

UNIVERSIDADE DE CAXIAS DO SUL

MATHEUS CARRA CONCATTO

**PROCESSO GAMIFICADO PARA GERENCIAMENTO ÁGIL DE PROJETOS NO
LABORATÓRIO DE CRIAÇÃO E APLICAÇÃO DE SOFTWARE**

CAXIAS DO SUL

2017

MATHEUS CARRA CONCATTO

**PROCESSO GAMIFICADO PARA GERENCIAMENTO ÁGIL DE PROJETOS NO
LABORATÓRIO DE CRIAÇÃO E APLICAÇÃO DE SOFTWARE**

Monografia apresentada como requisito para a obtenção do Grau de Bacharel em Sistemas de Informação da Universidade de Caxias do Sul.

Orientador: Prof Ms. Iraci Cristina da Silveira de Carli.

CAXIAS DO SUL

2017

MATHEUS CARRA CONCATTO

**PROCESSO GAMIFICADO PARA GERENCIAMENTO ÁGIL DE PROJETOS NO
LABORATÓRIO DE CRIAÇÃO E APLICAÇÃO DE SOFTWARE**

Monografia apresentada como requisito
para a obtenção do Grau de Bacharel em
Sistemas de Informação da Universidade
de Caxias do Sul

Orientador: Prof Ms. Iraci Cristina da
Silveira de Carli.

Aprovado (a) em ____/____/____

Banca Examinadora:

Presidente

Prof Ms. Iraci Cristina da Silveira de Carli
Universidade de Caxias do Sul - UCS

Examinadores:

Prof. Esp. Daniel Antônio Faccin
Universidade de Caxias do Sul - UCS

Prof. Dra. Elisa Boff
Universidade de Caxias do Sul - UCS

RESUMO

Este trabalho propõe um processo de gerenciamento de projetos, utilizando elementos de jogos. O processo é resultado da integração de práticas das metodologias tanto tradicionais quanto ágeis. A necessidade deste trabalho surgiu a partir do Laboratório de Criação e Aplicação de Software, localizado na Universidade de Caxias do Sul. Este local recebe projetos de diversas frentes, seja da própria universidade ou advindos do mercado. Atualmente não tem fins lucrativos e é de suma importância que os projetos não tenham custo. Devido a ser constituído por alunos (além dos professores) e não existir um comprometimento formalizado dos colaboradores, a rotatividade e quantidade de membros da equipe são alguns dos desafios enfrentados pelo laboratório. O coordenador de projeto tem dificuldades no registro das tarefas por esta comunicação não estar centralizada. O trabalho propõe a organização do processo do laboratório adaptando o processo atual. A proposta consiste em uma forma de administração dos projetos usufruindo do dinamismo proposto pela metodologia ágil agregado ao controle das formas tradicionais, engajando a equipe com as técnicas relacionadas a jogos. Além disso, será avaliado de forma qualitativa.

Palavras-chave: Projetos, Gerenciamento de Projetos, Desenvolvimento de Software, Gamificação, Métodos Ágeis.

ABSTRACT

This work proposes a project management process, using game elements. The process is a result of the integration of practices of traditional and agile methodologies. The need for this work arose from the Laboratory of Software Creation and Application, located at the University of Caxias do Sul. This site receives projects from different fronts, either from the university itself or from the market. It is currently not for profit and it is of the utmost importance that the projects have no cost. Due to the fact that it consists of students (besides teachers) and there is no formalized commitment of employees, the turnover and quantity of team members are some of the challenges faced by the laboratory. The project coordinator has difficulty logging tasks because this communication is not centralized. The work proposes the organization of the laboratory process adapting the current flow. The proposal consists of a way of administering the projects using the dynamism proposed by the agile methodology added to the control of the traditional forms, engaging the team with techniques related to games. In addition, the promoted process will be applied to a pilot project and the results will be evaluated qualitatively.

Keywords: Projects, Project Management, Software Development, Gamification, Agile Methods.

