability process (Attribute rating = P), since the majority of them reached a score between 15 and 85 (Figure 27). However, it can be observed that in general they share the same shortcoming, a low implementation of UP4 (Evaluate designs against requirements). The results also demonstrate that the most implemented sub-process is UP2 (Understand and specify the context of use) as shown in Figure 28 to Figure 31.

Figure 27 - Histogram of UP scores of SEs

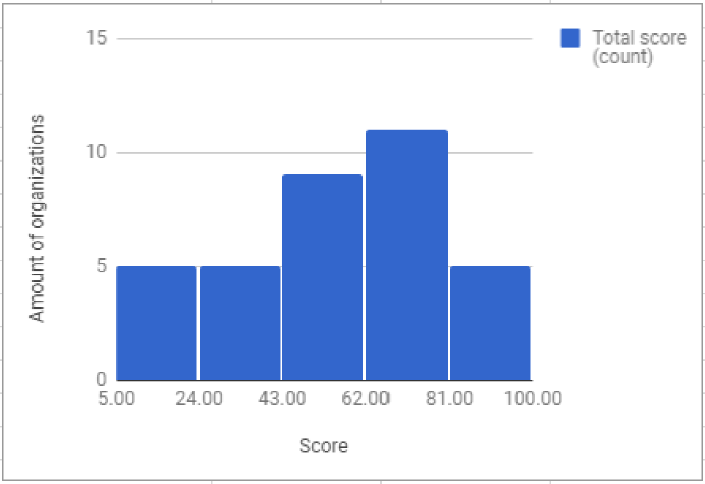


Figure 28 - Histogram of UP1 scores of SEs

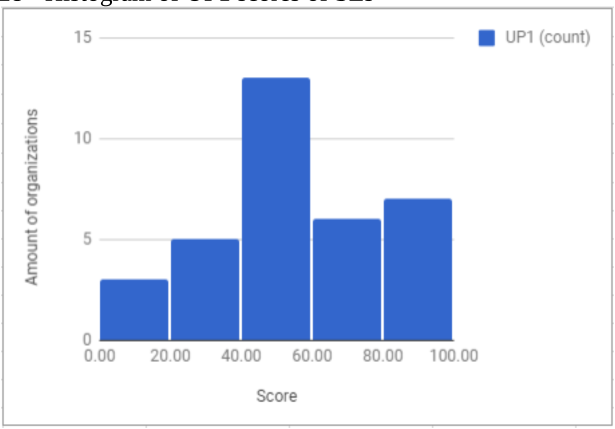


Figure 29 - Histogram of UP2 scores of SEs

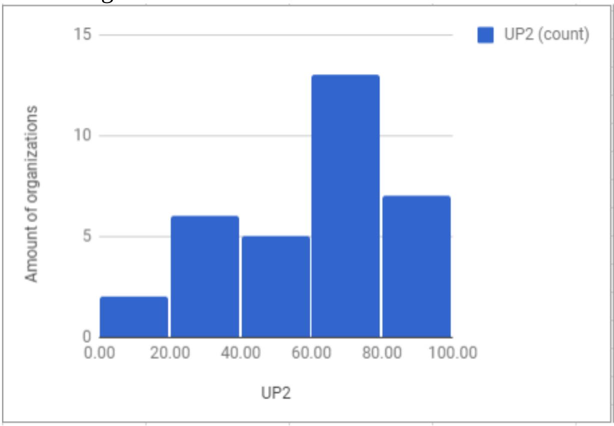


Figure 30 - Histogram of UP4 scores of SEs

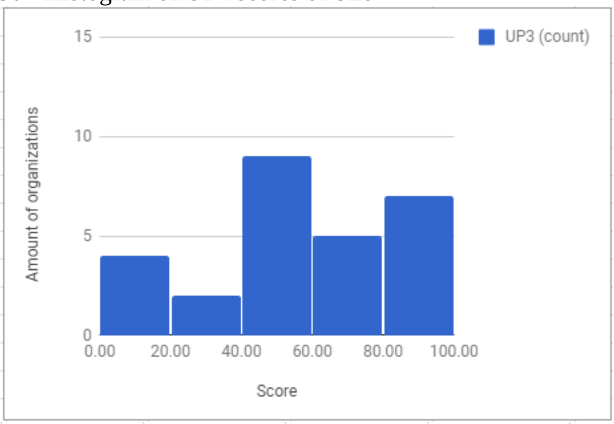
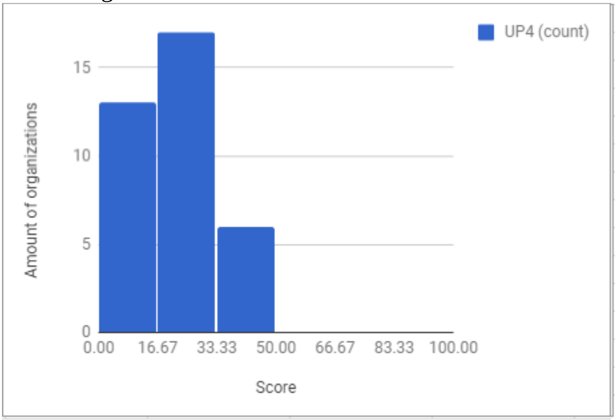


Figure 31 - Histogram of UP4 scores of SEs



7.2.6 Data analysis of unobserved case studies

RQ4. Is there evidence of internal consistency in UPCASE questionnaire?

Analyzing the 16 items of the UPCASE assessment questionnaire (presented in APPENDIX F), the value of Cronbach's is satisfactory (= .914). Hence, a strong evidence is observed towards consistent answers, indicating the internal consistency of the measurement instrument. With the purpose of verifying if all items contribute to the reliability of the questionnaire, the Cronbach alpha coefficient was also calculated n times ("n" = number of items of the questionnaire). In each time one of the questionnaire items is removed. The results of each calculations are presented in Table 53. Therefore, we can conclude that responses between the items are consistent and precise, indicating that the various items of the UPCASE assessment questionnaire purports to measure the same general construct.

Table 53 - Alpha coefficient calculated with n-1 items

Item removed	α
1	0.910850877
2	0.9116298484
3	0.9092369495
4	0.9145593958
5	0.9077940077
6	0.906182051
7	0.907962094
8	0.9138603492
9	0.9067474106
10	0.9068246905
11	0.9040393173
12	0.9071273581
13	0.9075389559
14	0.9071218217
15	0.9093347639
16	0.9105609457

7.3 DISCUSSION OF THE APPLICATION AND EVALUATION OF UPCASE

The obtained results provide a first evidence of the reliability, usability, comprehensibility and internal consistency of UPCASE method acceptable for assessing the capability of the usability process of SE.

Regarding UPCASE reliability (RQ1), analyzing the intra-class correlation coefficient the general results indicate an acceptable reproducibility of quantitative measurements made by different observers using UPCASE. Only the questionnaire items with respect to UP1 obtained inconclusive results, to be able to analyze it will be necessary to carry out more case studies. The analysis of the weighted kappa coefficient, indicated that UPCASE has a fair to good inter-rater agreement, that is, UPCASE is a fair to good method to assess the same object in different situations allowing agreement between assessors. Overall, the result indicated that the UPCASE questionnaire presents a fair degree of reliability when used in different moments to assess the same object.

We also observed that in general the SEs participating in the observed case studies were able to carry out the assessment process as expected, using only the material provided, in a reasonable period of time leaving the participants satisfied with the process used to carry out the assessment. Only one of the SEs did not carry out the assessment process as expected. All assessment meetings were carried out in a reasonable period of time, around 1 hour; and the participants felt satisfied. In this way, these results indicate that UPCASE has good usability (RQ2).

Concerning the ease of comprehension of the UPCASE method content (RQ3), in order to allow the execution of a correct assessment meeting and the correct completion of the assessment questionnaire, it was verified that in the case studies, the participants had difficulties in understanding the items 2, 3 and 4, thus, these items need be revised. Moreover, exception for one of the four case studies, all of them performed the assessment process properly. Therefore, it can be concluded that the UPCASE method has an adequate comprehensibility.

In terms of internal consistency (RQ4), the results of the analysis indicate a satisfactory Cronbach's α, indicating the set of items of the UPCASE questionnaire are measuring a single quality factor. Thus, evidencing that the items of UPCASE questionnaire are consistent and precise with respect to the assessment of the capability of the usability process of SE.

7.4 THREATS TO VALIDITY

As in any research, this one is subject to threats to its validity. Therefore, it was identified potential threats and applied mitigation strategies in order to minimize their impact on the research results.

External Validity

Regarding external validity, a threat to the possibility to generalize the results is related to the sample size and diversity of the data used for the evaluation (WOHLIN et al., 2016). With respect to sample size, we used data collected from 1 observed case studies and 1 unobserved case studies, involving a population of 4 and 32 respondents, respectively. We recognize, however, that in terms of statistical significance, this is a small sample size (MCCALL; RICHARDS; WALTERS, 1977). In addition, although the invitation to participate in the case study has been widely spread nationally and internationally, most participants were from the state of Santa Catarina. Thus, there is the need to execute more case studies to increase the sample size and to obtain more responses from different states and countries. In its respective domains, we believe that the results can be generalized to other small software organizations.

Internal Validity

Another issue refers to the correct choice of methods for conducting the data analysis. To minimize this threat we performed a statistical evaluation based on the approach for the construction of measurement scales as proposed by (CRONBACH, 1951), and for assessing disagreement between respondents and measurement items correlation (COHEN, 1960; FISHER, 1925). In respect to sample size and the user representativeness bias, we have used data collected from 6 case studies, involving a population of 36 and 4 case studies. In terms of statistical significance, these are small samples, but robust to estimate of the coefficient alpha (with 36 case studies) (YURDUGÜL, 2008) and enough to indicate the generation of initial reliable results via intraclass correlation analysis and the concordance analysis (with 4 case studies) (BUJANG; BAHARUM, 2017).

Construct Validity

Construct validity concerns generalizing the result of the study to the concept behind the study (WOHLIN et al., 2012). Threats to construction validity are related to instrument for data collection, which may not contain the set of the evaluation objective. In this way, the instrument of data collection was systematically developed using the GQM approach, each metric collected is related to a question, which seek to meet the evaluation objective.

Considering that interpretation of the collected data is among the main threats of validity in a research, we take a careful step in the analysis of subjective data, used to evaluate the usability and comprehensibility of the UPCASE method. To reduce the risk of misevaluate whether the assessment teams found that UPCASE method has good usability or if is difficult to comprehend, the data analysis was based on the triangulation of the data obtained from the case studies interviews and from the information collected by the case study observer.

8 CONCLUSIONS AND FUTURE WORK

Usability is an important factor, for the success of software products. Any other product quality, usability is directly influenced by the software process quality, thus it is important to define and implement appropriated usability processes. In this way, it is essential to make use of appropriate methods that allow the identification of opportunities for improvement, and thus, be able to initiate process improvement.

However, by reviewing the literature, only 15 usability models were found. Most of the models presented a measurement framework, few contained or indicated a process reference model, or did not provide sufficient elements to allow their application in practice. In addition, none of the models found are specific to small organizations. Focusing on self-assessment as a solution for SEs, 33 software process self-assessing methods were found in literature. Of these, 14 are specific to small organizations, that in general, propose simplified assessment processes, and a focus on data collection, most often performed through the application of questionnaires. No method for self-assessing the capability of the usability process in SE was encountered. In this context, it was identified that there was an increase in the number of usability process assessment methods, demonstrating the increased interest in this subject, however, most of the methods found were developed ad hoc way and without focusing on the needs of small organizations software.

Observing a gap in the state of art today for usability evaluation methods for small businesses, this study aimed to develop UPCASE, the method for self-Assessing the capability of the usability process in SEs. The method is based on the ISO/IEC 29110 and ISO/IEC 18529 standard and customized with respect to requirements specific to SEs. UPCASE includes a usability process reference model customized for SEs, a measurement framework and an assessment model, which includes a questionnaire, an assessment briefing script, a glossary and examples of work-products to be used as support throughout the assessment. In addition to the development of the assessment method, a free online tool was developed to facilitate the application of the method by organizations that wish to self-assess their processes, as well as to data gathering questionnaire answers.

