

DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Metodologia para desenvolvimento de software usado no auxílio à gestão da campanha de certificação do KC-390

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação da disciplina:

DAS 5511: Projeto de Fim de Curso

Renato Bock da Costa

Florianópolis, Fevereiro de 2018

**Metodologia para desenvolvimento de software usado
no auxílio à gestão da campanha de certificação do
KC-390**

Renato Bock da Costa

Esta monografia foi julgada no contexto da disciplina
DAS 5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Prof. Rômulo Silva de Oliveira

Banca Examinadora:

Rodrigo Daun Monici
Orientador na Empresa

Prof. Rômulo Silva de Oliveira
Orientador no Curso

Prof. Ricardo José Rabelo
Responsável pela disciplina

Fernando Silvano Gonçalves, Avaliador

Paulo Henrique de Oliveira Curado, Debatedor

João Gabriel Honorato de Assis, Debatedor

Agradecimentos

Gostaria de utilizar esse espaço para agradecer algumas das pessoas que passaram pela minha vida nesses anos de graduação. Foi certamente um período marcante e decisivo para mim, acredito ter crescido muito tanto profissionalmente quanto pessoalmente. Assim, não posso deixar de citar as pessoas que tornaram essa fase algo tão importante.

Inicialmente, agradeço meu orientador Rômulo Silva de Oliveira por sempre ter se mostrado disposto e disponível a tirar todas minhas questões.

Ao meu supervisor na Embraer, Rodrigo Daun Monici, pela confiança depositada e por ter feito com que eu me sentisse parte da equipe. Aproveito aqui também para agradecer à todos os excelentes engenheiros com os quais tive a oportunidade de trabalhar e por terem me ensinado, sempre com muita paciência, diversos aspectos da indústria aeronáutica.

Agradeço à minha mãe Geni, meu pai Wesler e meu irmão Lucas, por serem o principal pilar que sustenta minhas decisões. O suporte dado em todas as fases me deu forças pra continuar sempre de pescoço erguido e sem medo de enfrentar os obstáculos.

Por fim, agradeço todos meus amigos e amigas pelos momentos inesquecíveis que partilhamos diariamente no decorrer desse período. Suas amizades foram essenciais para amenizar as muitas horas de estudos e trabalhos. Desejo a todos o maior sucesso em suas vidas profissionais.

Resumo

Embraer S.A. é uma fabricante brasileira de aviões, atuando nas áreas de aviação comercial, executiva, defesa e agrícola.

O trabalho proposto neste documento entra no escopo de campanhas de certificação de novas aeronaves desenvolvidas pela Embraer. Especificamente no contexto da campanha do KC-390, uma nova aeronave multimissão militar em desenvolvimento pela Embraer Defesa & Segurança.

Para facilitar a visualização de informações-chave desta campanha de ensaios, utiliza-se uma plataforma web chamada SAG (Serviço de Apoio Gerencial). Nele encontram-se diversos módulos (ou ferramentas) utilizados para acompanhar o avanço da campanha de ensaios. Essas ferramentas possuem indicadores, tabelas, planilhas, etc., possibilitando uma melhor visibilidade de diversos fatores da campanha, facilitando as tomadas de decisões das diversas partes envolvidas.

O estágio realizado foca no desenvolvimento de melhorias nas ferramentas já existentes do SAG e na criação de novas. Como essas tarefas aparecem num ritmo imprevisível e possuem uma complexidade de implementação diferente, há uma dificuldade para aplicar-se uma metodologia no acompanhamento do desenvolvimento dos softwares do SAG.

No decorrer desse desenvolvimento, foi feita uma caracterização de como o acompanhamento é feito hoje na Embraer e como são gerenciadas as tarefas relacionadas ao SAG. Além disso, foi realizado um estudo das principais metodologias existentes, para que ao final, junto com a caracterização feita do ambiente atual, descrever a metodologia utilizada que se encaixou no desenvolvimento de plataformas de auxílio gerencial para a Embraer.

Palavras-chave: Engenharia de Software, Metodologias de desenvolvimento de Software, Desenvolvimento Ágil, Desenvolvimento Web

Lista de ilustrações

Figura 1 – Logo Embraer S.A.	19
Figura 2 – Vista frontal KC-390	22
Figura 3 – Vista lateral KC-390	22
Figura 4 – Primeiro protótipo do KC-390 (Fonte: http://kc-390.com)	22
Figura 5 – Ícone SAG	23
Figura 6 – Visão parcial do SAG	24
Figura 7 – Modelo Cascata	28
Figura 8 – Modelo Espiral	29
Figura 9 – Ciclo do <i>Scrum</i>	30
Figura 10 – Quadro <i>Kanban</i> básico (Fonte: https://criar.me/2017/02/a-ferramenta-kanban/)	31
Figura 11 – Diagrama demonstrando situação inicial do fluxo de tarefas dos estagiários	37
Figura 12 – Fluxo de gestão inicial das tarefas de desenvolvimento do SAG	42
Figura 13 – Fluxo de gestão regular das tarefas de desenvolvimento do SAG	44
Figura 14 – Quadrante para definição de prioridade	45
Figura 15 – Tecnologias e linguagens utilizadas no desenvolvimento do SAG	45
Figura 16 – Ferramenta de Controle de OEs - Visualização inicial	50
Figura 17 – Ferramenta de Controle de OEs - Classificação das OEs por tecnologia	50
Figura 18 – Ferramenta de Controle de OEs - Visualização com filtro	51
Figura 19 – Ferramenta de Controle de OEs - Visualização de planejamento	51

Lista de tabelas

Tabela 1 – Diferenças entre <i>Scrum</i> e <i>Kanban</i>	32
--	----

Lista de abreviaturas e siglas

AJAX - *Asynchronous Javascript and XML*

AMP - Apache, MySQL, PHP

ANAC - Agência Nacional de Aviação Civil

FAB - Força Aérea Brasileira

GPX - Gavião Peixoto (cidade)

HH - Homem-Hora

IFI - Instituto de Fomento e Coordenação Industrial

KPI - *Key Performance Indicators*

OE - Ordem de Engenharia

SAG - Sistema de Apoio Gerencial

WIP - *Work in Progress*

Sumário

1	INTRODUÇÃO	15
1.1	Escopo e objetivos do projeto	16
1.2	Estrutura do documento	16
2	SOBRE A EMPRESA	19
2.1	História	19
2.2	Unidades	20
2.3	Campanhas de Certificação	20
2.4	KC-390	21
2.5	Processos existentes	23
2.5.1	SAP	23
2.5.2	OEs	23
2.5.3	SAG	23
2.5.3.1	Desenvolvimento	24
3	METODOLOGIAS DE DESENVOLVIMENTO	27
3.1	Modelo Cascata	27
3.2	Modelo Incremental	28
3.3	Modelo Espiral	28
3.4	Modelo Ágil	29
3.4.1	<i>Scrum</i>	30
3.4.2	<i>Kanban</i>	31
3.4.3	Considerações	32
4	AMBIENTE ATUAL DE DESENVOLVIMENTO	35
4.1	Descrição	35
4.2	Valores e características	35
5	METODOLOGIA UTILIZADA	39
5.1	Análise das metodologias estudadas	39
5.1.1	Cascata	39
5.1.2	Incremental	40
5.1.3	Espiral	40
5.1.4	Ágil	40
5.2	Solução	41
5.2.1	Metodologia para acompanhamento e gestão	41

5.2.1.1	Fase de Aprendizado	41
5.2.1.2	Fase Regular	43
5.2.2	Metodologia para desenvolvimento	43
6	APLICANDO A METODOLOGIA	47
7	CONCLUSÕES E CONSIDERAÇÕES	53
	REFERÊNCIAS	55

1 Introdução

A construção de sistemas de software é complexa, pois deve lidar com requisitos intransigentes, restrições de integridade e a necessidade de um vasto conhecimento sobre a aplicação para que as interações esperadas entre o software e o ambiente possam ser adequadamente descritos. Quando os requisitos não são totalmente compreendidos, registrados e comunicados para a equipe de desenvolvimento, muito provavelmente, haverá discrepância entre o que o sistema construído faz e o que ele deveria fazer.

Já faz alguns anos que o desenvolvimento de software deixou de ser sinônimo apenas de código. Hoje em dia, sabe-se que é necessária a utilização de uma metodologia de trabalho [1].

Entende-se por metodologia, como a maneira de se utilizar um conjunto coerente e coordenado de métodos para atingir um objetivo, de modo que se evite, tanto quanto possível, a subjetividade na execução do trabalho. Fornecendo um roteiro, um processo dinâmico e interativo para desenvolvimento estruturado de projetos, sistemas ou software, visando à qualidade e produtividade dos projetos.

É objetivo de uma metodologia definir de forma clara “quem” faz “o que”, “quando”, “como”, para todos os que estejam envolvidos diretamente ou não com o desenvolvimento de software. Deve definir também qual o papel dos técnicos, dos usuários, e o da administração da empresa no processo de desenvolvimento. Com isso, evita-se a situação a qual o conhecimento sobre o sistema é de poucos, os chamados “donos do sistema”. Com isso, o uso de uma metodologia possibilita:

- Ao **gerente/supervisor**: controlar o projeto de desenvolvimento de software mantendo o rumo do projeto sobre controle para que não haja desvios de planejamentos de custos e prazos, que, se negligenciados ou mal conduzidos, podem por em risco o sucesso do projeto.
- Ao **desenvolvedor**: obter a base para produzir de maneira eficiente, software de qualidade que satisfaça os requisitos estabelecidos.

A escolha de uma metodologia a ser utilizada no desenvolvimento, deve ser realizada com base na natureza do projeto e do produto a ser desenvolvido e dos métodos e ferramentas a serem utilizadas.

Finalmente, vale ressaltar os seguintes pontos:

- Um projeto pode dar certo por utilizar uma metodologia adequada

- Um projeto pode dar certo mesmo utilizando uma metodologia não adequada
- Um projeto pode falhar por utilizar uma metodologia não adequada
- Um projeto pode falhar mesmo utilizando uma metodologia adequada

1.1 Escopo e objetivos do projeto

O estágio foca no desenvolvimento do software utilizado pela Embraer para acompanhamento das campanhas de ensaios de suas aeronaves em fase de certificação. As tarefas no decorrer do estágio apareciam numa frequência alta, e a complexidade de desenvolvimento delas foi bastante variado. Assim, para que as tarefas fossem concluídas no menor tempo possível, era necessário que houvesse disciplina e lógica na hora de definir a prioridade de cada uma e a forma como elas eram desenvolvidas. Uma metodologia mostra-se necessária para auxiliar essa gestão de prioridade entre as tarefas e facilitar o desenvolvimento das mesmas.