LISTA DE FIGURAS

Figura 1: Ciclo de Vida dos Projetos	19
Figura 2: Grupos de processos de gerenciamento de projetos	20
Figura 3: Áreas de conhecimento do PMBOK®	22
Figura 4: Desenvolvimento incremental	25
Figura 5: Perfis definidos por Richard Bartle	44
Figura 6: Diagrama flow	47
Figura 7: Situações dos projetos	53
Figura 8: Processo de planejamento atual	54
Figura 9: Execução do projeto atual	55
Figura 10: Processo completo	56
Figura 11: Fluxo proposto	57
Figura 12: Fase de iniciação	58
Figura 13: Fase de planejamento	60
Figura 14: Levantar informações das áreas de processo	61
Figura 15: Planejar escopo	62
Figura 16: Fase de execução	64
Figura 17: Sprint	65
Figura 18: Fase de encerramento	65
Figura 19: Arquitetura Lógica e Física	75
Figura 20: Funcionalidades	77
Figura 21: Cadastrar usuários - Administração do Redmine	78
Figura 22: Cadastrar usuários - Novo utilizador	79
Figura 23: Cadastrar usuários - Vincular usuário ao projeto - Administração	79
Figura 24: Cadastrar usuários - Vincular usuário ao projeto - Novo membro	79
Figura 25: Registrar tarefas - Cadastrar tarefa	81
Figura 26: Registrar tarefas - Tempo de trabalho	82
Figura 27: Registrar tarefas - Incluir tempo	83
Figura 28: Registrar backlog de produto - Nova versão	84
Figura 29: Registrar backlog de produto: Confirmar nova versão	84
Figura 30: Registrar backlog de produto - Menu Version Planning	85
Figura 31: Registrar backlog de produto: Taskboard Version Planning	85
Figura 32: Registrar sprints - Taskboard	86
Figura 33: Apresentar taskboard - Taskboard	87
Figura 34: Atribuir pontuação ao colaborador - Administração do Redmine	88
Figura 35: Atribuir pontuação ao colaborador - Acessar configurações	88
Figura 36: Atribuir pontuação ao colaborador - Configurações	89
Figura 37: Ranking de colaboradores - Acessar ranking	90
Figura 38: Ranking de colaboradores - Ranking	91
Figura 39: Registrar documentações de projeto - Menu documentos	91
Figura 40: Registrar documentações de projeto - Novo documento	92

LISTA DE TABELAS

Tabela 1: Resultados da Avaliação	71
Tabela 2: Questionário	94
Tabela 3: Fase de iniciação.....	97
Tabela 4: Fase de planejamento	98
Tabela 5: Fase de execução	100
Tabela 6: Fase de encerramento	101

LISTA DE ABREVIATURAS E SIGLAS

BPMN	<i>Business Process Model and Notation</i>
CRC	<i>Classe, responsabilidade, colaboração</i>
EAP	<i>Estrutura Analítica de Projeto</i>
GP	<i>Gerência de Projetos</i>
LSD	<i>Lean Software Development</i>
MDP	<i>Método do Diagrama de Precedência</i>
PGP	<i>Plano de Gerenciamento de Projeto</i>
RH	<i>Recursos Humanos</i>
TI	<i>Tecnologia da Informação</i>
XP	<i>eXtreme Programming</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 PROBLEMA DE PESQUISA.....	13
1.2 QUESTÃO DE PESQUISA	13
1.3 OBJETIVO DA PESQUISA	14
1.3.1 Objetivos específicos.....	14
1.4 METODOLOGIA	14
1.5 ESTRUTURA DO TRABALHO	16
2 GERÊNCIA DE PROJETOS	17
2.1 GERÊNCIA DE PROJETOS TRADICIONAL	17
2.1.1 Projeto.....	17
2.1.2 Gerenciamento de projetos	18
2.1.2.1 Ciclo de vida dos projetos	19
2.1.2.2 Processos de gerenciamento de projetos	20
2.1.2.3 Áreas de Conhecimento	22
2.2 GERÊNCIA DE PROJETOS ADAPTATIVA	24
2.2.1 Metodologias ágeis	25
2.2.1.1 Manifesto Ágil.....	26
2.2.1.2 Scrum.....	27
2.2.1.2.1 Papéis	28
2.2.1.2.2 Artefatos	30
2.2.1.2.3 Cerimônias e Sprint.....	31
2.2.1.3 Lean Software Development	33
2.2.1.4 Extreme Programming.....	35
2.2.1.4.1 Princípios.....	35
2.2.1.4.2 Práticas	36
2.3 TRADICIONAL X ADAPTATIVO	39
3 GAMIFICAÇÃO.....	41
3.1 JOGOS.....	41
3.2 ANÁLISE DE PERFIS	43
3.3 MOTIVAÇÃO.....	45