UPCASE was applied in four observed case studies and 32 remote case studies, in which software organizations self-assessed their usability process used the developed online tool. Analyzing the data collected in the case studies we obtained results of an initial evaluation that indicates that the method is reliable, has good usability and comprehensibility and has a questionnaire with an acceptable internal consistency. However, we recognize that the sample size may be small, and more case studies should be performed to validate the results obtained so far.

The main scientific contribution includes the review of the state of the art in relation to the usability maturity/capability models; and in relation to the methods for self-assessing software processes. Another contribution of this research is the UPCASE itself, as differently from the currently known usability reference models, including ISO/IEC 18529, it is defined specifically for the characteristics of small organizations. It also differs from other assessment methods by having a questionnaire and script assessment with a glossary and examples to aid during the self-assessments.

It is expected that the availability of the UPCASE method, will facilitate the realization of usability process assessment in small organizations, as it allows the organization itself to carry out the assessment, in a short period of time, using few financial and human resources and without the need to recruit members to the organization. Thus, it is expected that using the proposed method those organizations may improve their process, developing products with better usability, and consequently with higher quality.

In order to obtain further evidence, it would be important to conduct more case studies, using UPCASE to perform the self-evaluation of usability process in more SE, to obtain a larger number of data points in order to validate the effectiveness and efficiency of the method, or to verify the necessity to improve the method. In addition, more case studies could be carried out to evaluate the process of an organization comparing the results of respondents with different degrees of usability expertise, and thus to assess whether the method allows both experts and lay can use it.

It is expected that by developing UPCASE, the SEs may assess their usability process in an autonomous, cheap and fast way. It is also expected that using UPCASE the SEs may increase their knowledge on usability process, as well as identify the opportunities of improvement, and thus, have inputs that allow them to initiate an improvement of their usability process and consequently of the usability of the software systems they developed.

9 REFERENCES

ABES. Mercado Brasileiro de Software: panorama e tendências 2017. **São Paulo: ABES**, 2017.

ABUSHAMA, H. M. PAM-SMEs: process assessment method for small to medium enterprises. **Journal of Software: Evolution and Process**, v. 28, n. 8, p. 689–711, 2016.

ACM; IEEE CS. **Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science**. Ney York: ACM, 2013.

ACUÑA, S. T. et al. The Software process: modeling evaluation and improvement. Handbook of Software Engineering and Knowledge Engineering. **Handbook of Software Engineering and Knowledge Engineering**, v. 0, n. 0, p. 35, 2000.

ALEXANDRE, S.; RENAULT, A.; HABRA, N. **OWPL: A gradual approach for software process improvement in SMEs**. Conference on Software Engineering and Advanced Applications. SEAA'06. 32nd EUROMICRO. **Anais**...Cavtat, Croatia: IEEE, 2006

ANACLETO, A. **Método e Modelo de Avaliação para Melhoria de Processos de Software em Micro e Pequenas Empresas**. [s.l.] UFSC, 2004.

ANACLETO, R. et al. **A method for process assessment in small software companies**. 4th International SPICE Conference on Process Assessment and Improvement (SPICE2004). **Anais**...Lisbon, 2004

APRIL, A.; ABRAN, A.; DUMKE, R. **Software maintenance productivity measurement: how to assess the readiness of your organization**. International Conference on Software Process and Product Measurement (IWSM/Metrikon 2004). **Anais**...Konig- Winsterhausen, Germany: Springer, 2004

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. Goal and Question Metric Paradigm. In: MARCINIAK, J. J. (Ed.). . **Encyclopedia of Software Engineering**. [s.l.] John Wiley and & Sons, 1994. p. 528–532.

BEECHAM, S. et al. Using an Expert Panel to Validate a Requirements Process Improvement Model. **The Journal of Systems and Software**, v. 76, n.3, 2005.

BEVAN, N. **Usability process improvement and capability assessment**. Conference on Human Computer Interaction. **Anais**...Toulouse: Cepadues, 2001

BIAS, R.; MAYHEW, D. **Cost-justifying usability: An update for the Internet age**. Amsterdam: Elsevier, 2005.

BIEL, B.; GRILL, T.; GRUHN, V. Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application. J. of Syst. and Softw. v. 83, n. 11, p. 2031–2044, 2010.

BLANCHETTE, S.; KEELER, K. L. **Self Assessment and the CMMI-AM: A Guide for Government Program Managers**PittsburghCarnegie Mellon University, 2005. Disponível em: <http://www.sei.cmu.edu/publications/pubweb.html>

BOEHM, B. **A view of 20th and 21st century software engineering**. 28th international conference on Software engineering. **Anais**...2006

BOURQUE, P.; FAIRLEY, R. E.; OTHERS. **Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0**. [s.l.] IEEE Computer Society Press, 2014.

BUJANG, M. A.; BAHARUM, N. Guidelines of the minimum sample size requirements for Kappa agreement test. **Epidemiology, Biostatistics and Public Health**, v. 14, n. 2, 2017.

CARSON, D. et al. **Qualitative marketing research**. Thousand Oaks: Sage, 2001.

CHANG, S. K. **Handbook of software engineering and knowledge engineering**. [s.l.] World Scientific, 2001. v. 1

CICCHETTI, D. V. Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. **Psychological assessment**, v. 6, n. 4, p. 284, 1994.

CIGNONI, G. A. **Rapid software process assessment to promote innovation in SMEs**. Proceedings of Euromicro. **Anais**...1999

COHEN, J. A coefficient of agreement for nominal scales. **Educational and psychological measurement**, v. 20, n. 1, p. 37–46, 1960.

CONKLIN, P. F. **Bringing usability effectively into product development**. Workshop on Human-computer interface design: success stories, emerging methods, and real-world context: success stories, emerging methods, and real-world context. **Anais**...1995

CRONBACH, L. J. Coefficient alpha and the internal structure of tests. **Psychometrika**, v. 16, n. 3, p. 297–334, 1951.

CSTA (COMPUTER SCIENCE TEACHERS ASSOCIATION). **Bugs in the System: Computer Science Teacher Certification in the**

USCSTA. [s.l: s.n.].

DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. **MIS quarterly**, p. 319–340, 1989.

DER WIELE, T. et al. Improvement in organizational performance and self-assessment practices by selected American firms. **Quality Management Journal**, v. 7, n. 4, p. 8–22, 2000.

DERNIAME, J.-C.; KABA, B. A.; WASTELL, D. **Software process: principles, methodology, and technology**. [s.l.] Springer, 2006.

DEVELLIS, R. F. **Scale development: theory and applications**. 4. ed. [s.l.] SAGE Publications, 2016.

EARTHY, J. **Usability maturity model: processes.** Lloyd's Register of Shipping. v.2 pp. 84, 1999.

EASON, K.; HARKER, S. D. User centred design maturity. **Internal working document**, 1997.

FISHER, R. A. **Statistical methods for research workers**. [s.l.] Genesis Publishing Pvt Ltd, 1925.

FLANAGAN, G. A. Usability Leadership Maturity Model (Self-assessment Version). **CHI'**, v. 95, 1995.

FLEISS, J. L.; LEVIN, B.; PAIK, M. C. **Statistical methods for rates and proportions**. 3. ed. [s.l.] John Wiley & Sons, 2013.

FUCHS, B.; RITZ, T.; STRAUCH, J. **Usability of mobile applications: dissemination of usability engineering in small and medium enterprises**. International Conference on Data Communication Networking, e-Business and Optical Communication Systems. **Anais**...Rome: SciTePress, 2012

GARCIA, I.; PACHECO, C.; CRUZ, D. Adopting an RIA-Based Tool for Supporting Assessment, Implementation and Learning in Software Process Improvement under the NMX-I-059/02-NYCE-2005 Standard in Small Software Enterprises. **Software Engineering Research, Management and Applications**, p. 29–35, 2010.

GILL, A.; HENDERSON-SELLERS, B.; NIAZI, M. Scaling for agility: A reference model for hybrid traditional-agile software development methodologies. **Information Syst. Frontiers**, v. 2, n. 1, p. 1–27, 2016.

GOETHALS, A. **An example self-assessment using the NDSA levels of digital preservation**. Capability assessment and improvement workshop (CAIW) at IPRES. **Anais**...2013

GRENNING, J. Planning poker or how to avoid analysis paralysis while release planning. **Hawthorn Woods: Renaissance Software Consulting**, v. 3, 2002.

GRINDROD, K.; LI, M.; GATES, A. Evaluating user perceptions of mobile medication management applications with older adults: a usability study. **JMIR mHealth and uHealth**, v. 2, p. 1, 2014.

HABRA, N. et al. Initiating software Process Improvement in very small enterprises experience with light assessment tool. **Information and Software Technology**, v. 50, p. 1, 2008.

HADDAWAY, N. R. et al. The role of Google Scholar in evidence reviews and its applicability to grey literature searching. **PloS one**, v. 10, n. 9, p. e0138237, 2015.

HARTER, D. E.; KEMERER, C. F.; SLAUGHTER, S. A. Does software process improvement reduce the severity of defects? A longitudinal field study. **IEEE Transactions on Software Engineering**, v. 38, n. 4, p. 810–827, 2012.

HAUCK, J. C. R. **Um Método de Aquisição de Conhecimento para Customização de Modelos de Capacidade/Maturidade de Processos de Software**. [s.l.] UFSC, 2011.

HERING, D. et al. **Integrating usability-engineering into the software developing processes of SME: a case study of software developing SME in Germany**. Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering. **Anais**...2015

HIX, D.; HARTSON, H. R. **Developing User Interfaces: Ensuring Usability Through Product & Process**. [s.l.] Wiley Professional Computing, 1993.

HUANG, M.-X.; XING, C.-X.; YANG, J.-J. E-government maturity model and its evaluation. **Journal of Southeast University**, v. 24, n. 3, p. 389–392, 2008.

ISLAM, Z.; ZHOU, X. **Software Process Improvement Framework for Software Outsourcing Based On CMMI**. [s.l.] Göteborgs universitet, 2011.

ISO/IEC. **IEEE Standard Glossary of Software Engineering TerminologyIEEE**, 1990.

ISO/IEC. **ISO/IEC 9241-11 - Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on**

usability, Geneva, 1998.

ISO/IEC. **ISO/IEC 13407 - Human-centred design processes for interactive systems**Geneva, 1999.

ISO/IEC. **ISO 18529 - Ergonomics -Ergonomics of human- system interaction - Human-centred lifecycle process descriptions,** Geneva, 2000.

ISO/IEC. **ISO/IEC 9126-1 -- Software Engineering -- Part 1: Product Quality**Geneva, 2001a.

ISO/IEC. **ISO/IEC 9241-210 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems,** Geneva, 2001b.

ISO/IEC. **ISO/IEC 13485 - Medical Devices-Quality Management Systems-Requirements for Regulatory Purposes,** Geneva, 2003a.

ISO/IEC. **ISO/IEC 15504-2 Information technology - Process assessment - Part 2: Performing an assessmentISO/IEC,** Geneva, 2003b.

ISO/IEC. **ISO/IEC 15504-4 - Information Technology - Process Assessment - Part 4: Guidance on use for Process Improvement and Process Capability Determination**Geneva, 2003c.

ISO/IEC. **15504-3 - Information Technology - Process Assessment - Part 3: Guidance on Performing an Assessment,** Geneva, 2003d.

ISO/IEC. **ISO/IEC 15504-1 - Information technology - Process assessment - Part 1: Concepts and vocabularySoftware Process: Improvement and Practice**Geneva, 2004.

ISO/IEC. **15504-5 - Information Technology - Process Assessment - Part 5: An exemplar Process Assessment Model**, Geneva, 2005.

ISO/IEC. **ISO/IEC 21827 - Information technology - Security techniques - Systems Security Engineering - Capability Maturity Model,** Geneva, 2008a.

ISO/IEC. **IEC/IEC 12207 Systems and software engineering-software life cycle processes,** Geneva, 2008b.

ISO/IEC. **ISO/IEC 25010 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models,** Geneva, 2010.

ISO/IEC. **ISO/IEC 15504-6 - Information technology - Process assessment - Part 6: An exemplar system life cycle process assessment**

model, Geneva, 2013.

ISO/IEC. **ISO/IEC/IEEE 15288 - Systems and software engineering -- System life cycle processes,** Geneva, 2015.