No início tentou-se seguir uma metodologia chamada *Scrum* (maiores detalhes no capítulo 3), mas com o passar do tempo deixamos de seguir as diretrizes da metodologia. Essa falta de disciplina ocasionou no cenário onde os próprios desenvolvedores (composto por dois estagiários) escolhiam as tarefas que julgassem mais prioritárias, mesmo tendo uma visão mais restrita do sistema. Outra característica observada com a falta de uma metodologia foi que cada uma das tarefas requisitadas eram consideradas de prioridade alta pelo requerente, assim houveram diversas mudanças na ordem das tarefas a serem executadas ou até mesmo interrompidas para o início de outra. Por fim, os supervisores já não conseguiam mais manter um controle próximo das tarefas em desenvolvimento e quais deveriam ser as próximas na fila de implementação.

Utilizando de um estudo das metodologias mais utilizadas no desenvolvimento de software, juntamente com o grande volume de tarefas desenvolvidas, esse trabalho expõe a metodologia que se enquadrou bem no cenário descrito, para que o desenvolvimento futuro do SAG e em novos programas da Embraer consigam ser melhor acompanhados e as tarefas executadas de forma mais organizada.

1.2 Estrutura do documento

O capítulo 2 apresenta a Embraer e como a empresa está dividida. Também é introduzida uma visão geral de como são feitas as campanhas de certificação para aeronaves civis e militares, usando como exemplo o KC-390. Ainda serão apresentados de forma global os processos que envolvem o controle dos protótipos das aeronaves a serem certificadas, especificando a ferramenta onde o estágio focou.

Em seguida, no capítulo 3, explica-se a noção de metodologia, pontuando e caracterizando as mais utilizadas para o desenvolvimento de software.

Após esse estudo, uma caracterização de como era o processo de desenvolvimento ao início do estágio. Este capítulo (4) mostra os pontos do cenário atual que atrapalharam o bom fluxo do desenvolvimento das tarefas.

Uma vez caracterizado o ambiente e conhecendo melhor as metodologias existentes, juntamos as informações, analisando se as metodologias estudadas se adequam ao cenário de desenvolvimento atual, descrever então uma que se encaixe no cenário existente (capítulo 5). Em seguida, no capítulo 6, mostramos como essa metodologia foi aplicada para uma das tarefas executadas no decorrer do estágio.

Por fim, o capítulo 7 conclui o documento, expondo quantas melhorias e novas ferramentas foram feitas/criadas no decorrer do estágio para a concepção e otimização da metodologia utilizada. São expostos também os principais desafios encontrados e como a metodologia ajuda no desenvolvimento do SAG de futuros programas da Embraer.

2 Sobre a Empresa

Embraer S.A. (logo empresarial na figura 1, ao lado) é um conglomerado transnacional brasileiro fabricante de aviões comerciais, executivos, agrícolas e militares. A empresa tem sede na cidade de São José dos Campos, interior do estado de São Paulo, e possui diversas unidades no Brasil e no exterior.



Figura 1 – Logo Embraer S.A.

2.1 História

A Embraer – Empresa Brasileira de Aeronáutica S.A. foi criada em 19 de agosto de 1969, como uma companhia de capital misto e controle estatal [2].

Além de iniciar a produção do Bandeirante, a Embraer foi contratada pelo Governo Brasileiro para fabricar o jato de treinamento avançado e ataque ao solo EMB 326 Xavante, sob licença da empresa italiana Aermacchi. Outros desenvolvimentos que marcaram o início das atividades da Embraer foram o planador de alto desempenho EMB 400 Urupema e a aeronave agrícola EMB 200 Ipanema.

Ao final da década de 1970, o desenvolvimento de novos produtos, como o EMB 312 Tucano e o EMB 120 Brasília, seguidos pelo programa AMX, em cooperação com as empresas Aeritalia (hoje Alenia) e Aermacchi, permitiram que a Empresa alcançasse a um novo patamar tecnológico e industrial.

O Brasília tornou-se importante marco na história da Embraer. Desenvolvido como resposta às novas demandas do transporte aéreo regional, seu projeto utilizou as mais avançadas tecnologias disponíveis à época, estabelecendo novos padrões para a aviação regional, tornando-se o mais veloz, o mais econômico e o mais leve avião de sua categoria.

Durante crise financeira vivida na primeira metade da década de 1990, a Embraer reduziu o seu quadro de empregados, retardou o desenvolvimento do EMB 145 e cancelou o projeto do CBA 123 Vector. Finalmente, após longo processo e enfrentando muitas dificuldades, a Empresa foi privatizada em 7 de dezembro de 1994.

A partir daquele momento decisivo e contando com o comprometimento e apoio dos novos acionistas controladores – a Cia. Bozano, Simonsen e os fundos de pensão Previ e Sistel – a Embraer embarcou em profunda transformação cultural e empresarial, que culminou com sua recuperação e retomada do crescimento, impulsionada pelo projeto EMB 145, mais à frente rebatizado como ERJ 145.

Nos anos seguintes, com o lançamento dos novos produtos para o mercado de Defesa & Segurança, e a entrada no mercado de Aviação Executiva, a Embraer expandiu ainda mais sua atuação no mercado aeronáutico, ampliando receitas e diversificando mercados.

A entrada em operação da nova família de jatos comerciais EMBRAER 170/190 a partir de 2004, a confirmação da presença definitiva da Embraer no mercado de Aviação Executiva com o lançamento de novos produtos, assim como a expansão de suas operações no mercado de Serviços Aeronáuticos, estabeleceram bases sólidas para o desenvolvimento futuro da Empresa.

Apesar do cenário desafiador do final dos anos 2000, a Embraer alcançou grandes conquistas a partir da década seguinte, em todos os negócios. Ainda em 2010, a partir de um reposicionamento estratégico, a razão social Embraer – Empresa Brasileira de Aeronáutica foi alterada para Embraer S.A., o que permitiu ampliar e diversificar as áreas de negócios. Com isso, ainda em dezembro daquele ano foi criada a unidade de negócio Embraer Defesa e Segurança que, nos anos seguintes, ampliaria seu escopo de atuação por meio de aquisições de participações societárias e da criação de empresas coligadas. Recentemente, em 2016, a Embraer também criou uma unidade de negócios com foco em serviços e suporte ao cliente.

2.2 Unidades

A Embraer está presente em diversas cidades dentro do território nacional. Conforme dito anteriormente, a empresa tem sede em São José dos Campos - SP, e possui unidades em Eugênio de Melo - SP, Gavião Peixoto - SP, Botucatu - SP, Belo Horizonte - MG, etc. No cenário internacional, a empresa possui unidades e subsidiárias nos EUA, Portugal, China, França e mais.

O estágio e o desenvolvimento deste trabalho foram feitos na unidade de Gavião Peixoto (GPX), no interior do estado de São Paulo.

2.3 Campanhas de Certificação

A certificação (ou homologação) de aeronaves é uma atividade necessária à segurança do transporte aéreo. Consiste em avaliar e atestar que um determinado produto (aeronave ou seu componente) possui as características mínimas que assegurem seu uso seguro para o tipo de operação pretendida (transporte de passageiros, pulverização agrícola, combate a incêndio, transporte exclusivo de carga, operação somente em dia claro, operação noturna, entre outras).

No Brasil, o órgão certificador para aeronaves de uso civil é a ANAC (Agência Nacional de Aviação Civil). O IFI (Instituto de Fomento e Coordenação Industrial) é o

responsável pela certificação de aeronaves de uso militar [3].

Dentro do processo de certificação, existe a certificação de projeto. Ela é composta por avaliações de engenharia, nas quais se busca constatar que o projeto está de acordo com as normas de segurança vigentes e demonstrar que a aeronave não apresenta nenhum aspecto ou característica insegura quando operada dentro das limitações estabelecidas para o uso pretendido. Tais avaliações de engenharia são feitas em cima de dados e informações obtidos através de testes (no próprio protótipo, em laboratório, em túnel de vento, etc.), de simulações em computador, de argumentações analíticas, entre outras técnicas.

Para realização dos ensaios (testes) em solo e em voo das aeronaves, a Embraer utiliza duas pistas de pousos e decolagens:

- A do aeroporto de São José dos Campos, que possui 2.676 metros de extensão (a área pertence ao Departamento de Ciência e Tecnologia Aeroespacial, mas é utilizada tanto pela Embraer quanto para transporte aéreo comercial, sendo a infraestrutura compartilhada e mantida em parceria com a Infraero).
- A da unidade da Embraer em Gavião Peixoto (aeródromo de Gavião Peixoto), que possui 4.967 metros de extensão e é considerada a mais longa da América Latina

Atualmente a Embraer possui dois programas realizando suas campanhas de certificação, a família de jatos comerciais E-Jets E2 (E175-E2, E190-E2 e E195-E2) e o KC-390, aeronave multi-missão militar.

2.4 KC-390

O Embraer KC-390 (vistas frontal e lateral nas figuras 2 e 3, respectivamente) é uma aeronave para transporte tático/logístico e reabastecimento em voo desenvolvido e fabricado pela Embraer Defesa e Segurança em parcerias com a Argentina, Portugal e República Tcheca. A aeronave estabelece um novo padrão para o transporte militar médio, visando atender os requisitos operacionais da Força Aérea Brasileira (FAB), em substituição ao C-130 Hércules.

O KC-390 foi projetado para cumprir a seguinte gama de missões [4]:

- Transporte e lançamento de cargas e tropas;
- Reabastecimento em voo - caças, transporte ou ISR;
- Evacuação Aeromédica (UTI móvel, remoção de feridos);
- Transporte de cargas paletizadas;



Figura 2 – Vista frontal KC-390



Figura 3 – Vista lateral KC-390

Fonte: <http://kc-390.com>

- Transporte de veículos leves e médios;
- Ajuda humanitária;
- Lançamento a baixa altura (LAPES - *Low Altitude Parachute Extracting System*);
- Lançamento de cargas e paraquedistas em todas as altitudes;
- Operação em pistas não pavimentadas e curtas;
- Combate a incêndios florestais.

A aeronave iniciou sua campanha de ensaios em voo no dia 26 de outubro de 2015.

O cronograma de desenvolvimento do KC-390 sofreu atraso de dois anos, por conta de restrições orçamentárias governamentais. A certificação e as primeiras entregas para a FAB estão previstas para 2018. Atualmente o KC-390 possui dois protótipos, ambos estão sendo usados nos ensaios para garantir a certificação. O primeiro protótipo, já com as pinturas da FAB é mostrado na figura 4.

Figura 4 – Primeiro protótipo do KC-390 (Fonte: <http://kc-390.com>)

O estágio realizado entra no escopo da campanha de certificação do KC-390.

2.5 Processos existentes

Essa seção do capítulo mostra alguns dos processos envolvidos na campanha de certificação do KC-390 e como é feito o acompanhamento de seu avanço.