4 TRABALHOS RELACIONADOS	48
5 PROPOSTA DE SOLUÇÃO	50
5.1 APRESENTAÇÃO DO CENÁRIO	50
5.2 PROCESSO ATUAL.....	51
5.3 PROPOSTA DE ADAPTAÇÃO DO PROCESSO	55
5.3.1 Iniciação	57
5.3.2 Planejamento	59
5.3.3 Execução.....	62
5.3.4 Encerramento	65
5.4 FORMAS DE ENGAJAMENTO	66
6 DESENVOLVIMENTO DA SOLUÇÃO	68
6.1 AVALIAÇÃO DA FERRAMENTA	68
6.1.1 NBR ISO/IEC 9126 e 14598.....	68
6.1.2 Resultados da avaliação.....	70
6.2 IMPLEMENTAÇÃO DA SOLUÇÃO	74
6.2.1 Extensões	74
6.2.2 Instalação, Implantação e Treinamento.....	76
6.2.3 Funcionalidades	77
6.2.3.1 Cadastrar usuários	78
6.2.3.2 Registrar tarefas.....	80
6.2.3.3 Registrar backlog de produto	83
6.2.3.4 Registrar sprints	86
6.2.3.5 Priorização de histórias/requisitos	86
6.2.3.6 Apresentar taskboard	86
6.2.3.7 Atribuir pontuação ao colaborador	87
6.2.3.8 Ranking de colaboradores.....	90
6.2.3.9 Registrar documentações de projeto.....	91
6.2.3.10 Considerações finais	92
6.2.4 Avaliação.....	93
6.2.4.1 Iniciação	97
6.2.4.2 Planejamento	98
6.2.4.3 Execução.....	100

6.2.4.4 Encerramento	101
7 CONCLUSÃO	103
REFERÊNCIAS.....	105
APÊNDICE A – TERMO DE ABERTURA DE PROJETO	110
APÊNDICE B – PLANO DE GERENCIAMENTO DE PROJETO	112
APÊNDICE C – PLANO DE GERENCIAMENTO DE CUSTOS	114
APÊNDICE D – TERMO DE ENCERRAMENTO DO PROJETO	115

1 INTRODUÇÃO

Segundo Valeriano (2005), um projeto representa um conjunto de atividades dirigidas em prol de um objetivo, conforme os requisitos e as premissas definidas. Os projetos, mesmo que possuam assuntos, necessidades de recursos distintas e objetivos diferentes, devem ser geridos de maneira eficiente.

Uma sistemática deve ser utilizada para planejar, organizar, dirigir e controlar uma gestão de projetos de forma eficiente. Diante deste contexto, existem metodologias que promovem a execução dos projetos de formas distintas. As estratégias tradicionais, conforme Ribeiro e Arakaki (2006), podem ser aplicadas em qualquer projeto, contudo, não são apropriadas em ambientes com mudanças constantes. É adequada para um contexto estável. Sob outra perspectiva, as metodologias ágeis acolhem as mudanças e fazem o tratamento das mesmas durante o projeto sem impasses impostos pelas vias tradicionais. A comparação entre estratégias tradicionais e ágeis é inevitável. Cabe ao gerente de projeto realizar a análise conforme a demanda a ser atendida. Até o presente momento alguns trabalhos já foram publicados citando a utilização de uma metodologia ágil. O manifesto ágil reforça algumas premissas que caracterizam este tipo de metodologia. Desenvolver o software torna-se mais importante do que uma documentação complexa, trabalhar com o cliente, conseguir responder à mudanças e acima de tudo, valorizar mais os indivíduos do que as ferramentas.