ISO/IEC. **TR 29110-1 - Software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 1: Overview,** GenevaISO, 2016a.

ISO/IEC. **ISO/IEC TR 29110-3 - Systems and software engineering - Lifecycle profiles for Very Small - Part 3: Assessment guide,** Geneva, 2016b.

ISSCO. **Comprehensibility**. Disponível em: <https://www.issco.unige.ch/en/research/projects/isle/femti/html/180.html>. Acesso em: 5 fev. 2018.

JOKELA, T. **A Method-Independent Process Model of User-MODEL**. [s.l.] Springer, 2002.

JOKELA, T. **The KESSU usability design process model**. Oulu, 2004.

JOKELA, T. Evaluating the User-Centredness of Development Organisations: Conclusions and Implications from Empirical Usability Capability Maturity Assessments. **Interacting with Computers**, v. 16, p. 1, 2004b.

JOKELA, T. et al. A survey of usability capability maturity models: implications for practice and research. **Behaviour & Information Technology**, v. 25, n. 3, p. 263–282, 2006.

JOKELA, T.; LALLI, T. **Usability and CMMI: does a higher maturity level in product development mean better usability?** CHI’03 extended abstracts on Human factors in computing syst. **Anais**...New York: ACM, 2003

KALINOWSKI, M. et al. **Results of 10 years of software process improvement in Brazil based on the MPS-SW Model**. International Conference on the Quality of Information and Communications Technology (QUATIC),. **Anais**...2014

KALPANA, A. M.; JEYAKUMAR, A. E. Software process improvization framework using fuzzy logic based approach for Indian small scale software organization. **International Journal of Computer Science and Network Security**, v. 10, n. 3, p. 111–118, 2010.

KAR, S. et al. **Self-assessment Model and Review Technique for SPICE: SMART SPICE**. International Conference on Software Process

Improvement and Capability Determination. **Anais**...Berlin: Springer Berlin Heidelberg, 2012

KASUNIC, M. **Designing an Effective Survey**. Pittsburgh: Software Engineering Institute / Carnegie Mellon University, 2005. v. Hadbook CM

KAZI, A. M.; KHALID, W. Questionnaire designing and validation. **Journal of the Pakistan Medical Association**, v. 62, n. 5, p. 514, 2012.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering:a tertiary study. **Information and Softw. Technology**, v. 52, n. 8, p. 792–805, 2010.

KLEIN, L. **UX for lean startups: Faster, smarter user experience research and design**. 1. ed. [s.l.] O'Reilly Media, 2013.

KOMI-SIRVIÖ, S. **Development and evaluation of software process improvement methods**. [s.l.] VTT, 2004.

KROEGER, T. A.; DAVIDSON, N.; COOK, S. C. Understanding the characteristics of quality for software engineering processes: A Grounded Theory investigation. **Inf. Softw. Technol.**, v. 56, n. 2, p. 252–271, 2014.

KROSNICK, J. A.; PRESSER, S. Question and questionnaire design. **Handbook of survey research**, v. 2, n. 3, p. 263–314, 2010.

KRUEGER, R. A.; CASEY, M. A. **Focus groups**. Thousand Oaks: SAGE Publications, 2000. v. 610

KRUMM, J. **Ubiquitous computing fundamentals**. [s.l.] Chapman and Hall/CRC Press, 2016.

KUILBOER, J.; ASHRAFI, N. **Software process and product improvement: an empirical assessment**. [s.l.] Management Science/Information Systems, 2000.

KUVAJA, P.; PALO, J.; BICEGO, A. TAPISTRY: a software process improvement approach tailored for small enterprises. **Software Quality Journal**, v. 8, n. 2, p. 149–156, 1999.

LACERDA, T. C. et al. **UPCASE - A Method for self-assessing the capability of the usability process in small organizations**. Technical Report. Brazil: 2017.

LACERDA, T. C.; VON WANGENHEIM, C. G. A Systematic Literature Review of Usability Capability/Maturity Models. **Computer Standards & Interfaces**, v. 55, p. 95–105, 2017.

LARRUCEA, R. et al. Software Process Improvement in Very Small

Organizations. **IEEE Software**, v. 33, n. 2, p. 85–89, 2016.

MACMAHON, S. T.; MC CAFFERY, F.; KEENAN, F. **Development and Validation of the MedITNet Assessment Framework: Improving Risk Management of Medical IT Networks**. Proceedings of the 2015 International Conference on Software and System Process. **Anais**...2015.

MAJCHROWSKI, A. et al. Software development practices in small entities: an ISO29110-based survey. **Journal of Software: Evolution and Process**, v. 28, n. 11, p. 990–999, 2016.

MARCILLY, R. et al. Usability Flaws in Medication Alerting Systems: Impact on Usage and Work System. **Year book of medical informatics**, v. 10, n. 1, p. 55–67, 2015.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 7. ed. Sâo Paulo: Atlas, 2010.

MATOOK, S.; INDULSKA, M. Improving the quality of process reference models: A quality function deployment-based approach. **Decision Support Systems**, v. 47, n. 1, p. 60–71, 2009.

MAYHEW, D. J. **The usability engineering lifecycle : a practitioner's handbook for user interface design**. [s.l.] Morgan Kaufmann Publishers, 1999.

MC CAFFERY F., & C. G. The development of a low-overhead assessment method for Irish software SMEs. **Journal of Information Technology Management**, v. 18, p. 2, 2007.

MCCAFFERY, F.; TAYLOR, P. S.; COLEMAN, G. Adept: A unified assessment method for small software companies. **IEEE software**, v. 24, n. 1, p. 24–31, 2007.

MCCALL, J. A.; RICHARDS, P. K.; WALTERS, G. F. **Factors in Software Quality, Voll. I, II, III: Final Tech. Report**. [s.l: s.n.].

METZKER, E.; REITERER, H. Use and Reuse of HCI Knowledge in the Software Development Lifecycle. In: HAMMOND, T. G.; WESSON, J. (Eds.). . **Usability**. [s.l.] Springer US, 2002. p. 39–55.

MIRNA, M. et al. **Expected requirements in support tools for software process improvement in SMEs**. 2012 IEEE Ninth Conference Electronics, Robotics and Automotive Mechanics (CERMA). **Anais**...IEEE, 2012

MISHRA, D.; MISHRA, A. Software process improvement in SMEs:

A comparative view. **Computer Science and Information Systems**, v. 6, n. 1, p. 111–140, 2009.

MORAIS, C. H. B.; SBRAGIA, R. **Management of Multi-Project Environment by Means of Critical Chain Project Management : A Brazilian Multi-Case Study**. Proceedings of PICMET '12: Technology Management for Emerging Technologies. **Anais**...2012

NIELSEN, J. **Usability engineering**. Amsterdam: Elsevier, 1994.

O'CONNOR, R. V. **Exploring the role of usability in the software process: A study of irish software smes**. Software Process Improvement. **Anais**...Berlin: Springer, 2009

O'CONNOR, R. V; LAPORTE, C. Y. **Using ISO/IEC 29110 to harness process improvement in very small entities**. Systems, Software and Service Process Improvement. **Anais**...Springer, 2011

PATEL, C.; RAMACHANDRAN, M. Agile Maturity Model (AMM): A software process improvement framework for agile software development practices. **Int. J. of Software Engineering, IJSE**, v. 2, n. I, p. 3–28, 2009.

PAULK, M. **The Capability Maturity Model for Software: Guidelines for Improving the Software Process**. Reading: Addison-Wesley, 1995.

PAULK, M. C. Using the software CMM in small organizations. **Pacific Northwest Software Quality Conf.**, v. 350, 1998.

PETERSEN, K. et al. **Systematic Mapping Studies in Software Engineering.** Proceeding EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering. **Anais**...Italy: BCS Learning & Development, 2008

PINO, F. J. et al. Assessment methodology for software process improvement in small organizations. **Information and Software Technology**, v. 52, n. 10, p. 1044–1061, 2010.

PINO, F. J.; GARCIA, F.; PIATTINI, M. Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review. **Software Quality Journal**, v. 16, n. 2, p. 237–261, 2008.

POUS, M.; CECCARONI, L. **Multimodal interaction in distributed and ubiquitous computing**. International Conference on Internet and Web Applications and Services. **Anais**...Barcelona: IEEE, 2010

PPGCC/UFSC. **PPGCC: Linhas de pesquisa**. Disponível em: <http://ppgcc.posgrad.ufsc.br/linhas-de-pesquisa-2/>. Acesso em: 2 dez.

2017.

PRESSMAN, R. S. **Software engineering: a practitioner's approach**. [s.l.] Palgrave Macmillan, 2005.

RABELLO, R. et al. **Integração de engenharia de usabilidade em um modelo de capacidade/maturidade de processo de software**. Conference on Human Factors in Computing Systems. **Anais**...Brazilian Computer Society, 2012

RENZI, A. B. et al. **Startup Rio: user experience and startups**. International Conference of Design, User Experience and Usability. **Anais**...Springer, 2015

RITCHIE, L.; DALE, B. G. Self-assessment using the business excellence model: a study of practice and process. **International Journal of Production Economics**, v. 66, n. 3, p. 241–254, 2000.

RITTGEN, P. **Quality and perceived usefulness of process models**. Proceedings of the 2010 ACM Symposium on Applied Computing. **Anais**...2010

ROGERS, Y.; SHARP, H.; PREECE, J. **Interaction design: beyond human-computer interaction**. [s.l.] John Wiley & Sons, 2011.

ROUT, T. P. et al. **The rapid assessment of software process capability**. First International Conference on Software Process Improvement and Capability Determination. **Anais**...2000

SÁNCHEZ-GORDÓN, M.-L.; O'CONNOR, R. V. Understanding the gap between software process practices and actual practice in very small companies. **Software Quality Journal**, v. 24, n. 3, p. 549–570, 2016.

SAUNDERS M. N. K.; LEWIS P.; THORNHILL, A. **Research Methods for Business Students**. 5. ed. Research Methods for Business Students: Prentice Hall, 2009.

SEBRAE. **Normas e certificações em software - qual serve melhor para mim? ISO/IEC 29110 / ISO 9000 / CMMI / MPS-BR.** Brasília: [s.n.].

SEBRAE. **Participação das micro e pequenas empresas na economia brasileira**. [s.l: s.n.].

SHACKEL, B. **The concept of usability**. [s.l.] Prentice Hall, Englewood Cliffs (NJ), 1984.

SIG AUTOMOTIVE. **Automotive SPICE process assessment model. Final Release.**, 2010.

SILVA, T. et al. **User-Centered Design and Agile Methods: A Systematic Review**. Agile Conference (AGILE). **Anais**...2011.

SIMONSSON, M.; JOHNSON, P.; WIJKSTRÖM, H. **Model-based IT governance maturity assessments with COBIT**. European Conference on Information Systems (ECIS). **Anais**...2007

SINISCALCO, M. T.; AURIAT, N. Questionnaire design. **Quantitative research methods in educational planning**, p. 1–92, 2005.

SOFTEX. **MPS. BR - Melhoria de Processo do Software Brasileiro**. [s.l: s.n.].

SOLINGEN, R. V. Measuring the ROI of software process improvement. **IEEE Software**, v. 21, p. 32–38, 2004.

STOKES, D.; BERGIN, R. Methodology or “methodolatry”? An evaluation of focus groups and depth interviews. **Qualitative market research: An international Journal**, v. 9, n. 1, p. 26–37, 2006.

SULAYMAN, M. et al. Software process improvement success factors for small and medium Web companies: A qualitative study. **Information and Software Technology**, v. 54, n. 5, p. 479–500, 2012.

TATE, J. **Software process quality models: A comparative evaluation**. [s.l.] Durham University, 2003.

TEAM, C. P. **CMMI for Development, Version 1.3**. Carnegie Mellon University: Software Engineering Institute, 2010.

TIM, K. **Practical insight into the CMMI (2 ed.)**. Boston, MA: USA Artechhouse, 2004.

TREERATANAPON, T. **Design of the usability measurement framework for mobile applications.** International Conference on Computer and Information Technology. **Anais**...Bangladesh: IEEE, 2012

TROCHIM, W. M. K. Theory of reliability. 2006.