2.5.1 SAP

O SAP é o sistema ERP (*Enterprise Resource Planning*) utilizado pela Embraer para gerenciar seus recursos, incluindo materiais, pessoas, fábricas e equipamentos. O SAP é utilizado em vários dos programas da Embraer para fazer o controle dos recursos e dos processos existentes. Apesar de sua importância para a empresa, gerar os dados e navegar no ambiente SAP é pouco intuitivo e combinar resultados é uma tarefa complexa.

2.5.2 OEs

Uma OE (Ordem de Engenharia) é o documento utilizado para descrever alguma modificação a ser feita nas aeronaves. Elas são geradas pelos diversos times de engenharia da Embraer e contempla todos os detalhes de uma modificação, como sua descrição, a(s) aeronave(s) onde ela deve ser executada, quais peças serão afetadas, etc.

Uma vez a OE criada (através do ambiente SAP), ela passa ao time de projetistas da engenharia para criação dos desenhos que serão utilizados pelo time de manufatura, o qual executa a modificação.

2.5.3 SAG

Para facilitar a visualização de informações-chave da campanha de ensaios, criou-se o SAG (Sistema de Apoio Gerencial) cujo ícone é mostrado na figura 5. Ele é uma plataforma WEB onde encontram-se diversos módulos/ferramentas utilizadas para auxiliar o acompanhamento da campanha de ensaios do KC-390.



Figura 5 – Ícone SAG

A maioria dos dados que compõem o SAG são provenientes do SAP, porém mostrados de forma mais intuitiva. O SAG possibilita uma melhor visibilidade de diversos fatores da campanha, facilitando as tomadas de decisões das diversas partes envolvidas. Os diversos módulos do SAG disponibilizam essas informações na forma de tabelas, indicadores, planilhas, etc. A figura 6 mostra parte da janela do SAG, com algumas de suas ferramentas.



Figura 6 – Visão parcial do SAG

2.5.3.1 Desenvolvimento

O desenvolvimento das ferramentas do SAG foram o foco do estágio realizado. Ele possui dezenas de módulos, cada um com informações diferentes umas das outras. Assim, existem diversas melhorias a serem feitas para que essas informações fiquem mais fáceis de serem interpretadas e representem melhor a situação da campanha de ensaios.

O Jira (software comercial desenvolvido pela *Atlassian*) é a plataforma utilizada para o monitoramento e acompanhamento de diferentes projetos da Embraer e suas atividades. Onde são definidos, para cada atividade, o responsável, horas necessárias para conclusão, log com trabalho parcial, etc.

Dentro do Jira há um projeto somente com as tarefas referentes às melhorias e novos módulos do SAG do KC-390. Essas tarefas são executadas pelo time de desenvolvedores composto por dois estagiários.

Por vezes, há uma certa dificuldade para definir a prioridade das atividades, manter um acompanhamento delas, retomar aquelas que foram interrompidas por outras mais urgentes. Como o SAG é utilizado por diversas pessoas, sugestões de melhorias aparecem de diferentes formas e áreas. Sem uma boa gestão dessas tarefas e diretrizes para sua execução, tarefas importantes podem demorar para serem realizadas enquanto outras menos urgentes tomam a frente da fila.

No início do estágio tentamos usar a metodologia *Scrum* (com as ferramentas disponíveis pelo próprio Jira) para o desenvolvimento do SAG, porém com o passar do

tempo deixamos de seguir suas diretrizes até o abandono quase que completo.

Este trabalho consiste em encontrar uma metodologia que se encaixe nesse cenário e que poderá ajudar os futuros responsáveis pelo desenvolvimento do SAG, essa metodologia pode ser uma já existente, ou apenas uma adaptação.

3 Metodologias de desenvolvimento

Conforme mencionado no capítulo 1, podemos definir metodologia como um conjunto estruturado de práticas que pode ser repetível durante o processo de produção de um projeto, nesse caso, um software.

No decorrer dos anos, foram criadas e aprimoradas diversas metodologias, cada uma com seus pontos fortes e fracos. Existem diversas formas de se abordar o desenvolvimento de um software: algumas tomam caminhos mais estruturados, de maneira onde o todo é construído de forma conjunta, outras utilizam técnicas incrementais, desenvolvendo o software passo a passo. As metodologias geralmente combinam os seguintes estágios de desenvolvimento de software:

- Análise do problema
- Pesquisa de mercado
- Definição de requisitos
- Plano e design do software
- Desenvolvimento propriamente dito
- Teste do software
- Implementação

Conjuntamente, esses estágios são normalmente referidos como sendo o ciclo de vida de um projeto de desenvolvimento de software. Cada metodologia ordena e foca em diferentes estágios, assim, cada uma possui um cenário de projeto ideal para ser utilizada. A seguir são apresentados alguns dos modelos existentes, com suas principais características.

3.1 Modelo Cascata

Cascata é um modelo de desenvolvimento de software caracterizado por um processo contínuo "sempre em frente" (diagrama da figura 7). É um modelo ingênuo para desenvolvimentos reais, pois não podemos ter um processo inflexível e não-iterativo.

A definição de requisitos nem sempre é completamente definida no início do projeto. Muitas vezes o cliente pode perceber algum requisito quando a fase de implementação já começou, inviabilizando uma metodologia não-iterativa. Metodologias que se encaixam nesse modelo costumam focar bastante na fase de definição de requisitos e design [5].

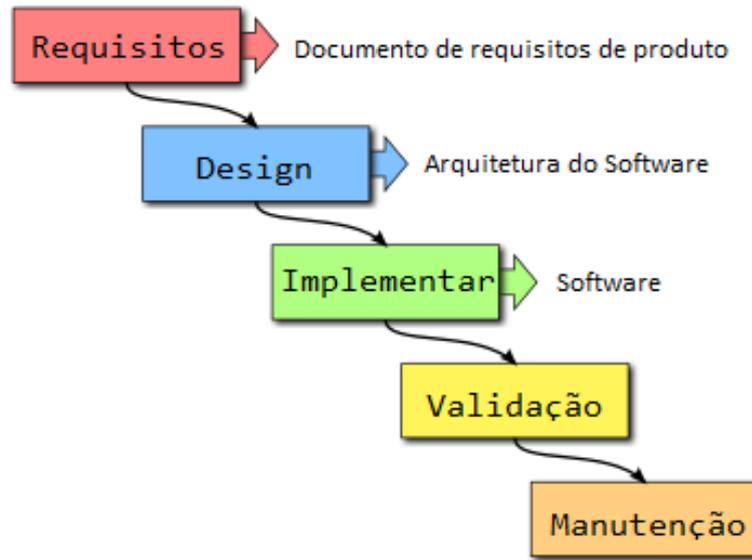


Figura 7 – Modelo Cascata

3.2 Modelo Incremental

Nesse modelo de desenvolvimento, o produto é criado de forma incremental até que o produto final esteja pronto. Para o desenvolvimento de cada parte (ou incremento) é normalmente utilizado o modelo de cascata. Ele pode ser considerado uma mistura do modelo cascata com o modelo de prototipagem, visto que cada incremento é adicionado ao "protótipo" (i.e. uma versão incompleta do produto sendo desenvolvido) até que o mesmo se torna o produto final [6].

3.3 Modelo Espiral

O modelo espiral normalmente é aplicado para grandes projetos, onde logo no início o projeto é dividido em etapas que serão implementadas a cada ciclo [7]. Conforme mostra a figura 8.

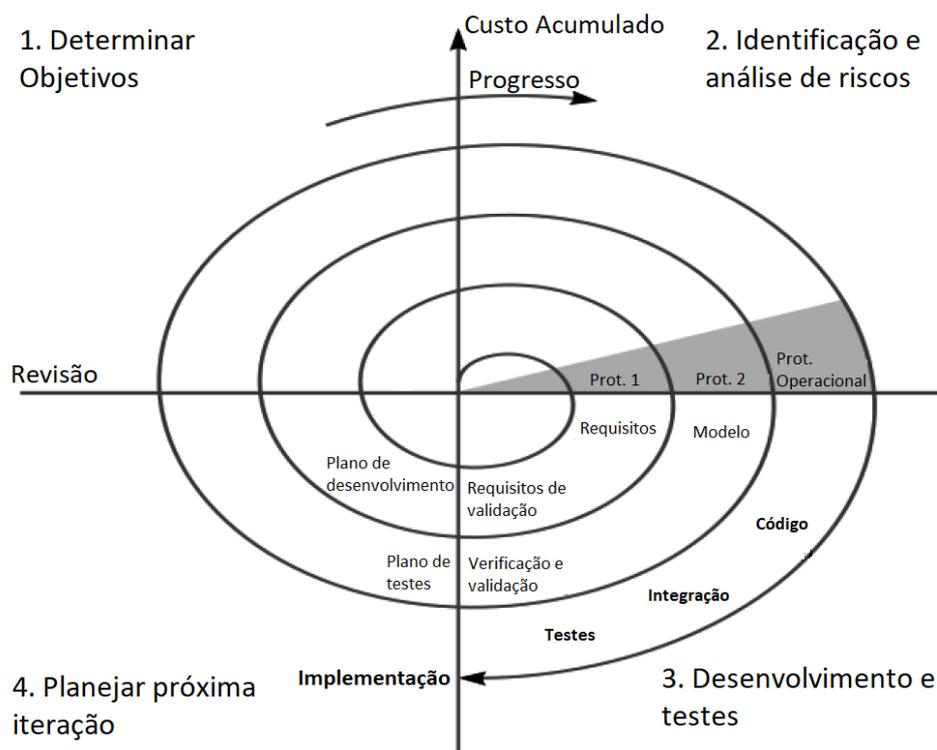


Figura 8 – Modelo Espiral

3.4 Modelo Ágil

Nos modelos ágeis [8], tentamos minimizar o risco envolvido no desenvolvimento de um software usando-se de curtos períodos, chamados de iteração, os quais gastam tipicamente de uma semana a até quatro. Cada iteração é como um projeto de software em miniatura de seu próprio, e inclui todos os estágios de desenvolvimento citados anteriormente.

Os quatro principais valores do modelo de desenvolvimento ágil são:

- Os **indivíduos** e suas **interações** acima de procedimentos e ferramentas
- O **funcionamento do software** acima de documentação abrangente
- A **colaboração com o cliente** acima da negociação e contrato
- A **capacidade de resposta a mudanças** acima de um plano pré-estabelecido

Assim, conclui-se que métodos ágeis enfatizam a comunicação em tempo real, a documentos escritos. Enfatizam também o próprio software como uma medida de progresso. Combinando essas duas características, vemos que o modelo ágil produz pouca documentação com relação a outros métodos, o que pode ser considerado um ponto negativo em alguns casos.

A seguir são mostradas duas das metodologias ágeis mais conhecidas e utilizadas.

3.4.1 Scrum

Scrum se baseia em ciclos com período de tempo definido, chamados *Sprints*, onde trabalha-se para alcançar objetivos bem definidos.

Estes objetivos são apresentados no *Product Backlog*, uma lista com os requisitos e funcionalidades esperadas no produto final. A figura 9 ilustra o ciclo do *Scrum*.

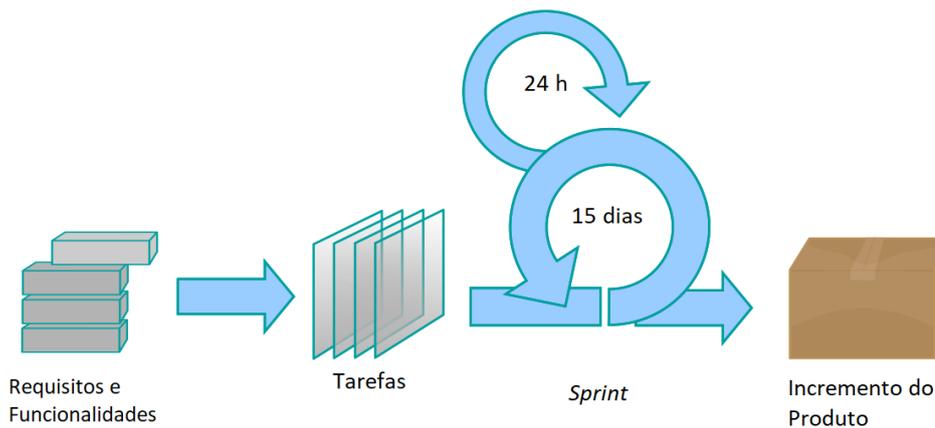


Figura 9 – Ciclo do *Scrum*

Scrum é um esqueleto de processos que contém grupos de práticas e papéis pré-definidos. Os três papéis principais que o compõem são:

- **Equipe:** responsável por entregar soluções. Geralmente é formada por um grupo pequeno (entre 5 a 9 pessoas).
- **Product Owner:** responsável pela visão de negócios do projeto. Cabe a este papel definir e priorizar o *Product Backlog*. Geralmente é o papel desempenhado pelo cliente ou por um gerente de produtos da empresa que desenvolve o software.
- **Scrum Master:** é uma mistura de gerente, facilitador e mediador. Seu papel é remover obstáculos da equipe e assegurar que as práticas de *Scrum* estejam sendo executadas com eficiência.

Inicialmente no *Scrum*, todas as funcionalidades ou mudanças no produto são definidas pelo *Product Owner* no *Product Backlog*. Esta lista é priorizada para refletir a necessidade dos clientes ou demandas do mercado.

No início do ciclo do *sprint*, uma reunião de planejamento da *sprint* é realizado. Nele é selecionado o trabalho a ser feito (a "parte" do produto final), as tarefas detalhadas com o tempo necessário estimado para sua execução.

O *Scrum* não define uma duração obrigatória para os *sprints*, podendo durar alguns dias ou algumas semanas, dependendo do projeto em questão. O importante é, uma vez regulada a duração do *sprint*, não fazer mais alterações até que o mesmo seja finalizado.

Ao final da *sprint*, é analisado o trabalho que foi concluído e não concluído. Como objetivo ao final de uma *sprint* é sempre um trabalho completo, o mesmo é apresentado ao cliente.

Finalmente, os membros da equipe refletem sobre a *sprint* finalizada e analisam a possibilidade de fazer uma melhoria contínua nos processos.

3.4.2 Kanban

No *Kanban*, os itens de trabalho são representados visualmente em um quadro do *Kanban*, permitindo que os membros da equipe vejam o estado de cada parte do trabalho a qualquer momento. Ele utiliza dos mesmos princípios do Toyotismo, com seu processo de fabricação "*just in time*". O *Kanban* combina a quantidade de trabalho em andamento (WIP - *Work In Progress*) com a capacidade da equipe, isso proporciona às equipes opções de planejamento mais flexíveis, saída mais rápida, foco mais claro no ciclo de desenvolvimento.

O trabalho de todas as equipes *Kanban* gira em torno de um quadro do *Kanban*, uma ferramenta usada para visualizar o trabalho e otimizar o fluxo do trabalho entre a equipe. Um quadro básico do *Kanban* tem um fluxo de trabalho de três etapas: "*To Do*", "*In Progress/Doing*" e "*Done*" (conforme mostrado na figura 10). No entanto, dependendo do tamanho, da estrutura e dos objetivos da equipe, o fluxo de trabalho pode ser mapeado para atender ao processo exclusivo de qualquer equipe específica.

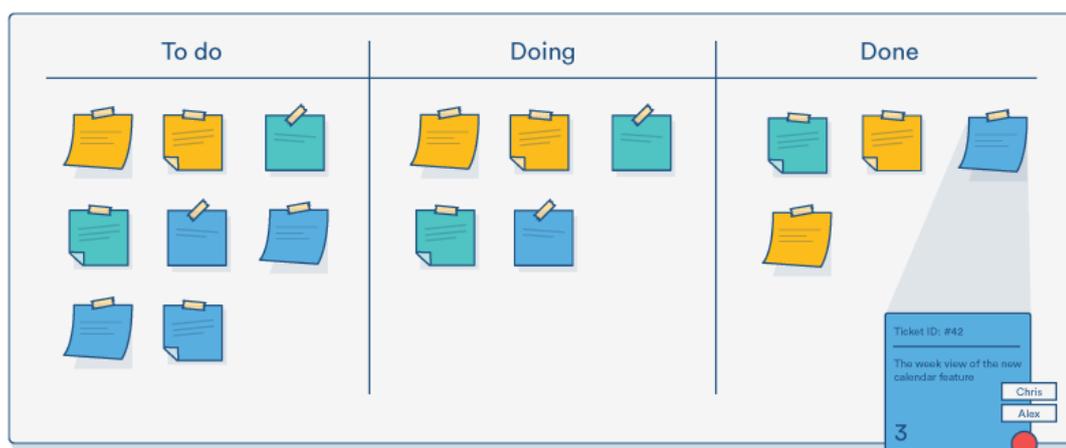


Figura 10 – Quadro *Kanban* básico (Fonte: <https://criar.me/2017/02/a-ferramenta-kanban/>)

O *Kanban* é uma das metodologias de desenvolvimento de software mais populares adotadas por equipes ágeis atualmente. Ele oferece várias vantagens adicionais para o

planejamento e a transferência de tarefas para equipes de todos os tamanhos. Entre as principais vantagens, podemos citar:

- **Flexibilidade de planejamento:** Uma equipe *Kanban* concentra-se apenas no trabalho ativamente em andamento. Depois que a equipe conclui um item de trabalho, retira-se o próximo item de trabalho do topo da lista de pendências.
- **Tempos de ciclos reduzidos:** O tempo de ciclo é uma métrica-chave para as equipes *Kanban*. O tempo de ciclo é a quantidade de tempo que uma unidade de trabalho leva para passar pelo fluxo de trabalho da equipe desde o momento que o trabalho é iniciado até o seu envio. Ao otimizar o tempo de ciclo, a equipe pode prever com confiança a entrega do trabalho futuro.
- **Serviço constante:** Quanto mais rápido uma equipe puder oferecer inovação para o mercado, mais competitivo o produto estará no mercado. E as equipes *Kanban* focam exatamente nisto, otimizar o fluxo de trabalho para clientes.

3.4.3 Considerações

Uma vez que as duas metodologias ágeis mais usadas hoje em dia foram mostradas, suas principais diferenças são resumidas na tabela 1.

	<i>Scrum</i>	<i>Kanban</i>
Ritmo	<i>Sprints</i> regulares com extensão fixa	Fluxo contínuo
Metodologia da versão	No final de cada <i>sprint</i> , se aprovado pelo proprietário do produto	Entrega contínua, ou a critério da equipe
Papéis	Equipe de desenvolvimento, <i>Product Owner</i> , <i>Scrum Master</i>	Sem funções existentes
Principais métricas	Velocidade	Tempo de ciclo
Filosofia de mudança	As equipes devem se esforçar para não fazer alterações na previsão de <i>sprint</i> durante o mesmo	Mudanças podem ocorrer a qualquer momento

Tabela 1 – Diferenças entre *Scrum* e *Kanban*

Vale ressaltar que a escolha de uma metodologia não é algo definitivo, o importante é fazer avaliações no processo em prol da melhoria contínua. Não é necessário escolher uma metodologia existente e seguir a risca cada uma das suas diretrizes, pode-se combiná-las para conseguirmos no final uma que se encaixe no projeto atual. O desenvolvimento de software possui uma realimentação rápida, isso ajuda na hora de definirmos as características do sistema, e então definir a metodologia que se encaixa.

No próximo capítulo é feita essa caracterização do ambiente atual na Embraer para o desenvolvimento do SAG.

4 Ambiente atual de desenvolvimento

Para utilizar-se uma metodologia da forma correta, é necessário conhecer o ambiente no qual o projeto será desenvolvido, assim como sua complexidade. Assim, foi feita uma análise do ambiente de desenvolvimento do SAG quando foi iniciado o estágio. Essa análise pontua as principais características, e o que os engenheiros da Embraer valorizam no produto final, com isso é possível adaptar ou criar uma metodologia que se adéque ao sistema e auxilie o bom acompanhamento e desenvolvimento das ferramentas do SAG.

4.1 Descrição

O desenvolvimento dos módulos do SAG é feito por um time de dois estagiários, o qual é anualmente renovado. Os clientes são os próprios engenheiros da Embraer, usuários do SAG, que sentem a necessidade de melhoria em alguma ferramenta existente, ou possuem ideias de novas funcionalidades a serem implementadas, que auxiliem na visualização dos status da campanha de ensaios.

Inicialmente, o Jira era utilizado como ferramenta para o acompanhamento das tarefas. Seu ambiente possui as ferramentas necessárias para seguir a metodologia *Scrum*. Semanalmente eram feitas reuniões de definição de novos *sprints*, onde as tarefas para a semana eram estipuladas e o resultado da semana que passou eram verificados. Nessa fase, as tarefas passadas costumavam ser de prioridade baixa e simples, com o objetivo principal de familiarizar os desenvolvedores às ferramentas e conhecer melhor o SAG.

Com o passar do tempo, o *Scrum Master* começou a prorrogar as reuniões até o ponto onde os próprios desenvolvedores escolhiam as tarefas a serem desenvolvidas. Nesse cenário, muitas vezes as tarefas em desenvolvimento não eram àquelas com maior prioridade, então os clientes requisitavam que o trabalho atual fosse interrompido para que a tarefa de alta prioridade iniciasse. Esse fluxo de tarefas dessa fase do estágio está mostrado na figura 11.

4.2 Valores e características

A seguir estão algumas das métricas comumente utilizadas em projetos de software e uma breve descrição do cenário atual segundo a métrica.