USABILITY NET. **Brainstorming**. Disponível em: <http://www.usabilitynet.org/tools/brainstorming.htm>. Acesso em: 23 nov. 2017.

VARKOI, T. **Deployment Package Basic profile**. 2009.

VON WANGENHEIM, G. et al. **Systematic literature review of software process capability/maturity models.** International Conference on Software Process. Improvement And Capability dEtermination (SPICE). **Anais**...Pisa: 2010

VON WANGENHEIM, G.; C. ANACLETO, A. & S. S. S. Helping Small companies assess software processes. **IEEE Software**, v. 23, n. 1, 2006.

WASSERMAN, A. I. **Software engineering issues for mobile application development**. FSE/SDP workshop on Future of soft engineering research. **Anais**...Santa Fe, NM: ACM, 2010

WESKE, M. **Business Process Management: Concepts and Languages, Architectures**. 2. ed. [s.l.] Springer, 2012.

WOHLIN, C. et al. **Experimentation in software engineering**. New York: Springer Science & Business Media, 2016.

YIN, R. K. **Case study research: design and methods.** 4. ed. Beverly Hills: Sage Publications, 2009.

YUCALAR, F.; ERDOGAN, S. Z. A questionnaire based method for CMMI level 2 maturity assessment. **Journal of aeronautics and space technologies**, v. 4, n. 2, p. 39–46, 2009.

YURDUGÜL, H. Minimum sample size for Cronbach’s coefficient alpha: a Monte-Carlo study. **Hacettepe Üniversitesi Journal of Education**, v. 35, n. 35, p. 397–405, 2008.

ZHOU, X. et al. **A Map of Threats to Validity of Systematic Literature Reviews in Software Engineering**. Software Engineering Conference (APSEC). **Anais**...2016.

9.1 REFERENCES FROM SECTION 4.1

[1] EARTHY, J. **Usability maturity model: processesLloyd’s Register of Shipping**. 1999.

[2] DE BRUIN, T. ET AL. **Understanding the main phases of developing a maturity assessment model**. Australasian Conference on Information Systems (ACIS). **Anais**...Sydney: Association for Information Systems, 2005.

[3] VON WANGENHEIM, G. ET AL. **Systematic literature review of software process capability/maturity models.** International Conference on Software Process. Improvement And Capability dEtermination (SPICE). **Anais**...Pisa: 2010.

[4] JOKELA, T. M. **Assessment of user-centred design processes as a basis for improvement action: An experimental study in industrial settings.** Oulu: 2003.

[5] PAULK, M. C. ET AL. Capability maturity model, version 1.1. **IEEE software**, v. 10, n. 4, p. 18–27, 1993.

[6] SWARD, D. **User Experience Design: A Strategy for Competitive Advantage**. Americas Conference on Information Systems. **Anais**...Keystone: 2007.

[7] RAZA, A.; CAPRETZ, L. F.; AHMED, F. An open source usability maturity model (OS-UMM). **Computers in Human Behavior**, v. 28, n. 4, p. 1109–1121, 2012.

[8] EARTHY, J. Usability Maturity Model: Human Centredness Scale. **INUSE Project Deliverable D5**, v. 5, p. 1–34, 1998.

[9] MARCUS, A.; GUNTHER, R.; SIEFFERT, R. Validating a standardized usability/user-experience maturity model: A progress report. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 5619 LNCS, p. 104–109, 2009.

[10] STAGGERS, N.; RODNEY, M. **Promoting usability in organizations with a new health usability model: implications for nursing informatics.** International Congress on Nursing Informatics. **Anais**...Montreal: American Medical Informatics Association, 2012.

[11] FORCE, H. U. T. Promoting Usability in Health Organizations: initial steps and progress toward a healthcare usability maturity model. **Health Information and Management Systems Society**, 2011.

[12] CHAPMAN, L.; PLEWES, S. A UX maturity model: Effective introduction of UX into organizations. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8520 LNCS, n. PART 4, p. 12–22, 2014.

[13] KIEFFER, S.; VANDERDONCKT, J. **STRATUS : A Questionnaire for Strategic Usability Assessment**. ACM Symposium on Applied Computing. **Anais**...Pisa: ACM, 2016.

[14] EARTHY, J.; SHERWOOD-JONES, B. Human Factors Integration Capability Maturity Model--Assessment model. In: **Presented at Human Interfaces in Control Rooms, (pp. 320-236)**. IET, 2000.

[15] ISO/IEC. **ISO 18529 - Ergonomics -Ergonomics of human- system interaction - Human-centred lifecycle process descriptions.** Geneva, 2000.

[16] SWARD, D.; MACARTHUR, G. **Making user experience a business strategy**. E. Law et al.(eds.), Proceedings of the Workshop on Towards a UX Manifesto. **Anais**...Lancester. 2007.

[17] VAN TYNE, S. **Corporate user-experience maturity model**. International Conference on Human Centered Design. **Anais**...San Diego: Springer. 2009.

[18] MOSTAFA, D. Maturity Models in the Context of Integrating Agile Development Processes and User Centred Design. n. June, 2013.

[19] ISO/IEC. **ISO/IEC 18152 - Ergonomics of human-system interaction - Specification for the process assessment of human-system issues**. Geneva. 2010.

[20] PERES, A. L. ET AL. **AGILEUX Model: Towards a reference model on integrating UX in developing software using agile methodologies**. Agile Conference (AGILE), 2014. **Anais**...Lisbon: IEEE. 2014.

[21] VASMATZIDIS, I.; RAMAKRISHNAN, A.; HANSON, C. **Introducing usability engineering into the Cmm model: an empirical approach**. Human Factors and Ergonomics Society Annual Meeting. **Anais**...Los Angeles: SAGE Publications. 2001.

[22] SALAH, D.; PAIGE, R. **A Framework for the Integration of Agile Software Development Processes and User Centered Design (AUCDI)**. 6th International Symposium on Empirical Software Engineering and Measurement, Lund, Sweden. **Anais**... New York: IEEE. 2012.

[23] SY, D. Adapting usability investigations for agile user-centered design. **Journal of usability Studies**, v. 2, n. 3, p. 112–132, 2007.

[24] RATCLIFFE, L.; MCNEILL, M. **Agile experience design: A digital designer's guide to agile, lean, and continuous**. San Franscisco: New Riders, 2011.

[25] BROWN, D. **Agile user experience design: a practitioner's guide to making it work**. Burlington: Newnes, 2012.

[26] METTLER, T. Maturity assessment models: a design science research approach. **International Journal of Society Systems Science**, v. 3, n. 1–2, p. 81–98, 2011.

[27] MARCH, S. T.; SMITH, G. F. Design and natural science research on information technology. **Decision support systems**, v. 15, n. 4, p. 251–266, 1995.

[28] JÄRVINEN, P. **On Research Methods. Opinpajan Kirja, Tampere, 2004**.

[29] SHERWOOD-JONES, B. Total Systems Maturity. **Project report, Glasgow, UK**, 1995.

[30] SCHAFFER, E. **Institutionalization of usability: a step-by-step guide**. Boston: Addison-Wesley Professional, 2004.

[31] Nielsen, J. *Corporate UX Maturity*. 2006, April. Retrieved from: http://www.useit.com/alertbox/maturity.html. Accessed at: 23 nov. 2017.

[32] Brennan, C., Earthy, J. and Jennings, D. Draft HCI Functions for BCS Industry Structure Model (version 3). Human Factors Unit, BT Research Labs, Martlesham Heath. 1995.

[33] MCCLELLAND, I. et al. Humanware Process Improvement - institutionalising the principles of user centred design. p. 1–6, New Jersey: Prantice Hall. 1998.

[34] HADDAWAY, N. R. et al. The role of Google Scholar in evidence reviews and its applicability to grey literature searching. **PloS one**, v. 10, n. 9, 2015.

9.2 REFERENCES FROM SECTION 4.2

[1] GARCIA, I.; PACHECO, C.; CRUZ, D. Adopting an RIA-Based Tool for Supporting Assessment, Implementation and Learning in Software Process Improvement under the NMX-I-059/02-NYCE-2005 Standard in Small Software Enterprises. **Software Engineering Research, Management and Applications**, p. 29–35, 2010.

[2] GRADEN, C.; NIPPER, D. An Innovative Adaptation of the EIA/IS 731.2 Systems Engineering Capability Model Appraisal Method. **INCOSE International Symposium**, p. 590–594, 2000.

[3] MULADI, N.; SURENDRO, K. **The readiness self-assessment model for green IT implementation in organizations**. International Conference of Advanced Informatics: Concept, Theory and Application. **Anais** IEEE, pp. 146-151, 2014.

[4] GLANZNER, R. A.; AUDY, J. L. N. 2DAM-WAVE: An evaluation method for the WAVE capability model. **International Conference on Global Software Engineering, ICGSE,** IEEE, p. 75–83, 2012.

[5] WIDERGREN, S. et al. Smart grid interoperability maturity model. **Power and Energy Society General Meeting**, IEEE, p. 1–6, 2010.

[6] BURNSTEIN, I. et al. A model to assess testing process maturity. **Crosstalk The Journal of Defense Software Engineering**, v. 11, n. 11, p. 26–30, 1998.

[7] GRCEVA, S. Software Process Self-Assessment Methodology. **TEM Journal**, v. 1, n. 2, p. 65–71, 2012.

[8] AMARAL, L. M. G.; FARIA, J. P. A Gap Analysis Methodology for the Team Software Process. **Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the**, p. 424–429, 2010.

[9] SERRANO, M. A.; DE OCA, C. M.; CEDILLO, K. **An experience on using the team software process for implementing the Capability Maturity Model for software in a small organization**. Third International Conference on Quality Software. **Anais IEEE,** pp. 327-334, 2003.

[10] BOLLINGER, P.; MILLER, D. Internal Capability Assessments**. INCOSE International Symposium** , v. 11, n. 1, p. 813–817, 2001.

[11] SHRESTHA, A. et al. Evaluation of software mediated process assessments for IT service management. **Communications in Computer and Information Science**, v. 526, p. 72–84, 2015.

[12] BLANCHETTE JR, S.; KEELER, K. L. **Self Assessment and the CMMI-AM-A Guide for Government Program Managers**. Carnegie-Mellon Univ. Pittsburgh Pa Software Engineering Inst. 2005.

[13] WIEGERS, K.; STURZENBERGER, D. A modular software process mini-assessment method. **IEEE Software**, v. 17, n. 1, p. 62–69, 2000.

[14] YUCALAR, F.; ERDOGAN, S. Z. A questionnaire based method for CMMI level 2 maturity assessment. **Journal of aeronautics and space technologies**, v. 4, n. 2, p. 39–46, 2009.

[15] KASURINEN, J. et al. A self-assessment framework for finding improvement objectives with ISO/IEC 29119 test standard. **European Conference on Software Process Improvement**, p. 25–36, 2011.

[16] KARVONEN, T. et al. **Adapting the lean enterprise self-assessment tool for the software development domain**. 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA). **Anais IEEE**, pp.266-273, 2012.

[17] VARKOI, T. **Process Assessment In Very Small Entities-An ISO/IEC 29110 Based Method**. Seventh International Conference on the Quality of Information and Communications Technology. **Anais** IEEE, pp.436-440, 2010.

[18] CATER-STEEL, A. et al. **Software-mediated process assessment in IT service management**. International Conference on Software Process Improvement and Capability Determination. **Anais**, pp. 1- 10, 2013.

[19] BÖCKING, S.; MAKRIDAKIS, P.; KOLLER, G. & MEISGEN, F. (2005). A Lightweight Supplier Evaluation based on CMMI. *GI Jahrestagung,* 2, 274-278.

[20] PATEL, C.; RAMACHANDRAN, M. Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices. **International Journal of Software Engineering**, v. 2, n. 1, p. 3–28, 2009.

[21] PINO, F. J. et al. Assessment methodology for software process improvement in small organizations. **Information and Software Technology**, v. 52, n. 10, p. 1044–1061, 2010.

[22] TIMALSINA, A.; THAPA, P. **Assessment of Software Process Improvement Based on Capability Maturity Model Integration**. Conference: Technology and Innovation Management. **Anais**. 2016.

[23] GOBEL, H.; CRONHOLM, S.; SEIGERROTH, U. **Towards an agile method for ITSM self-assessment: A Design Science Research Approach**. Proceedings of the International Conference on Management, Leadership and Governance (ICMLG 2013). **Anais**. Pp.135-142, 2013.