- **Tamanho do projeto:** O SAG possui dezenas de módulos, criados e aprimorados no decorrer da campanha do KC-390. Porém as ferramentas são, em sua maioria, trabalhadas independentemente umas das outras. Portanto considera-se que as tarefas

são de pequeno tamanho, uma vez que mexem com uma pequena parte do projeto global.

- **Tecnologias disponíveis:** Conforme dito anteriormente, o SAG é uma plataforma web, o servidor onde encontram-se as páginas roda um sistema AMP (Apache + MySQL + PHP). Portanto a parte de servidor e comunicação com o banco de dados (*back-end*) é desenvolvida em PHP. Já para a parte do cliente (*browser*), chamada de *front-end*, existe uma liberdade na escolha da tecnologia (Vanilla Javascript, jQuery, AngularJS, etc.).
- **Definição de requisitos:** As tarefas normalmente não possuem requisitos bem definidos, o cliente informa quais informações ele sente falta e como imagina o resultado final, se através de tabelas ou indicadores por exemplo. Isso deixa os desenvolvedores com uma certa liberdade para implementação, como por exemplo a forma de salvar as informações no bando de dados, como recuperá-las e como disponibilizá-las na ferramenta do SAG.
- **Revisão de código:** Para as páginas do SAG, é dado valor ao resultado final e não à maneira como foi estruturado o código. Portanto o código só é revisado pelo próprio desenvolvedor.
- **Tempo para entrega:** Um dos valores mais importantes percebidos no decorrer do estágio foi a necessidade de rápida implementação. Como muitas das tarefas relacionadas ao SAG estão relacionadas a problemas nas ferramentas, sua rápida correção torna-se necessária.
- **Documentação:** Conforme o ponto anterior e como não há revisão de código, os desenvolvedores não produzem documentação dos módulos criados/modificados. Isso dificulta possíveis modificações futuras em ferramentas já existentes.
- **Escalabilidade:** Considerando novamente a necessidade de rápida implementação, e pelo fato dos módulos normalmente não serem modificados depois de prontos, não é feito um estudo e projeto para tornar simples a escalabilidade das ferramentas.

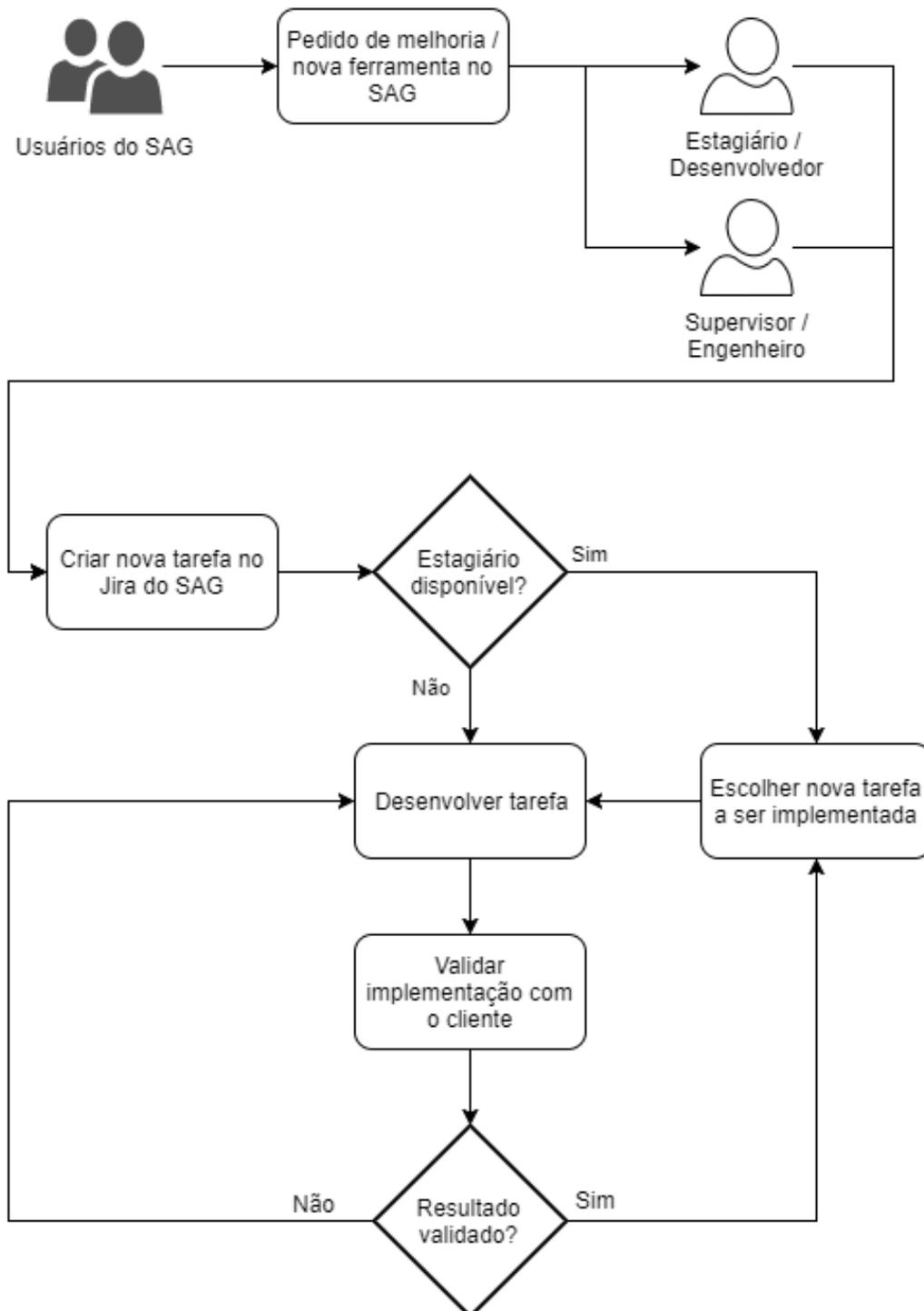


Figura 11 – Diagrama demonstrando situação inicial do fluxo de tarefas dos estagiários

5 Metodologia utilizada

Como foi dito no capítulo anterior, vemos que com o passar do tempo, deixou-se de usar qualquer metodologia para o acompanhamento do desenvolvimento do SAG.

Com o estudo de metodologias feito pelo estagiário e descrito no capítulo 3, juntamente com a caracterização dos valores atuais feita no capítulo 4, podemos analisar qual metodologia se adéqua melhor ao ambiente descrito e, se necessário, adaptá-la para as particularidades do projeto.

Este capítulo descreve a análise feita nas metodologias estudadas, para entender e pontuar quais não se encaixam no ambiente de desenvolvimento atual e o porquê. Em seguida é descrita a metodologia definida pelo estagiário para acompanhar e desenvolver as tarefas relacionadas ao SAG.

5.1 Análise das metodologias estudadas

Metodologias diferem umas das outras em diversos aspectos, uma das várias formas de compará-las é com base no número de regras que elas proveem [8]. Chamamos de metodologias prescritivas aquelas que definem procedimentos detalhados, já as que não possuem essa característica são chamadas de metodologias adaptativas, pois elas proveem conselhos. Uma metodologia 100% prescritiva significa que existem regras e passos para todos os possíveis cenários, sem nenhuma liberdade e autonomia para a equipe. 100% adaptativa se equivale à algo sem regras nem restrições, onde cada um da equipe faz o que achar melhor para o projeto. Ambos extremos são claramente ineficientes no desenvolvimento de produtos. A seguir temos um resumo dos pontos fortes e fracos [9] das metodologias citadas anteriormente e qual dos valores do desenvolvimento do SAG dificulta a utilização da mesma.

5.1.1 Cascata

O maior problema do modelo cascata é sua inflexibilidade para mudanças. Os requisitos precisam estar bem definidos antes do desenvolvimento de código propriamente dito. Essas características se encaixam em projetos grandes e de alta complexidade, com requisitos e objetivos claramente definidos e onde não há uma pressão para implementação e término o mais rápido possível. Esse não é o caso para tarefas do SAG, pois normalmente os requisitos são modificados após a implementação de uma primeira versão da ferramenta, a qual deve estar disponível o quanto antes.

5.1.2 Incremental

O modelo incremental quebra o produto final em diversas partes e aplica a metodologia cascata para cada uma delas. Embora isso possibilite uma maior flexibilidade à mudanças em requisitos, essa metodologia é melhor aplicável em grandes projetos. Apesar do SAG como um todo poder ser considerado um projeto grande, as tarefas relacionadas a ele são normalmente de pequena duração, portanto não adequado ao modelo incremental de desenvolvimento.

5.1.3 Espiral

Bastante semelhante ao incremental, o modelo espiral utiliza da mesma estratégia de dividir um grande projeto em partes de complexidade menor. A principal diferença é que o modelo incremental faz essa divisão completa logo no início do planejamento, já o espiral define as partes a cada nova iteração de desenvolvimento. Os mesmos motivos anteriores explicam o porquê de não se encaixar bem no ambiente do SAG.

5.1.4 Ágil

Analisando os princípios de metodologias ágeis apresentados anteriormente, vemos que eles estão alinhados com as características do processo atual. Ainda assim, uma metodologia considerada ágil e adaptativa como o *Scrum* deixou de ser utilizada por não terem sido respeitadas suas regras. Dois fatores principais fizeram com que o *Scrum* deixasse de ser utilizado:

- O papel do *Scrum Master*, que garante que as práticas do *Scrum* estejam sendo executadas deixou de ser exercido.
- Como eram dois estagiários responsáveis pelo desenvolvimento, não foi possível ajustar as tarefas de forma eficiente nos *sprints* para ambos. Ao final do *sprint* era normal acontecer de um desenvolvedor não possuir mais tarefas enquanto o outro estava no meio do processo de implementação, e vice-versa.

A metodologia do *Kanban* é ainda mais adaptativa que o *Scrum*. Seus princípios ajudam na solução dos dois problemas apresentados:

- Não requer explicitamente o papel de gerente que decide o momento em que uma tarefa é iniciada, são os próprios desenvolvedores que o fazem.
- Não há uma noção de período de desenvolvimento como os *sprints*, portanto ao terminar uma tarefa o desenvolvedor pode, logo em seguida, puxar a próxima na lista de prioridade. Porém sem ultrapassar o WIP máximo.

5.2 Solução

Com o decorrer do estágio, das ferramentas desenvolvidas e do estudo das metodologias existentes e suas características, o estagiário conseguiu adaptar e seguir uma metodologia que se encaixasse nesse cenário.

A metodologia definida possui duas partes. A primeira refere-se à gestão das tarefas e como priorizá-las, já a segunda parte refere-se ao desenvolvimento efetivo das tarefas. Ambas partes de metodologia foram definidas e otimizadas com o passar do tempo pelo estagiário, conforme o mesmo adquiriu uma maior experiência e autonomia para a realização das tarefas do SAG.

5.2.1 Metodologia para acompanhamento e gestão

Essa parte da metodologia refere-se à maneira como são gerenciadas as tarefas, definidas suas prioridades e validados seus resultados. Ela foi dividida em duas fases, uma de aprendizado, onde o estagiário ainda não conhece muito bem o sistema e precisa de orientação constante do supervisor e engenheiros da Embraer para realizar as tarefas. A outra fase refere-se ao período onde o estagiário já possui autonomia suficiente para priorizar sozinho a maioria das tarefas, e conhece o sistema a ponto de não precisar mais tirar dúvidas constantes com seus superiores.

5.2.1.1 Fase de Aprendizado

Conforme mostra o diagrama da figura 12, essa fase inicial da metodologia possui as seguintes características:

- Para criação de novas tarefas, os clientes pedem diretamente ao supervisor dos estagiários. O mesmo então cria a tarefa no Jira, ordenando e priorizando as tarefas segundo as características dessa fase inicial.
- O supervisor distribui tarefas de complexidade baixa ao estagiário, as mesmas vão servir para que o desenvolvedor se familiarize com os códigos e tabelas existentes, assim como as tecnologias utilizadas para o desenvolvimento das diversas ferramentas.
- Semanalmente são marcadas reuniões para entender as dificuldades e acompanhar o avanço das tarefas. Essa fase vale também para que o estagiário estude a parte das tecnologias com as quais não está acostumado a desenvolver e que são essenciais às ferramentas do SAG.
- Uma vez finalizada a tarefa, é feita uma reunião para avaliar os resultados e expor as principais dificuldades, para que elas sejam trabalhadas nas próximas tarefas.

- Manter esse processo, aumentando progressivamente a complexidade das tarefas, até que o estagiário consiga priorizar a maior parte das tarefas sozinho e considere ter um bom conhecimento do sistema e do tempo necessário para desenvolvimento das tarefas.
- Vale ressaltar que nessa fase, é papel fundamental do supervisor não deixar que as tarefas em desenvolvimento sejam interrompidas. Uma vez iniciada, a tarefa deve proceder até sua implementação.

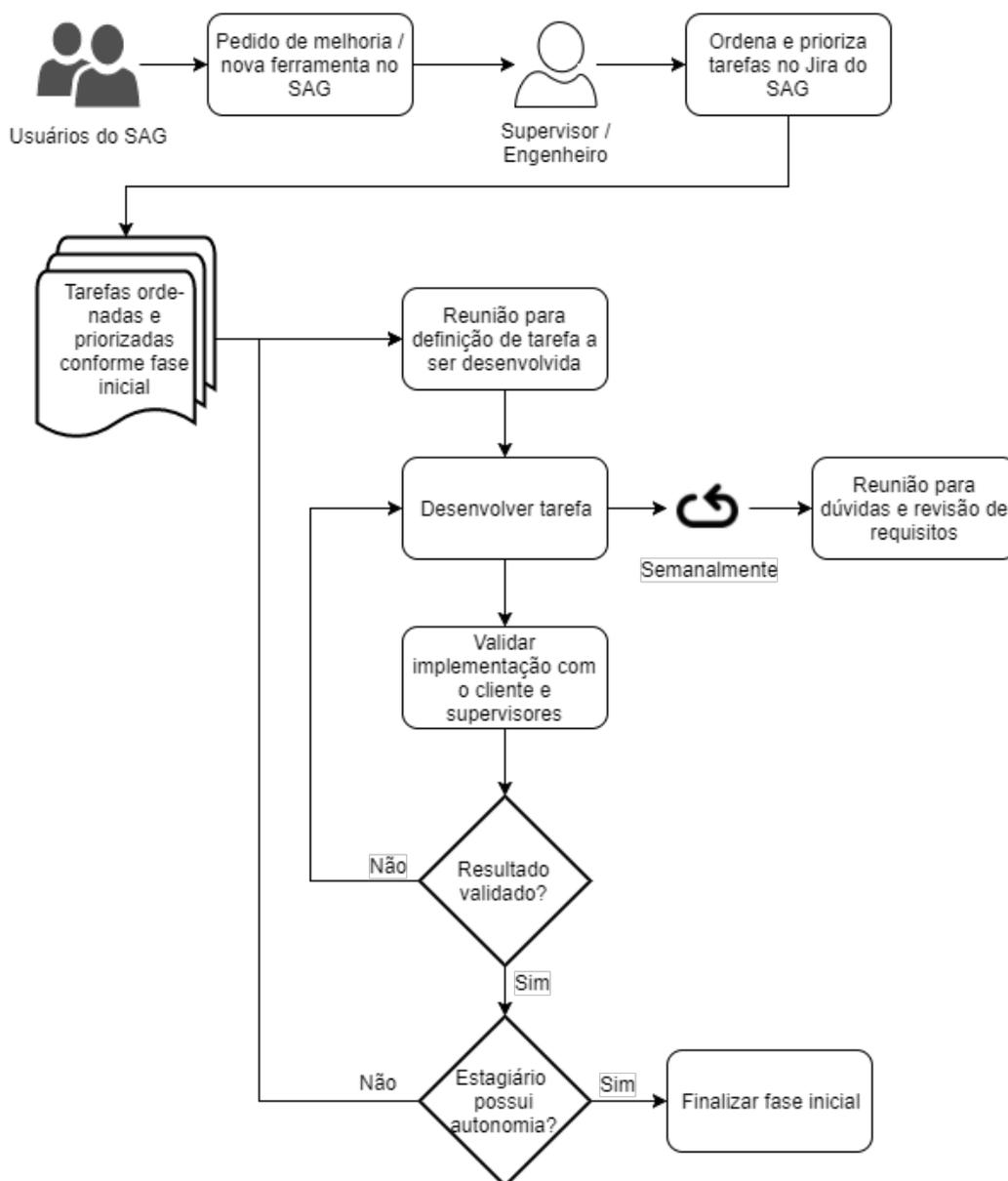


Figura 12 – Fluxo de gestão inicial das tarefas de desenvolvimento do SAG

5.2.1.2 Fase Regular

Conforme dito anteriormente, nessa fase o estagiário já possui autonomia para definir a prioridade da tarefa e sabe quanto tempo aproximadamente será necessário para realização da mesma. Dessa forma, a metodologia passa a seguir os seguintes passos (mostrados também na figura 13).

- Novas tarefas podem ser criadas pelos estagiários e supervisor.
- Como já conhece um pouco melhor as ferramentas e a complexidade que envolve um tarefa, o desenvolvedor agora consegue definir sua prioridade. A figura 14 mostra os dois fatores que definem a prioridade das tarefas: o impacto previsto para a modificação (ou utilidade) e a quantidade de horas necessárias para implementá-la.
- Ao criar a tarefa no Jira, contatar o cliente para informar quando a tarefa deverá ser iniciada e a prioridade que ela possui.
- Assim que iniciar a tarefa, discutir detalhes de requisitos sobre a modificação ou nova ferramenta com o requerente. Ao final, informar quando a tarefa deverá ser finalizada.
- Sempre que necessário, mostrar resultados parciais ao cliente para validação, até que todos os requisitos sejam cumpridos.
- Entrega e finalização da tarefa.

5.2.2 Metodologia para desenvolvimento

Essa parte da metodologia utilizada refere-se ao desenvolvimento propriamente dito, uma vez definida a próxima tarefa a ser implementada. O desenvolvimento das tarefas segue os seguintes passos:

- Caso seja uma modificação, é necessário o estudo e a compreensão da página e dos códigos (quais tecnologias e lógica por trás da implementação existentes). Caso seja uma nova ferramenta, então deve-se escolher tecnologias para desenvolvimento. A figura 15 mostra as tecnologias e linguagens utilizadas no decorrer do estágio.
- Análise e definição das tabelas do banco de dados que serão lidas, modificadas ou, eventualmente, criadas.
- Criação de um ambiente de desenvolvimento e testes (exceto seja uma modificação pequena e simples), visto que não há controle de versão para as ferramentas do SAG.

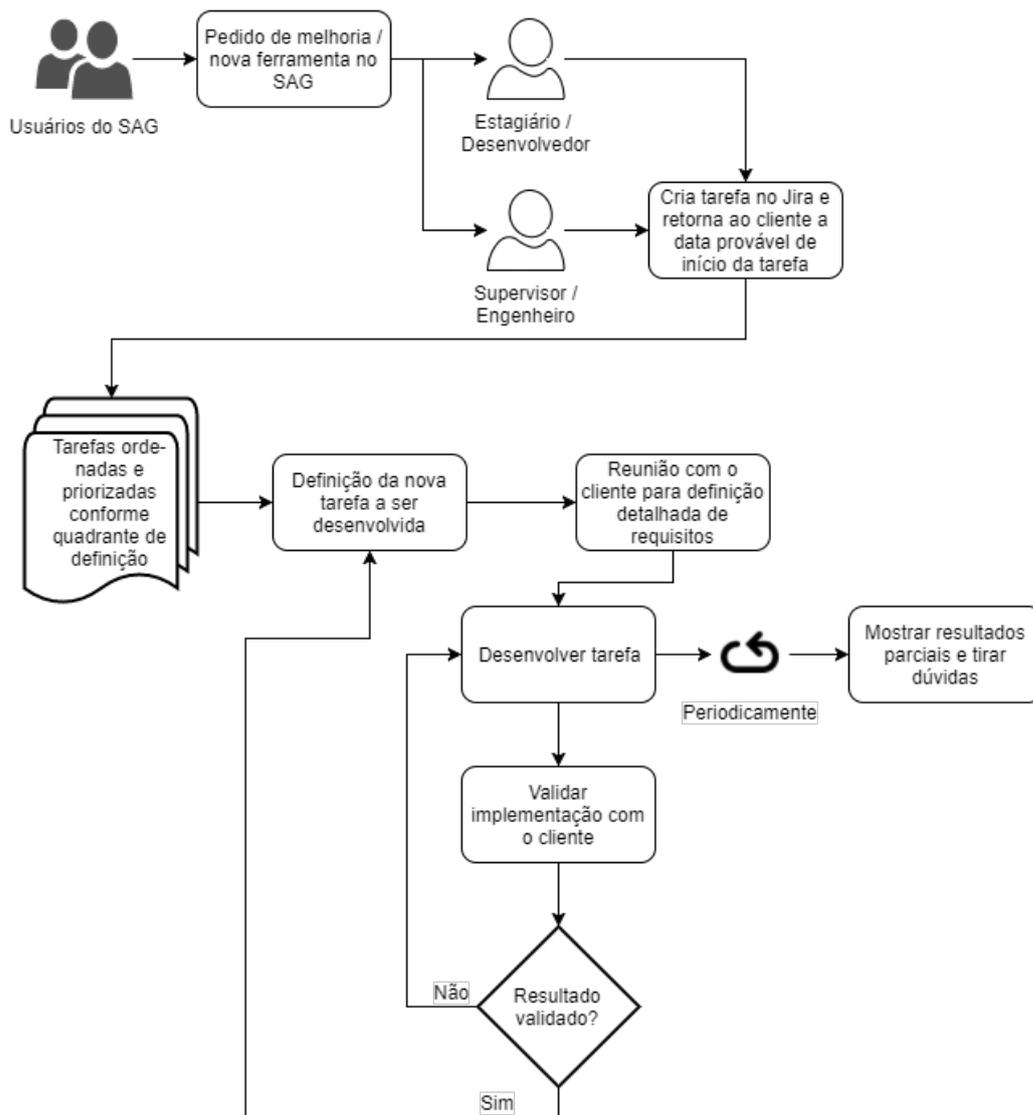


Figura 13 – Fluxo de gestão regular das tarefas de desenvolvimento do SAG

- Planejamento e definição dos passos necessários para implementação. Definir classes, arquivos, funções/métodos a serem desenvolvidos. Esses passos servem de grande ajuda para acompanhar o ponto de desenvolvimento que a tarefa se encontra, para recomeço rápido no caso de uma interrupção.
- Teste e contínua validação com o cliente, modificando algum requisito caso necessário.
- Implementação no SAG e remoção do ambiente de desenvolvimento e possíveis tabelas de teste do banco de dados.