[24] HOMCHUENCHOM, D. et al. SPIALS: A light-weight software process improvement self-assessment tool. **2011 5th Malaysian Conference in Software Engineering, MySEC 2011**, p. 195–199, 2011.

[25] ORCI, T.; LARYD, A. **Dynamic CMM for Small Organizations: Implementation Aspects**. European Software Process Improvement Conference. **Anais**. Copenhagen: 2000.

[26] DAILY, K.; DRESNER, D. Towards software excellence-informal self-assessment for software developers. **Software Process: Improvement and Practice**, v. 8, n. 3, p. 157–168, 2003 DAILY, K.; DRESNER, D. Towards software excellence-informal self-assessment for software developers. **Software Process: Improvement and Practice**, v. 8, n. 3, p. 157–168, 2003.

[27] COALLIER, F. et al. Trillium Model for Telecom Product Development & Support Process Capability, Release 3.0. **Internet edition**, 1994.

[28] MACMAHON, S. T.; MC CAFFERY, F.; KEENAN, F. **Development and validation of the MedITNet assessment framework: improving risk**

management of medical IT networks. Proceedings of the 2015 International Conference on Software and System Process. **Anais ACM**, pp.17-26, 2015.

[29] KAR, S. et al. **Self-assessment Model and Review Technique for SPICE: SMART SPICE**. International Conference on Software Process Improvement and Capability Determination. **Anais**, pp.222-232, 2012.

[30] RAZA, A.; CAPRETZ, L. F.; AHMED, F. An open source usability maturity model (OS-UMM). **Computers in Human Behavior**, v. 28, n. 4, p. 1109–1121, 2012.

[31] RAPP, D. et al. **Lightweight requirements engineering assessments in software projects**. 22nd International Requirements Engineering Conference (RE). **Anais** IEEE, pp.354-363, 2014.

[32] ABUSHAMA, H. M. PAM-SMEs: process assessment method for small to medium enterprises. **Journal of Software: Evolution and Process**, v. 28, n. 8, p. 689–711, 2016.

[33] KUVAJA, P.; PALO, J.; BICEGO, A. TAPISTRY: a software process improvement approach tailored for small enterprises. **Software Quality Journal**, v. 8, n. 2, p. 149–156, 1999.

APPENDIX A Assessment briefing script

Present the purpose of the assessment:
"The purpose of this meeting is to assess the usability process of the organization in order to identify its strengths and weaknesses, and in this way to be able to initiate a program to improve our process."

Explain how the assessment will be performed:
"To perform the assessment, we will use the UPCASE method, which has an online tool, which will be used to conduct this assessment and generate the results.
The assessment should not last longer than one hour. The assessment of the process consists in making an "Assessment poker", which will be carried out as follows:
I will act as moderator, I will read the description of each usability sub-process. I will then read each of the items in the questionnaire regarding this sub-process. I will also present the descriptions and examples of work product and techniques that can be used to generate them. We will reflect if we think that the item is:

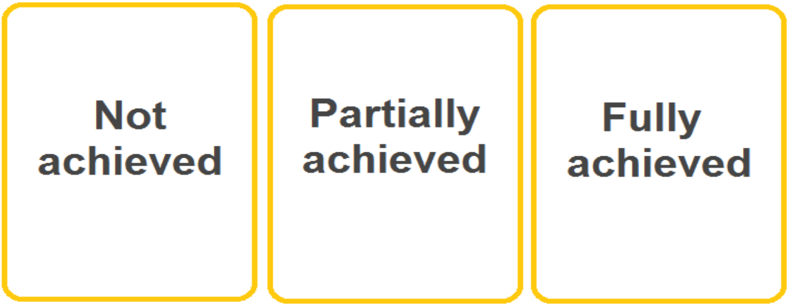
Not achieved (if we think that the item that was read is not carried out in our projects).
Partially achieved (if we think that the item is carried out sometimes in our projects).
Fully achieved (if we think that the item is always carried out in our projects).

Then we will all present at the same time the card that contains our opinion on the item.
If there are different answers, each participant must justify his / her choice. Then the cards are played again.

This process must be repeated, until it reaches consensus on the answers.
If the consensus is that the item is not achieved, the answer is marked on the questionnaire and we proceed to the next item in the questionnaire.
If the consensus is reached and the item is partially achieved or fully achieved, we should provide an example of activity that confirms the item is achieved.
If an example is provided, then the questionnaire item can be marked based on the voting, otherwise the item should be marked as "Not achieved".
After performing this process for all 16 items of the questionnaire, the UPCASE Tool will generate the score of our usability process, as well as elicit the points that can be improved."
Confirm that all participants understood the purpose of the assessment and how it will be carried out.
Give a card of "Not achieved", "Partially achieved" and "Fully achieved" for each participant and start the assessment.

APPENDIX B Assessment poker deck of cards

Figure 32 - Assessment poker cards



APPENDIX C Examples of work products from UPCASE processes

Table 54 - UPCASE process work products

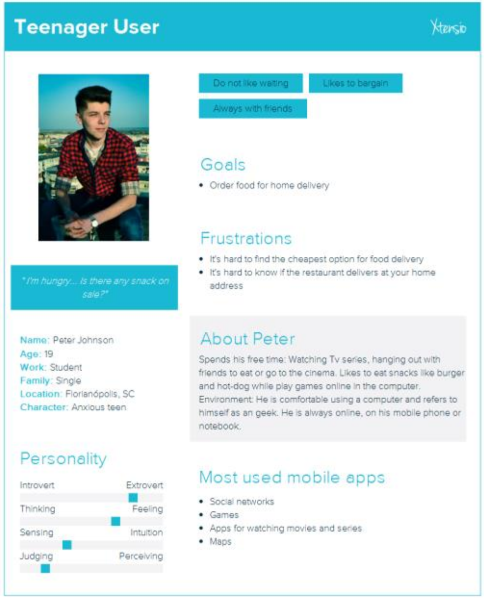
N Process	N Practice	Work products of ISO/IEC 18529	Customized work products	Example of Work product
UP1	1	-	Purpose(s) of the system	Example of system purpose: - Order food for home delivery
	2	Stakeholder/User Requirements Specification	A list of system performance and behavior requirements.	Examples of user performance and behavior requirements: - System should be easy to install, easy to read in lighting and noise environments, easy to use by person with low vision capability.
	3	A statement of the human-centred design goals	A list of usability requirements	Examples of usability requirements: Order a meal from the mobile application: - Maximum time to complete task: 2 minutes - Minimum percentage of users who can complete the task: 95% - All users must assess the system with at least 80 points in the SUS questionnaire.
UP2	4	Specification of the range of intended users, tasks and environments Task information	A list of user tasks and their characteristics and a list of use cases.	**List of user tasks:** - Register account - Login - Search restaurant - See restaurant menu - Add meals to cart - Order meal.

				Example of task characteristics: **Task** / **Order food** Frequency of use: Variable (daily, weekly, monthly) Duration of the task: 2-3 min Task Flexibility: Ordered food (pizza, hamburger, hot-dog), amount of food portion, delivery address. Alternative events: Delivery address do not exist Physical and mental demands: Low Task dependencies: User account in the system, internet access. Task result: The restaurant receives the request for food delivery Alternative events: Restaurant is closed at time. **Example of "use case":**

Nested table (from the last cell above):

Task	Order food
Frequency of use	Variable (daily, weekly, monthly)
Duration of the task	2-3 min
Task Flexibility	Ordered food (pizza, hamburger, hot-dog), amount of food portion, delivery address.
Alternative events	Delivery address do not exist
Physical and mental demands	Low
Task dependencies	User account in the system, internet access.
Task result	The restaurant receives the request for food delivery
Alternative events	Restaurant is closed at time.

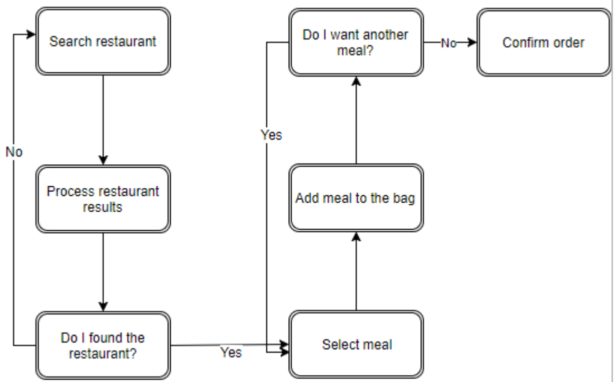
				Use case Use Case Name: Order Meal Actors: Customer Description: A customer accesses the Meal Delivery System using his mobile phone, searches for a specific restaurant, selects food items, and places an order meal to be delivered to a specified location. Preconditions: 1. Customer is logged into Meal Delivery System. 2. Customer already found the intended restaurant. Postconditions: 1. Meal order is stored in the system with a status of "accepted". Normal Flow: **1.0 Order a Meal** 1. Customer asks to view the menu. 2. System displays menu of available food items. 3. Customer selects one or more food items from menu. 4. Customer indicates that meal order is complete. 5. System displays ordered menu items, individual prices, and total pric and delivery charge. 6. Customer confirms meal order or requests to modify meal order (bac step 3). 7. Customer specifies payment method. 8. System confirms acceptance of the order. 9. System stores order in database and notify the restaurant Cafeteria S sending the food item information. Alternative Flows: **1.1 Order multiple meals** (branch after step 4) 1. Customer asks to order another meal. 2. Return to step 2.
	5	Stakeholder information User information	User information, user profile, personas	**Example of persona:**

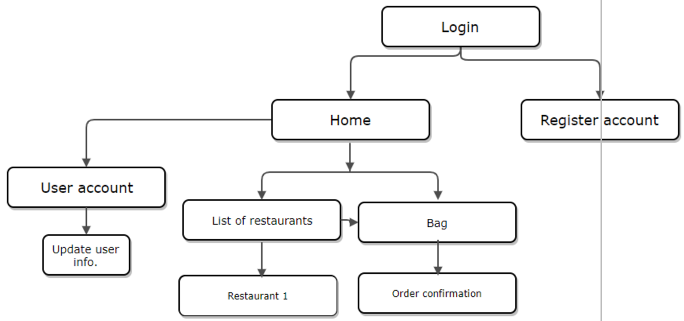
				 **Example of user profile:**

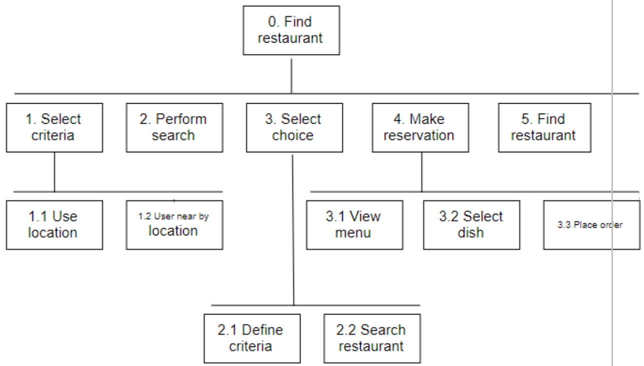
<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><th colspan="3">User profile</th></tr>
<tr><th>Personal attributes</th><th>Young</th><th>Adult</th></tr>
<tr><td>Age</td><td>15-30 years old</td><td>31-50 years old</td></tr>
<tr><td>Gender</td><td>50% male and 50% female</td><td>50% male and 50% female</td></tr>
<tr><td>Physical disabilities</td><td>May have some visual disability.</td><td>May have some visual disability.</td></tr>
<tr><td>Level of education</td><td>High school or graduation</td><td>High school or graduation</td></tr>
<tr><td>Competence in the use of the device</td><td>A lot of experience using mobile phones.</td><td>Have some experience using mobile phones.</td></tr>
<tr><td>Motivation</td><td>They have a lot of motivation to use the app.</td><td>They have a motivation to use the app.</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Organisational analysis
The sources from which the user and organisational requirements were derived</td>
<td>Description of social and organizational environment, management structure, communications and organizational practices or legislation.</td>
<td>Example of social environment characteristics:</td>
</tr>
</table>