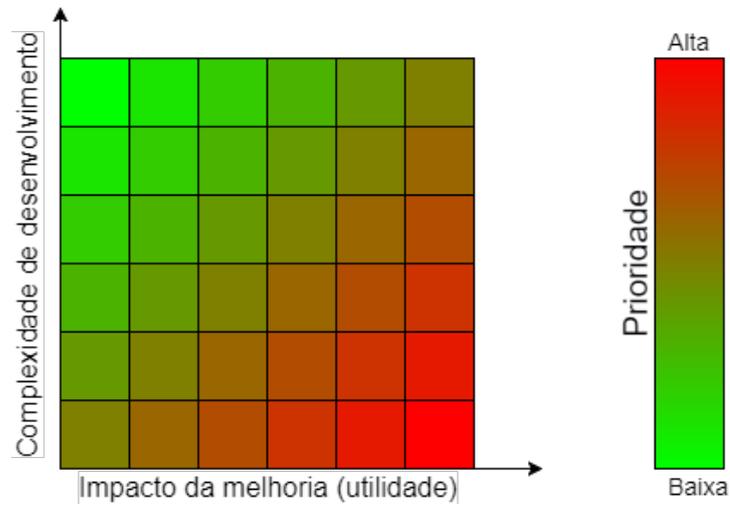


Figura 14 – Quadrante para definição de prioridade

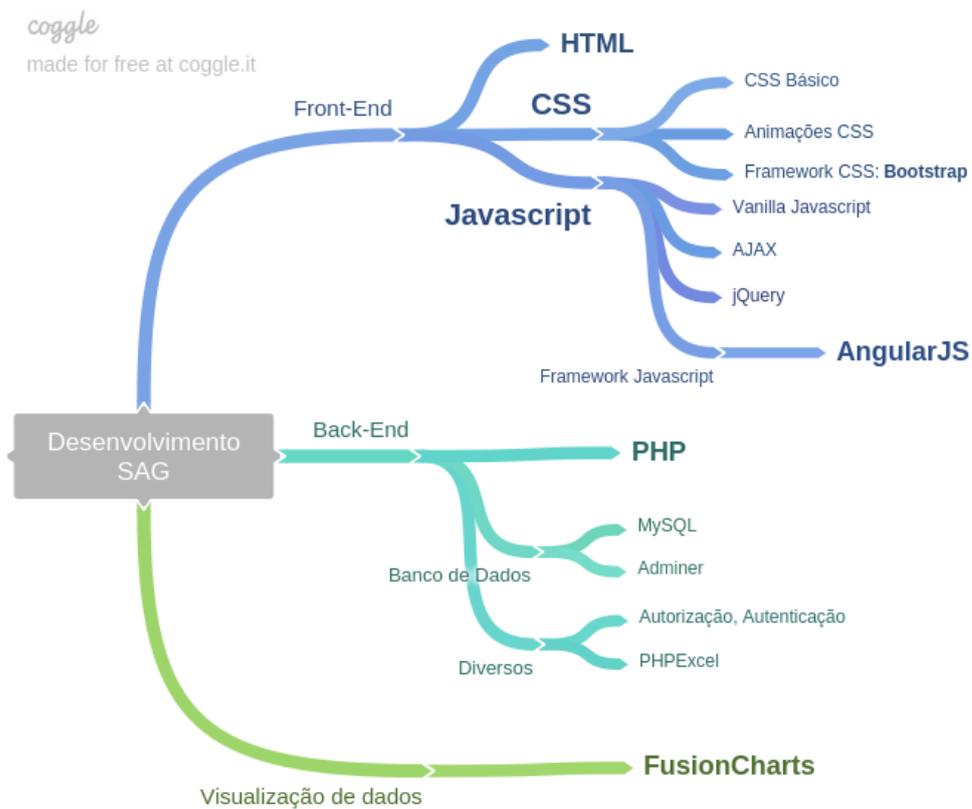


Figura 15 – Tecnologias e linguagens utilizadas no desenvolvimento do SAG

6 Aplicando a metodologia

Neste capítulo será apresentado um exemplo da aplicação da metodologia descrita no capítulo anterior. O exemplo é baseado em uma das tarefas realizadas pelo estagiário, entre os meses de agosto e setembro. Nessa etapa, o estagiário já havia definido a metodologia descrita no capítulo anterior e possuía um conhecimento do tempo estimado necessário para desenvolvimento das tarefas.

Conforme dito no capítulo 2, as OEs (Ordem de Engenharia) são documentos que descrevem modificações a serem feitas nas aeronaves. O SAG possui uma ferramenta para controle das OEs que precisam ser finalizadas. Nela são listados diversos campos, como o número do documento, seu status atual, em qual aeronave ela deverá ser aplicada, o sistema envolvido na modificação (tecnologia relacionada - e.g. Hidromecânicos, Elétrica, Interiores, etc), o responsável pela finalização e liberação da OE, etc.

A ferramenta em seu estado inicial era composta por uma única tabela onde cada linha era composta por diferentes dados de uma mesma OE. Com o passar do tempo, a equipe de projeto (equipe responsável pela liberação de OEs) passou a usar uma planilha Excel para fazer o acompanhamento das OEs, pois haviam diversas limitações na ferramenta do SAG que impediam o controle adequado desses documentos. Houve então uma reunião com a equipe para que a tarefa de recriação da ferramenta fosse criada no Jira e eles pudessem expor os problemas que sentiam na ferramenta atual. Entre elas estavam:

- A página demora aproximadamente 20 segundos para carregar todas as informações.
- Não é possível colocar duas tecnologias para uma mesma OE, sendo que existem vários casos onde a OE afeta dois ou mais sistemas da aeronave.
- Diversas colunas não estão sendo mais utilizadas, ocupando espaço desnecessário e dificultando a visualização das informações relevantes.
- Não há colunas para definir prioridade e HH (Homem-Hora) necessária à uma OE.
- Para cada filtro da lista, a página é recarregada. Quando o filtro é limpaado, demoram mais 20 segundos para recarregar as informações.

Ao fim da reunião, foi feita uma análise da situação atual das tarefas já existentes no Jira e dada uma previsão de quando a tarefa da ferramenta de controle de OEs seria iniciada.

Assim que esta tarefa tornou-se a de mais alta prioridade, foi feita uma nova reunião junto com a equipe de projeto para discutirmos o layout da nova ferramenta e

funcionalidades adicionais que a equipe gostaria que tivesse. Alguns dos requisitos definidos foram:

- Página mais dinâmica, sem recarregar a mesma para cada novo filtro de informação da lista.
- Dividir a página em várias listas (tabelas), uma para cada tecnologia. Uma lista inicial "Sem Tecnologia" para aquelas OEs que ainda não tenham sido classificadas.
- Novas colunas editáveis nas tabelas: prioridade da OE e HH necessária para conclusão.
- Ordenar OEs pelo seu nível de prioridade.
- Visualização de planejamento, como um calendário, onde o supervisor pode definir quantos funcionários irão trabalhar em cada OE e em quais datas.
- Na visão de planejamento, coluna com HH restantes atualizada a cada novo recurso (funcionário) adicionado no calendário.

Uma vez com os requisitos definidos, o estagiário inciou os passos definidos pela metodologia descrita no capítulo anterior:

1. Primeiramente, foi feito um estudo no código-fonte da ferramenta existente para entender a lógica por trás do funcionamento e encontrar o motivo da lentidão da página.
2. Após o estudo, foi visto que a página usava uma solução pura em PHP + HTML, sem nada que tornasse a página dinâmica. Como um dos requisitos era dar uma dinamicidade maior à página, optou-se por desenvolver a ferramenta do zero, em uma aplicação usando *AngularJS*, com requisições AJAX para buscar as informações no banco de dados com scripts em PHP. Dessa forma não seria mais necessário recarregar a página a cada filtro da tabela que fosse utilizado. Assim, o código existente não foi utilizado.
3. Com o estudo da ferramenta também foi encontrado em quais tabelas do banco de dados as informações de OEs estavam e definir também a estrutura de uma nova tabela que guardasse os detalhes com as informações requisitadas que não estavam presentes na versão antiga da ferramenta.
4. Em seguida, com as tabelas criadas e a escolha de tecnologia definida, foi criado um ambiente de desenvolvimento para a nova ferramenta.
5. Com tudo pronto, os passos necessários para a realização da tarefa foram definidos. O objetivo desta etapa é criar algo pequeno de cada vez, de forma gradual, para

então adicionar novas funcionalidades até a implementação final. Os passos definidos foram:

- a) Criação do esqueleto da página (HTML, CSS, AngularJS), sem informação do banco de dados.
 - b) Arquivo que busca as informações do banco de dados e estrutura a informação no formato JSON.
 - c) Divisão das OEs entre as tabelas de cada tecnologia, caso não tenha sido definida ainda, aparece na lista "Sem Tecnologia".
 - d) Janelas modais para edição de: prioridade, HH, observação, executante, status e tecnologia afetada.
 - e) Ao definir a tecnologia, tirar a OE da lista "Sem Tecnologia" e inserir nas tabelas das tecnologias correspondentes às escolhidas.
 - f) Ordenar OEs pela prioridade.
 - g) Criação do filtro de resultados.
 - h) Possibilidade de exportar conteúdo em uma planilha Excel.
 - i) Botão para trocar o tipo de visualização entre lista e calendário de planejamento.
 - j) Possibilidade de adicionar, remover recursos (funcionário) em uma célula de data.
6. Além da definição dos passos, ao final de cada dia o estagiário escrevia um log diário de trabalho para que a retomada no dia seguinte fosse rápida.
 7. As etapas acima são implementadas sequencialmente, quando houveram dúvidas, eram feitas reuniões com alguns dos membros da equipe de projeto para que pudessem ser discutidos detalhes e garantir que a nova página estava de acordo com as expectativas.
 8. Com todas as etapas completas, o estagiário reuniu-se com a equipe de projetos e seu supervisor para mostrar a nova ferramenta e validar seu funcionamento. Em seguida, a tarefa foi finalizada no Jira.