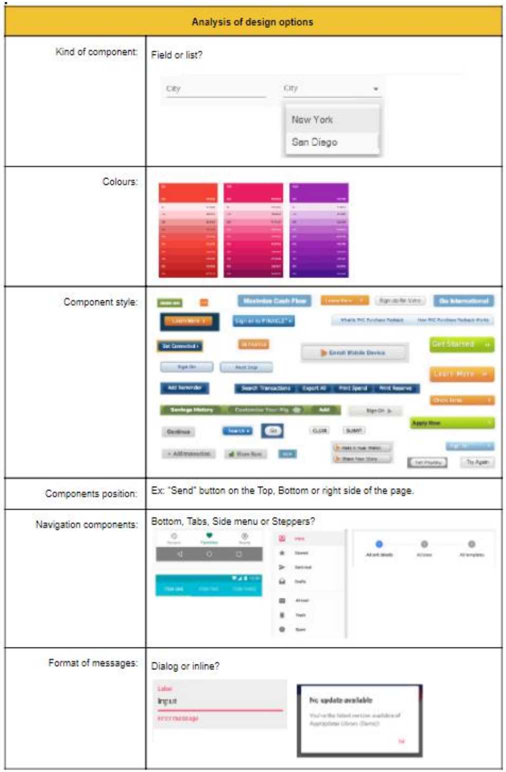
<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th colspan="2">Organizational characteristics</th></tr>
<tr><td>Use in group</td><td>Yes</td></tr>
<tr><td>Assistance</td><td>No assistance will be provided while using the app, but the app provides a tutorial.</td></tr>
<tr><td>Interruptions</td><td>The user is subject to interruptions during the use of the system, having to pause the activity that is being carried out to resume later.</td></tr>
<tr><td>Discretion</td><td>May decide to call instead use the app, if the user think the app is too complicated or time consuming,</td></tr>
</table>
</td></tr>
<tr><td></td><td>7</td><td>--</td><td>Analysis of the device characteristics</td><td>Example of device characteristics:</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th colspan="2">Device characteristics</th></tr>
<tr><td>Device</td><td>Mobile phone</td></tr>
<tr><td>Platform</td><td>Android</td></tr>
<tr><td>Weight:</td><td>4 - 6 Ounces</td></tr>
<tr><td>Height:</td><td>5 - 7 Inches</td></tr>
<tr><td>Width:</td><td>2.5 - 3.5 Inches</td></tr>
<tr><td>Keyboard:</td><td>Touch Only</td></tr>
<tr><td>Display Size:</td><td>4.0 - 7.0 Inches</td></tr>
<tr><td>Front Camera MP:</td><td>4.00 - 10.00 Megapixels</td></tr>
<tr><td>Rear Camera MP:</td><td>9.00 - 16.00 Megapixels</td></tr>
<tr><td>Battery Standby:</td><td>200 -300 Hours</td></tr>
<tr><td>Internal Memory</td><td>8.00 - 32.00 GB</td></tr>
<tr><td>Features:</td><td>GPS, accelerometer</td></tr>
</table>
</td></tr>
<tr><td></td><td>8</td><td>--</td><td>Description of physical environment characteristics</td><td>Example of physical environment characteristics:</td></tr>
</table>

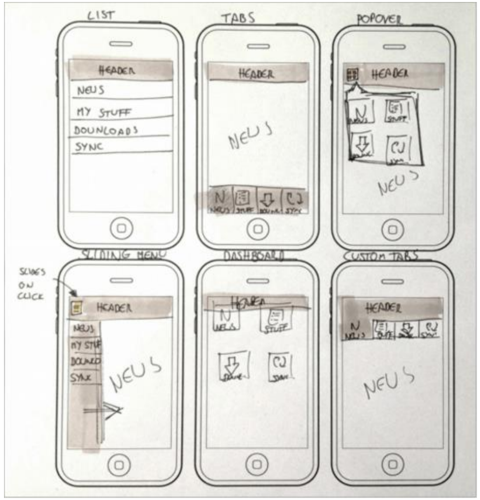
<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><th colspan="2">Physical environment characteristics</th></tr>
<tr><td>Atmospheric conditions</td><td>The app may be used indoors.</td></tr>
<tr><td>Acoustic Environment</td><td>The app may be used in an environment that can be noisy.</td></tr>
<tr><td>Visual environment</td><td>The app may be used in an environment with sunlight or artificial lighting.</td></tr>
<tr><td>User Posture</td><td>The user may be sitting in a sofa or a chair</td></tr>
<tr><td>Health risks</td><td>The use of the app does not imply a risk to the health of the user.</td></tr>
</table>
</td>
</tr>
<tr>
<td>UP3</td>
<td>9</td>
<td>Task model
Worksystem design</td>
<td>Identification of main system screens, flowchart task, navigation diagram, task hierarchy.</td>
<td>Example of task flowchart:

</td>
</tr>
</table>

				Example of navigation diagram:  **Example of task hierarchy analysis:**

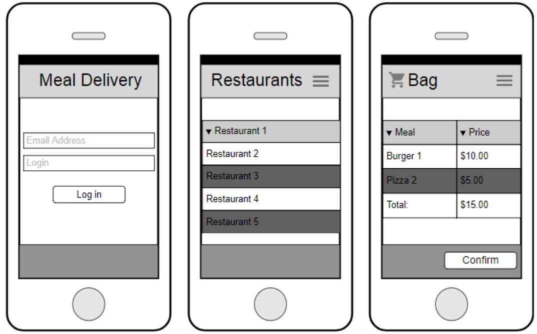
<table>
<tr><td></td><td></td><td></td><td></td><td>
</td></tr>
</table>

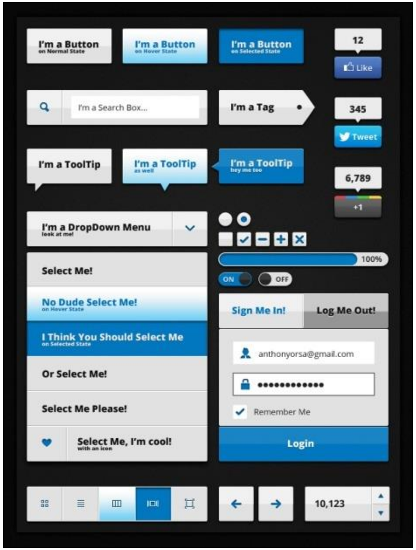
	10	--	Analysis of design options	

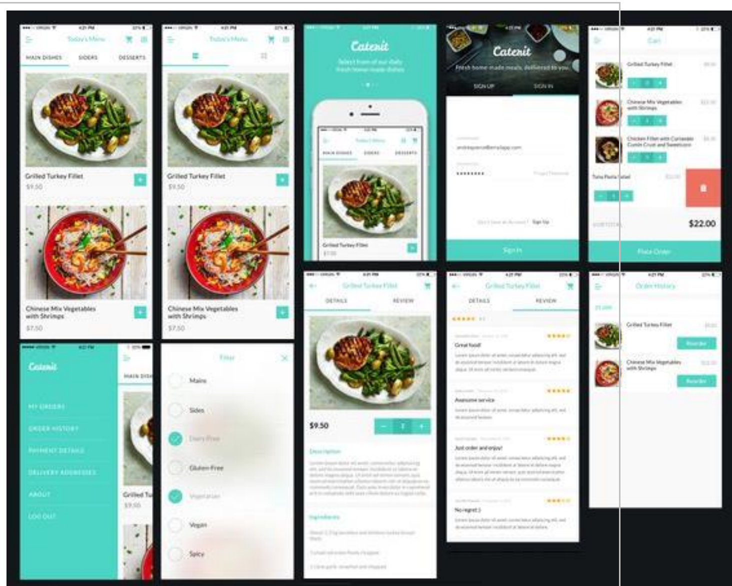
	11	List of standards used and how applied The sources of existing knowledge and the standards used, with an indication of how they have been incorporated (or why they have not been followed, if appropriate). Means of feedback and use of results in other design activities Evidence of revision in accordance with results of evaluations	--	--
	12	User Interaction Specification	Wireframes, sketches, specification of system components behavior, storyboards.	**Example of sketches:**

<table>
<tr><td></td><td></td><td></td><td></td><td>

Example of wireframe:</td></tr>
</table>

				
	13	Dialogue detail Look and feel Layout and other UI issues Prototype(s) of parts and all of the system Simulations of specification	Definition of the “look and feel”, high fidelity prototypes, and detailed user interface design specification.	**Example of definition of the "look and feel":**

				 (Source: http://www.vectorarea.com/free-clean-blue-gloss-ui-kit-ui-kit) **Example of high fidelity prototype:**

				 (Source: https://br.pinterest.com/pin/323625923204182489/)
UP4	14	Which parts of the system are to be evaluated and how they are to be evaluated Context of evaluation Number of users taking part in testing, including evidence of adequacy of	Test script, test case, evaluation goals, metrics to be collected	**Where, who, when:** The test will be performed in organization meeting room, with 5 persons that represent the app user profile, Monday, 5 July at 2 pm. **Example of test script:** Welcome the participant and make him feel at ease; Reading the introduction script;

		number of users and their representativeness of those identified in the context of use Survey plan Source of evaluation feedback Trial plans and records Testing and data collection methods, including evidence of appropriateness of these methods for the system and context use		Presentation and signature of the free and informed consent term; Raise demographic information (by interview or by requesting the completion of a background questionnaire); Ask the participant if he or she is already familiar with the device on which the test will be performed or if they wish to undergo a short training; If necessary, perform the training with the device before performing the test with it; Present and Deliver the list with the tasks to the participant; Participants perform the tasks (collect data in parallel for example via web cam / researcher notes) Apply the satisfaction questionnaire; Ask if the participant would like to make any further comments about the system; Thank the participant for participating in the test. **Example of test case: Order meal in the Food delivery app** Peter came home from football practice. He is tired and hungry, so he opens his food delivery app to find out if he has any meal deal at moment. He chooses the meal deal and asks to have it delivered to his house. Using the cell phone, access the app with the data from Peter's account and order a meal deal. **Example of evaluation goals:** Measure the extent with the system can be used with effectiveness, efficiency and satisfaction. **Example of metrics to be collected:** Time to complete a task; Number of errors committed; User degree of satisfaction.

	15	Usability and ergonomic defects Recommendations for improvement Video and audio tapes from trials User observation logs Measurements of ergonomic parameters A report of major and minor non-compliances and observations and an overall assessment Survey criteria Survey report	List of recommendation changes, list of usability problems found, violated heuristics, degree of severity of the usability problem.	**Example of usability problems:** - "The page has no title". - "The same symbol is used in two different icons". **Example of recommendations changes:** - "Add title in all pages". - "Use different symbols for icons with different functions". **Example of violated heuristics:** - "Visibility of System Status: The system do not inform the user if operation has finished or not. Severity: Major usability problem: important to fix, so should be given high priority." - "User control and freedom: The screen does not contain an option for the user to cancel the operation. Severity: Minor usability problem: fixing this should be given low priority."
	16	A clear pass/fail decision in relation to the requirements Revisions to requirements Full description of the system tested and its status Simulations of specification	A pass/fail decision regarding each requirement.	Example of a pass/fail decision: - Requirement: Maximum time to complete task: Pass. All test participants took less than 1,5 minutes. - Requirement: Minimum percentage of users who can complete the task: Fail. Only 80% of the test participants manage to complete de task. - Requirement: All users must assess the system with at least 80 points in the SUS questionnaire: Fail. Only 70% of the participants assess the system with more than 80 points in SUS questionnaire.

APPENDIX D Practices descriptions and indicators from UPCASE

Table 55 - Practices descriptions and indicators

N Process	N Practice	Practice	Description	Indicator	Example of techniques	Example of work products
UP1	1	Identify system purpose	Identify and describe the purpose of the system, this is, the objective(s) that the user wants to achieve using the system.	Our team identifies and describes the purpose of the system.	Survey, brainstorming, interview, observation.	Purpose(s) of the system
	2	Define system performance and behavior requirements desired by the user.	Identify the stakeholder's requirements regarding the behavior and performance of the system. The requirements cover each aspect of the system related to its use and its interface in a context of use.	Our team identifies system performance and behavior requirements desired by the user.	Survey, brainstorming, interview, observation.	System performance and behavior requirements desired by the user.
	3	Define usability requirements.	Define an explicitly statement for each usability requirements, regarding its effectiveness, efficiency and user satisfaction based on the context of use analysis. The statements should be measurable objectives.	Our team defines explicit statements of usability requirements based on the context analysis.	Benchmarking with concurrent systems, synchronic analyzes formal work analyses.	A list of usability requirements

UP2	4	Identify and describe the user's tasks of the system	Describe the tasks the users need perform in the system in order to achieve their goals.	Our team identifies and describes the characteristics of the tasks the user performs in the system.	Survey, interview, observation, formal work analysis, brainstorming with user's task, user's tasks modeling.	A list of user tasks and their characteristics and a list of use cases.
	5	Identify user characteristics	Identify relevant characteristics of the users, such as knowledge about the system domain, degree of literacy, physical capabilities, level of experience with the tasks and with the device he will use to interact with the system, motivations in using the system, etc.	Our team identifies and describes the characteristics of the users.	Survey, interview, observation.	User information, user profile, personas
	6	Identify social environment characteristics	Identify relevant social and organizational milieu, management structure, communications and organizational practices in the environment in which the system will be used.	Our team identifies and describes the organizational and social characteristics regarding the environment in which the system will be use.	Survey, observation, interview.	Description of social and organizational environment, management structure, communications and organizational practices or legislation.
	7	Identify device characteristics	Identify relevant characteristics of the device with which the users will directly interact, such as memory and process capacity, ways of input and output data, screen size, etc.	Our team identifies and describes the characteristics of the device with which the users will interact to use the system.	Review platform documentation.	Analysis of the device characteristics
	8	Identify physical environment characteristics	Identify relevant characteristics of the location, workplace equipment and ambient conditions and its implications for the system design, such as lighting, noise levels,	Our team describes the physical environment characteristics in which the system will be used.	Survey, observation, interview.	Description of physical environment characteristics

			vibration, heat, hazards, dimensions of working and living space.			
UP3	9	Analyze user's tasks	Analyze the user's tasks in terms of alternative navigation pathways and flowcharts and identifying the main system screens and constraints.	Our team analyzes the use cases in terms of its flow, navigation, main screens and constraints.	Conceptual model design, wireframes development, navigation definition, task hierarchy analysis, information architecture definition, card sorting.	Identification of main system screens, flowchart task, navigation diagram, task hierarchy.
	10	Develop and analyze design options during interface development	Analyze a range of design options for each aspect of the system related to its use and its effect on stakeholders, such as definition of system controls, location and format of display components, use of colors, terminology, fonts, and wording of messages.	Our team analyzes a range of design options for each aspect of the system related to its use and its interface during the system development.	Wireframe development, sketches development, storyboarding.	Analysis of design options
	11	Develop design solutions using existing knowledge	Applied existing usability knowledge, such as stakeholder requirements, information about the context of use, international standards, usability good practice and style guides to the design of the system and is used to select the appropriate alternatives of design.	Our team applies existing usability knowledge (such as stakeholder requirements, usability guidelines) in the system design.	--	--

	12	Specify all user-related elements of the system	Specified the design of all the user-related components of the system. This specification is a description of how the components and the system will be used, such as the kind of systems controls will be used, location and format of display components, colors, terminology, fonts, wording of messages).	Our team specified each aspect of the system related to its use and its interface.	Wireframe and sketch development, storyboarding.	Wireframes, sketches, specification of system components behavior, storyboards.
	13	Prototype all user-related elements of the system	Refine design through the development of high-fidelity prototypes of all aspect of the system related to its use and its interface.	Our team develops high-fidelity prototypes each component of the system interfaces.	Prototype development.	Definition of the "look and feel", high fidelity prototypes, and detailed user interface design specification.
P4	14	Prepare prototype/system evaluation	Prepared and defined all arrangements necessary to evaluate the prototype or the system, such as definition of which evaluation method will be used, who will be the assessor, place, scripts, questionnaires, cameras and etc.	Our team prepares the prototypes and system evaluation.	Prepare usability evaluation necessary materials, select participants.	Test script, test case, evaluation goals, metrics to be collected.
	15	Evaluate prototypes and system to find usability problems	Prototypes are evaluated against usability knowledge, style guides, standards, guidelines in order to find usability problems and verify if the required good practices have been followed.	Our team evaluates the usability of the prototypes.	Usability test with users, heuristic evaluation, cognitive walkthrough, key level stroke model evaluation.	List of recommendation changes, list of usability problems found, violated heuristics, degree of severity of the usability problem.
	16	Evaluate system against usability requirements	System is evaluated to ensure that it meets the requirements of the users,	Our team evaluates the system in order to check if it meets the usability requirements.	Usability test with users.	A pass/fail decision regarding each requirement

			the tasks and the environment, as defined in its specification.			

APPENDIX E UPCASE glossary

Table 56 - Glossary

Concept	Definition	Reference
Process	A set of interrelated activities, which transform inputs into outputs.	ISO/IEC 18529
Practice	A technical or management activity that contributes to the creation of the output (work products) of a process or enhances the capability of a process.	ISO/IEC 18529
Work product	A document, piece of information, product or other item which acts as input to or output from a process	ISO/IEC 18529
Indicator	Sources of objective evidence used to support the assessors' judgment in rating process attributes	ISO/IEC 29110
Usability requirement	Usability requirements define the intended objectives and context of use and specifies levels of measures and criteria for effectiveness, efficiency and satisfaction for the product under development	ISO/IEC 9142-11
Use case	A use case is all the ways of using a system to achieve a particular goal for a particular user. Taken together the set of all the use cases gives you all of the useful ways to use the system.	(Jacobson, I., 2011)
User characteristic	Is a general description of a user group of specific software. Typically includes characteristics that may influence design choices, such as: demographic characteristics, education, language, computer expertise, domain experience, motivation, or expectations.	(Human Factors International, 2014).
Social environment characteristic	Describe the relevant social milieu, management structure, communications and organizational practices. At a lower level it describe the structure of the organization, the way people use the system, individually and in groups, the availability of assistance and the frequency of interruptions, political and interpersonal	(Maguire, M., 2001) ISO/IEC 18529

	factors, degree of freedom, influence in decision-making.	
Physical environment characteristic	Characteristics of the physical environment in which the users will interact with the system, such as the physical environment can have a profound effect on the usability of a product. Bad lighting or loud noise in the place may prevent the users from receiving feedback from the system.	(Maguire, M., 2001)
Task characteristic	Overview of a given task outlining its characteristics that impact usable design, including importance, frequency, sequence, dependency and flow, criticality.	(Human Factors International, 2014).
Design options	Design options are artifacts that present design alternatives for each aspect of the system related to its use, such as the definition of system controls, location and format of display components, use of colors, terminology, fonts, and wording of messages.	ISO/IEC 18529
Prototype	Representation of all or part of an interactive system that, although limited in some way, can be used for analysis, design and evaluation.	ISO/IEC 9241-110
High-fidelity prototype	Representation of all or part of an interactive system that is typically quite close to the final product, with all (or almost all) elements of the screen detailed and refined in relation to their positioning, size, color and shape.	(Usability First, 2015)
Low-fidelity prototype	Representation of all or part of an interactive system that is typically made in paper, slides, or other non-interactive mock-ups of an interface developed early in design. Typically, do not contain too much detail about the look and feel of the screen elements.	(Human Factors International, 2014).
Usability problem	An aspect of the system which makes it unpleasant, inefficient, onerous or impossible for the user to achieve their goals in typical usage situations.	ISO/IEC 9241-110
Stakeholder	Individual or organization having a right, share, claim or interest in a system or in its	ISO/IEC 9241-110

	possession of characteristics that meet their needs and expectations.	
Context of use	Evolves the users, tasks, technical (hardware, software and materials), and the physical and social environments in which a product is used.	ISO/IEC 9241-110
Effectiveness	Accuracy and completeness with which users achieve specified goals.	ISO/IEC 9241-110
Efficiency	Represents the amount of effort or resources expended in relation to the accuracy and completeness with which users achieve goals.	ISO/IEC 9241-110
User satisfaction	The freedom from discomfort and positive attitudes towards the use of the product.	ISO/IEC 9241-110
Usability	Extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use	ISO/IEC 9241-110
User task	Activity required to the user achieving a certain goal in the system.	ISO/IEC 9241-110
Flowchart	Visual way of representing a task or procedure. Steps of a process are represented in boxes and flow is represented by arrows connecting the boxes. Input and output are typically represented in skewed parallelograms, and decision points are usually represented with diamonds.	(Usability First, 2015)
Style guide	Reference that establishes the look-and-feel of a user interface by defining the its standards and conventions. It usually includes the principles that guide the design of the interface, graphic layout grids, exact size and spacing of elements in the interface, fonts, colors, interactive behavior and standard text messages (such as error messages).	(Usability First, 2015)
Heuristic evaluation	Technique for finding usability problems in a user interface. A small number of trained evaluators (typically 3 to 5) individually inspect a user interface by applying a set of “heuristics”, broad guidelines that are generally relevant. They	(Usability First, 2015)

	then combine their results and rank the importance of each problem to prioritize fixing each problem.	
Cognitive walkthrough	Technique to evaluate a user interface based on stepping through common tasks that a user would need to perform and evaluating the user's ability to perform each step (e.g., "How many users will click this button for task A? What happens when they do?"). This approach is intended specially to help understand the usability of a system for first-time or infrequent users, that is, for users in an exploratory learning mode.	(Usability First, 2015)
Key level stroke model	Technique to predict how long it will take a user to accomplish a task without errors using a system. KLM defines an estimated time to execute each operator (typing a character, pointing with the mouse, clicking the mouse, etc.). KLM estimate the task execution time by listing the sequence operators required to perform a user task and then summing the times of the individual operators.	(Card et al., 1980)
Brainstorming	Brainstorming is a method for generating group creativity. A group of people come together and focus on a problem or proposal. There are two phases of the activity. The first phase generates ideas, the second phase evaluates them.	(Usability Net , 2006)
Interview	One-on-one interactions between end-users and usability analysts, designed to elicit the users' conceptual model of a system, the tasks and task flows, or other issues related to design	(Human Factors International, 2014).
Observation	Method in each an investigator view user as they work in a field study, and taking notes on the activity that takes place. Observation may be either direct, where the investigator is actually present during the task, or indirect, where the task is viewed by some other means such as through use of a video recorder.	(Preece, J. et al., 1994)

Conceptual model design	Is a model constructed by the users in their mind to understand the working or the structure of objects, based on their mental model and previous experience, to speed up their understanding. Humans establish mental models of how things work, or how they would behave in a particular situation.	(Human Factors International, 2014).
Wireframe	A wireframe is a two-dimensional illustration of a screen interface that specifically focuses on space allocation and prioritization of content, functionalities available, and intended behaviors. For these reasons, wireframes typically do not include any styling, color, or graphics.	(Usability Gov., 2017)
Information architecture	Is an activity of the conceptual design stage associated with defining the system content. Includes the processes of defining the system screens hierarchies, content organization, and labeling schemes for all types of menu systems, and the techniques for creating and evaluating them	(Human Factors International, 2014).
Navigation pathway	Based on task design and information architecture definitions developed in conceptual design, navigation design marks the first formal step of design. It includes the development of wire frames and prototypes to test the design structure and aesthetic. A set of core navigation screens are designed, tested, and iterated during this stage to ensure that the user interface structure is sound before investing in detailed design.	(Human Factors International, 2014).
Task hierarchy analysis	Activity in which the hierarchy of the user tasks is analyzed. The task hierarchy is an organization of elements that, according to prerequisite relationships, describes the path users must take to achieve any single behavior that appears higher in the hierarchy. Thus, in a hierarchical analysis, the designer decomposes a task from top to bottom, thereby, showing the hierarchical relationship amongst the tasks in a bottom up order.	(Stanley, T., 1999)

Sketch	Simply or hastily drawing giving the essential features of the system without the details. It excludes the level of detail that goes into the final product.	(Human Factors International, 2014).
Card sorting	A technique to investigate how users tend to group things. The users are given a set of cards containing individual item names and are told to sort them into related piles and label the groups. Card sorting provides insight into the user's mental model and suggests the structure and placement of items on a system.	(Human Factors International, 2014).
Storyboarding	A series of illustrations that represent a user task, such as the steps necessaries to perform a task using a system.	(Usability First, 2015)