As figuras seguintes mostram o resultado da ferramenta desenvolvida. A figura 16 mostra a página inicial, onde é possível ver os campos de filtro do resultado e a primeira das listas, onde encontram-se as OEs cuja tecnologia ainda não foi definida.

Na figura 17 temos a visualização de algumas das listas que classificam as OEs segundo a tecnologia relativa à ela.

A figura seguinte (18) mostra o resultado utilizando o filtro do campo "Executante". Na versão antiga a página era recarregada para aplicar os filtros selecionados, na nova

Controle de OEs

Status OE: Sub-bloco: OE:

Status Atividade: Executante:

Esconder Tabelas

Sem Tecnologia (16 OEs)

Prioridade	Autor	Subbloco	Num OE	Descrição	Tipo OE	Efetividade	Status OE	Tecnologia	Executante	Status Atividade	Observação	HH	Data Liberação
3	FILIFE DEBONZI GORLA		350003713-00	LRU Reidentification (Landing Gear)	EXPERIMENTAL	[1/1]	PREL	Sem Tecnologia	-		-	0	
3	ADRIANO DE DEUS FRACAO	02-LAYUP-CHS	350003709-00	LH Cargo ramp actuator (FT) instl	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[2/2]	INAP	Sem Tecnologia	-		-	0	
3	ADRIANO DE DEUS FRACAO		350003720-00	CARGO DOOR PANEL INSCRIPTION	EXPERIMENTAL	[2/2]	PREL	Sem Tecnologia	-		-	0	
3	ADRIANO DE DEUS FRACAO		350003724-00	LAYUP CHS: CONFIG CDS/PIQD	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[2/2]	PREL	Sem Tecnologia	-		-	0	
3	ADRIANO DE DEUS FRACAO		350003725-00	LAYUP CHS: CONFIG BSA	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[2/2]	PREL	Sem Tecnologia	-		-	0	
3	ADRIANO DE DEUS FRACAO		350003726-00	LAYUP CHS: CONFIG ENSAIOS	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[2/2]	PREL	Sem Tecnologia	-		-	0	
3	ADRIANO DE DEUS FRACAO		350003727-00	LAYUP CHS: CONFIG LVADILAPES	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[2/2]	PREL	Sem Tecnologia	-		-	0	

Figura 16 – Ferramenta de Controle de OEs - Visualização inicial

Controle de OEs

Status OE: Sub-bloco: OE:

Status Atividade: Executante:

Esconder Tabelas

Sem Tecnologia (16 OEs)

AMS (7 OEs)

Diagramas (11 OEs)

Elétrica (17 OEs)

Estruturas (31 OEs)

Hidráulica (7 OEs)

Interiores (13 OEs)

Mecanismos (6 OEs)

Figura 17 – Ferramenta de Controle de OEs - Classificação das OEs por tecnologia

versão os dados são filtrados dinamicamente conforme são preenchidos e/ou selecionados os campos.

Por fim, a figura 19 mostra a visualização de planejamento feito para uma das OEs, onde é possível ver a distribuição dos recursos no decorrer da semana.

Como resultado, a nova ferramenta de controle de OEs carrega todas as informações em aproximadamente 1 segundo, ao invés dos quase 20 segundos anteriores devido à uma busca ineficaz ao banco de dados. Ela também possibilitou que a equipe deixasse de

Controle de OEs

Status OE:

Status Atividade:

Sub-bloco:

Executante:

OE:

Esconder Tabelas
Mostrar Tabelas
X

Sem Tecnologia (0 OEs)

AMS (4 OEs)

Prioridade	Autor	Subbloco	Num OE	Descrição	Tipo OE	Efetividade	Status OE	Tecnologia	Executante	Status Atividade	Observação	HH	Data Liberação
1	FLAVIANO DO COUTO	SETUP B4	390002990-00	Precooler Seal Ducts Rework	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[1/2]	PREL	AMS	MAURO HIROSHI IZUMI	Não Iniciado	-	40	
3	EVERTON LUIZ SALLES		390001952-01	PACK FTI retrofit	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[1/2]	PREL	AMS	MAURO HIROSHI IZUMI	Não Iniciado	-	0	
3	DANIEL AUGUSTO CARLOS FERREIRA JORGE	RBD	390002248-00	Pneumatic pressure sensor update	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[1/2]	PREL	AMS	MAURO HIROSHI IZUMI	Não Iniciado	-	0	
3	EVERTON LUIZ SALLES		390003421-01	PACK non-FTI retrofit	SOMENTE PROTÓTIPO / PRÉ-SÉRIE	[2/2]	PREL	AMS	MAURO HIROSHI IZUMI	Não Iniciado	-	0	

Diagramas (0 OEs)

Figura 18 – Ferramenta de Controle de OEs - Visualização com filtro

Controle de OEs

Status OE:

Status Atividade:

Sub-bloco:

Executante:

OE:

Esconder Tabelas
Mostrar Tabelas
X

AMS (0 OEs)

Diagramas (12 OEs)

Elétrica (18 OEs)

Prioridade	OE	Tecnologia	HH	<<	11 Sep	12 Sep	13 Sep	14 Sep	15 Sep	16 Sep	17 Sep	18 Sep	19 Sep	20 Sep	21 Sep	22 Sep	23 Sep	24 Sep	25 Sep	26 Sep	27 Sep	28 Sep	29 Sep	30 Sep	01 Oct	02 Oct	03 Oct	04 Oct	05 Oct	06 Oct	07 Oct	08 Oct	>>
1	390003302-00	Elétrica	08																														
2	390003084-00	Elétrica	002																														
2	390003005-00	Elétrica	01:00																														
2	390002616-01	Elétrica	00																														
3	390002244-01	Elétrica	5680																														
3	390002779-00	Elétrica	00																														
3	390002870-00	Elétrica	00																														
3	390002864-00	Elétrica	00																														
3	390002726-01	Elétrica	00																														
3	390003181-02	Elétrica	00																														
3	390003616-00	Elétrica	010																														

Figura 19 – Ferramenta de Controle de OEs - Visualização de planejamento

usar uma planilha Excel compartilhada para o planejamento de prioridades e recursos. Considerando que não há documentação para as ferramentas do SAG, o estagiário focou em deixar o código-fonte da página auto-explicativo, com comentários que facilitem o desenvolvimento de possíveis futuras melhorias.

7 Conclusões e Considerações

Esse documento apresentou a metodologia adaptada e utilizada no contexto do estágio realizado na Embraer, onde foram desenvolvidas diversas ferramentas para auxílio à gestão da campanha de certificação da aeronave multimissão KC-390.

Para definição e otimização da metodologia descrita, o estagiário realizou mais de 40 tarefas do Jira. Dentre essas tarefas havia o desenvolvimento de novas ferramentas, semelhantes à tarefa de recriação da ferramenta de controle de OEs, descrita no capítulo 6 e também o desenvolvimento de melhorias em ferramentas já existentes. Houveram também modificações menores (em média uma hora de trabalho), para as quais não eram criadas tarefas no Jira.

A metodologia possibilitou que as tarefas fossem melhor ordenadas e desenvolvidas de maneira mais eficiente. Um dos principais pontos positivos dela foi a autonomia dada ao estagiário (desenvolvedor), não necessitando de muitas horas de atenção dos supervisores ou de outros engenheiros. Essa metodologia possivelmente será utilizada para os novos estagiários que continuarão o desenvolvimento do SAG ou até em outros programas da empresa. O Jira deverá ser mantido como ferramenta de criação e gestão das tarefas, pois ele possui as ferramentas e dados necessários para isso. As ferramentas desenvolvidas no decorrer do estágio tiveram bons retornos dos engenheiros usuários do SAG. Supervisores de outros programas gostaram da ferramenta de controle de OEs e pediram ajuda para adaptá-la aos seus próprios times e programas.

Como trabalho futuro, seria possível a modularização do SAG. Tratá-lo como um projeto de grande porte, utilizando-se de conceitos de projeto de software, para que ele pudesse ser facilmente iniciado e utilizado em novos programas que a Embraer possua.

Por fim, acredito que a formação ampla do curso de Engenharia de Controle e Automação ajudou no decorrer do estágio. A disciplina de bancos de dados mostrou-se essencial para o trabalho desenvolvido, visto que todas as ferramentas do SAG buscavam suas informações em tabelas. Todas as disciplinas de programação e desenvolvimento influenciaram para aprimorar a lógica de programação e o funcionamento de algoritmos. Finalizando, o estágio foi de grande aprendizado, o conhecimento em desenvolvimento web era pequeno antes do trabalho, o que tornou-se um obstáculo a superar. Porém, novamente, com o ensino amplo do curso, o estudante torna-se capaz de aprender de forma autônoma, algo valioso no papel do engenheiro. Assim, com dedicação foi possível realizar um trabalho que obteve retornos satisfatórios por parte do time de desenvolvimento de produto do KC-390.

Referências

- [1] METODOLOGIA de desenvolvimento de Software. Disponível em: <<https://www.devmedia.com.br/metodologia-de-desenvolvimento-de-software/1903>>. 15
- [2] TRAJETÓRIA Embraer. Disponível em: <<http://institutoembraer.org.br/centrohistoricoembraer/t>>. 19
- [3] INSTITUTO de Fomento e Coordenação Industrial. Disponível em: <<http://www.ifi.cta.br/index.php/produtos-certificacao/certificacao-de-produto-aeroespacial>>. 21
- [4] EMBRAER Defense & Security | KC-390. Disponível em: <<http://kc-390.com/>>. 21
- [5] WATERFALL Software Development Model. Disponível em: <<https://www.oxagile.com/company/blog/the-waterfall-model/>>. 27
- [6] WHAT is Incremental model - advantages, disadvantages and when to use it? Disponível em: <<http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/>>. 28
- [7] BOEHM, B. *Spiral Development: Experience, Principles, and Refinements*. [S.l.]: Carnegie-mellon univ pittsburgh pa software engineering inst, 2000. 28
- [8] KNIBERG, H. *Kanban vs Scrum - making the most of both*. [s.n.]. Disponível em: <<https://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf>>. 29, 39
- [9] SELECTING A DEVELOPMENT APPROACH. United States Department of Health and Human Services (HHS). Disponível em: <<https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>>. 39