APPENDIX F Assessment questionnaire template (Version 0.1)

N	Indicator	N	P	F	Description
1	Our team identifies and describes the purpose of the system.				Identify and describe the purpose of the system, this is, the objective(s) that the user wants to achieve using the system.
2	Our team identifies stakeholders' expectations regarding the performance and behavior of the system.				Identify the stakeholder's requirements regarding the behavior and performance of the system. The requirements cover each aspect of the system related to its use and its interface in a context of use.
3	Our team defines explicit statements of usability requirements based on the context analysis.				Define an explicitly statement for each usability requirements, regarding its effectiveness, efficiency and user satisfaction based on the context of use analysis. The statements should be measurable objectives.
4	Our team identifies and describes the characteristics of the tasks the user performs in the system.				Describe the tasks the users need perform in the system in order to achieve their goals.
5	Our team identifies and describes the characteristics of the users.				Identify relevant characteristics of the users, such as knowledge about the system domain, degree of literacy, physical capabilities, level of experience with the tasks and with the device he will use to interact with the system, motivations in using the system, etc.
6	Our team identifies and describes the organizational and social characteristics regarding the environment in which the system will be use.				Identify relevant social and organizational milieu, management structure, communications and organizational practices in the environment in which the system will be used.
7	Our team identifies and describes the characteristics of the device with which the users will interact to use the system.				Identify relevant characteristics of the device with which the users will directly interact, such as memory and process capacity, ways of input and output data, screen size, etc.
8	Our team describes the physical environment characteristics in which the system will be use.				Identify relevant characteristics of the location, workplace equipment and ambient conditions and its implications for the system design, such as lighting, noise levels, vibration, heat, hazards, dimensions of working and living space.
9	Our team analyzes the use cases in terms of its flow, navigation, main screens and constraints.				Analyze the user's tasks in terms of alternative navigation pathways and flowcharts and identifying the main system screens and constraints.
10	Our team analyzes a range of design options for each aspect of				Analyze a range of design options for each aspect of the system related to its use and its effect on stakeholders, such as definition of

	the system related to its use and its interface.				system controls, location and format of display components, use of colors, terminology, fonts, and wording of messages.
11	Our team applies existing usability knowledge (such as stakeholder requirements, usability guidelines) in the system design.				Applied existing usability knowledge, such as stakeholder requirements, information about the context of use, international standards, usability good practice and style guides to the design of the system and is used to select the appropriate alternatives of design.
12	Our team specifies each aspect of the system related to its use and its interface.				Specified the design of all the user-related components of the system. This specification is a description of how the components and the system will be used, such as the kind of systems controls will be used, location and format of display components, colors, terminology, fonts, wording of messages).
13	Our team prototypes high-fidelity each component of the system interfaces.				Refine design through the development of high-fidelity prototypes of all aspect of the system related to its use and its interface.
14	Our team plans the prototypes and system evaluation.				Prepared and defined all arrangements necessary to evaluate the prototype or the system, such as definition of which evaluation method will be used, who will be the assessor, place, scripts, questionnaires, cameras and etc.
15	Our team evaluates the usability of the prototypes.				Prototypes are evaluated against usability knowledge, style guides, standards, guidelines in order to find usability problems and verify if the required good practices has been followed.
16	Our team evaluates the system in order to check if it meets the requirements.				System is evaluated to ensure that it meets the requirements of the users, the tasks and the environment, as defined in its specification.

APPENDIX G Assessment report template

Usability Process Assessment Report	
Assessment date:	
Assessment meeting moderator:	
Assessment meeting participants:	
Usability process rating	
Usability sub-process rating	
UP1. Specify stakeholder and user requirements:	
UP2. Understand and specify the context of use:	
UP3. Produce design solutions:	
UP4. Evaluate designs against requirements:	
Legend: 0 - 15 points: Not achieved. 16 - 85 points: Partially achieved. 86 - 100: Fully achieved.	
Strength points:	
Points to be improved:	

APPENDIX H Example of UPCASE use

Example of completed UPCASE questionnaire		
Sub-process	**Indicators**	Rating (**0-Not achieved, 1-Partially achieved, 2-Fully achieved**)
1 - Context of use	Our team identifies the purpose of the system.	**1-partially achieved**
	Our team identifies stakeholders' expectations regarding the performance and behavior of the system.	**2-fully achieved**
	Our team defines explicit statements of usability requirements based on the context analysis.	**1-partially achieved**
2 - User requirements	Our team identifies the characteristics of the tasks the user performs in the system.	**2-fully achieved**
	Our team identifies the characteristics of the users.	**0-Not achieved**
	Our team identifies the organizational and social characteristics regarding the environment in which the system will be use.	**2-fully achieved**
	Our team identifies the characteristics of the device with which the users will interact to use the system.	**1-partially achieved**
	Our team identifies the physical environment characteristics in which the system will be use.	**1-partially achieved**
3 - Produce design solutions	Our team analyzes the use cases in terms of its flow, navigation, main screens and constraints.	**2-fully achieved**
	Our team analyzes a range of design options for each aspect of the system related to its use and its interface.	**1-Not achieved**

	Our team applies existing usability knowledge (such as stakeholder requirements, usability guidelines) in the system design.	**2-fully achieved**
	Our team specifies each aspect of the system related to its use and its interface.	**2-partially achieved**
	Our team prototypes high-fidelity each component of the system interfaces.	**2-partially achieved**
4 - Evaluate designs against requirements	Our team plans the prototypes and system evaluation.	**0-Not achieved**
	Our team evaluates the usability of the prototypes.	**2-fully achieved**
	Our team evaluates the system in order to check if it meets the requirements.	**1-partially achieved**

Assessment results		
Sub-process	**Score**	**Rating**
1 - Context of use	4 points (from a total of 6 points = 66,6% of achievement)	P - Partially achieved
2 - User requirements	6 points (from a total of 10 points = 60% of achievement)	P - Partially achieved
3 - Produce design solutions	9 points (from a total of 10 points = 90% of achievement)	F - Fully achieved
4 - Evaluate designs against requirements	3 points (from a total of 6 points = 50% of achievement)	P - Partially achieved
Total score of the usability process	22 points (from a total of 32 points = 68,75% of achievement)	P - Partially achieved

List of points that can be improved with the indicators that can be improved in the sub-processes that were classified with N or P:

- Identify the purpose of the system.
- Define explicit statements of usability requirements based on the context analysis.
- Identify the characteristics of the users.
- Identify the characteristics of the device with which the users will interact to use the system.
- Identify the physical environment characteristics in which the system will be use.
- Analyze a range of design options for each aspect of the system related to its use and its interface.
- Plan the prototypes and system evaluation.
- Evaluate the system in order to check if it meets the requirements.

APPENDIX I General Information Collected in the Case Studies

SE	Domain	Platform	Total Scor	Attribute rate	UP1 Score	UP 2 Sco	UP 3 Sco	UP 4 Sco
1	Government	Web	34.3	P	33.33333	30	60	0
2	Healthcare	Web	68.7	P	66.66666	70	90	20
3	Healthcare	Web	62.5	P	50	60	100	10
4	Other	Other	68.7	P	50	60	100	30
5	Energy	Deskt	59.3	P	50	90	60	10
6	Information	Deskt	56.2	P	66.66666	80	30	30
7	Government	Web	65.6	P	50	70	60	30
8	Information	Mobil	71.8	P	66.66666	60	100	20
9	Information	Web	65.6	P	83.33333	50	80	20
1	Information	Web	59.3	P	83.33333	70	50	20
1	Information	Deskt	15.6	N	16.66666	40	0	0
1	Government	Web	96.8	F	83.33333	100	100	40
1	Information	Mobil	68.7	P	50	80	80	20
1	Information	Web	50	P	50	50	60	20
1	Information	Web	68.7	P	66.66666	70	60	40
1	Information	Web	25	P	33.33333	50	10	0
1	Healthcare	Mobil	81.2	P	83.33333	80	100	20
1	Training/Educ	Other	46.8	P	50	70	40	10
1	Healthcare	Deskt	84.3	P	100	80	100	20
2	Information	Web	53.1	P	50	70	50	20
2	Telecom	Web	31.2	P	50	20	40	10
2	Training/Educ	Web	37.5	P	33.33333	30	40	20
2	Other	Deskt	18.7	P	16.66666	0	20	30
2	Information	Web	9.37	N	33.33333	10	0	0
2	Financial	Web	34.3	P	50	30	50	0
2	Financial	Web	18.7	P	50	30	0	0
2	Healthcare	Web	71.8	P	66.66666	60	90	30
2	Information	Mobil	50	P	50	50	50	20
2	Information	Web	65.6	P	83.33333	60	50	40
3	Retail	Web	21.8	P	16.66666	20	40	0
3	Healthcare	Web	78.1	P	66.66666	90	90	20
3	Information	Web	96.8	F	83.33333	100	100	40
3	Government	Web	56.2	P	50	60	80	0
3	Healthcare	Deskt	59.3	P	33.33333	70	100	0
3	Government	Mobil	81.2	P	83.33333	80	80	40
3	Information	Deskt	100	F	100	100	100	40

APPENDIX J Responses from Assessments Conducted in the Case Studies

SE	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11	q12	q13	q14	q15	q16	Total
1	2	0	0	1	0	1	1	0	1	1	1	1	2	0	0	0	11
2	2	1	1	1	1	2	2	1	2	1	2	2	2	0	0	2	22
3	1	1	1	1	0	2	2	1	2	2	2	2	2	0	1	0	20
4	1	1	1	1	0	2	2	1	2	2	2	2	2	0	1	2	22
5	1	2	0	2	2	2	2	1	0	1	2	1	2	0	0	1	19
6	1	1	2	1	1	2	2	2	1	0	1	0	1	0	2	1	18
7	1	1	1	1	1	2	1	2	1	2	1	2	0	2	1	2	21
8	2	1	1	2	1	1	2	0	2	2	2	2	2	1	1	1	23
9	2	2	1	2	1	1	1	0	2	1	2	2	1	1	1	1	21
10	2	2	1	2	1	1	2	1	1	1	1	1	1	0	1	1	19
11	0	1	0	2	1	0	1	0	0	0	0	0	0	0	0	0	5
12	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	31
13	1	1	1	2	2	1	2	1	2	2	1	2	1	1	1	1	22
14	2	1	0	0	1	1	1	2	1	1	2	2	0	0	2	0	16
15	2	1	1	0	1	2	2	2	1	2	1	1	1	1	2	2	22
16	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	8
17	2	1	2	1	2	2	2	1	2	2	2	2	2	1	1	1	26
18	1	1	1	2	1	1	2	1	1	1	1	1	0	0	0	1	15
19	2	2	2	2	2	2	2	0	2	2	2	2	2	1	1	1	27
20	1	1	1	1	1	1	2	2	1	0	1	2	1	0	1	1	17
21	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	1	10
22	1	1	0	0	1	0	1	1	2	1	0	0	1	1	1	1	12
23	0	0	1	0	0	0	0	0	0	1	0	1	0	0	1	2	6
24	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3
25	1	1	1	1	1	1	0	0	1	2	0	1	1	0	0	0	11

26	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	6
27	1	2	1	2	1	1	1	1	2	2	2	1	2	1	1	2	23
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16
29	2	1	2	1	1	2	1	1	2	1	0	1	1	1	2	2	21
30	1	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	7
31	1	2	1	1	2	2	2	2	2	2	2	1	2	1	1	1	25
32	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	31
33	1	1	1	2	0	1	1	2	2	1	1	2	2	1	0	0	18
34	2	0	0	2	2	2	1	0	2	2	2	2	2	0	0	0	19
35	1	2	2	2	2	1	1	2	1	2	2	2	1	1	2	2	26
36	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	32
Total	48	41	34	46	37	46	49	36	48	46	43	47	42	21	31	36	651
Var	0.3333333333	0.3418209877	0.4413580247	0.4783950617	0.5270061728	0.4783950617	0.4529320988	0.6111111111	0.5	0.5339506173	0.6566358025	0.5455246914	0.6388888889	0.4652777778	0.5640432099	0.6111111111	8.179783951