Para a execução de um processo, é necessário que as pessoas estejam engajadas na aplicação do mesmo. Para isso, as práticas de Gamificação (ou Ludificação) utilizam-se de características dos jogos dentro dos processos para torná-los mais atraentes aos responsáveis de cada atividade. A evolução proposta aos colaboradores baseia-se em pontos, níveis e desafios, semelhante ao contexto de um videogame. O que distinguiu esta atividade das demais tarefas já aplicadas, que trouxe o envolvimento dos alunos, foi o uso de ferramentas motivadoras para o raciocínio (FERREIRA et al., 2016). Neste caso, os autores utilizaram o método em sala de aula durante a disciplina de Gerência de Projetos. Foram bem-sucedidos. Além disso, há registros do uso da Gamificação em ambiente empresarial. Por exemplo, a companhia aérea GOL¹ desenvolveu um jogo para os seus colaboradores aprimorarem seu atendimento ao cliente. Já a empresa Solera², que fornece soluções diversas em tecnologia, aplicou a Ludificação buscando novas ideias para o seu negócio aplicando uma técnica em três fases. Ambas as ideias promoveram o engajamento dos colaboradores, principal foco desta metodologia.

¹ Aennova. Disponível em: <<http://www.aennova.com/blog/?p=1081>>. Acesso em: 23 mar 2017.

² Aennova. Disponível em: <<http://www.aennova.com/blog/?p=825>>. Acesso em 23 mar 2017.

1.1 PROBLEMA DE PESQUISA

O Laboratório de Criação e Aplicação de Software iniciou seu trabalho em 2015 e atua dentro da Universidade de Caxias do Sul. Atualmente, sua equipe é constituída por professores, voluntários e estagiários remunerados da universidade. Seu objetivo é proporcionar o desenvolvimento de sistemas sob demanda e neste contexto, trazer experiência para os integrantes da equipe, que normalmente são iniciantes na área de TI.

O Laboratório recebe demandas de diversas frentes com temas diversos, ou seja, não há foco em um tipo de software em específico. As necessidades podem ser classificados entre acadêmicos, mercado e startup. Assim sendo, o controle de todos os projetos que são desenvolvidos em simultâneo torna-se um trabalho árduo e passível a falhas. Além disso, a alta rotatividade dos colaboradores trata-se de outro inconveniente devido à gestão do conhecimento ser inadequada e projetos em andamento são diretamente afetados. A administração existente é inadequada, visto que as informações são distribuídas e há desgaste em reuniões desnecessárias. A resistência dos usuários em utilizar uma ferramenta mútua para este controle causa algumas dificuldades para os líderes diagnosticarem a situação atual de um projeto e terem conhecimento das incumbências de cada colaborador no momento.

Cabe aos professores coordenar essa equipe. Contudo, em algumas ocasiões os mesmos não podem estar presentes. Logo, dependendo da situação, pode causar atrasos no projeto bem como a insatisfação do cliente. Conforme Chiavenato (2014), uma equipe saudável possui um ambiente favorável, motivação mútua para alcançar os objetivos e a comunicação clara entre os membros, resultante de um bom relacionamento. Os integrantes nos dias de hoje são resistentes à mudança, portanto, faz-se necessária uma mudança de cultura por um objetivo favorável a todos.

Vargas (2016) exemplifica o círculo em que este processo deverá ser aplicado. Em um ambiente em que as mudanças são frequentes, é indispensável que a gerência de projetos seja feita baseado nos objetivos. Além das mudanças exteriores que são inevitáveis, a constante circulação de pessoal e a pouca experiência da equipe torna este projeto ainda mais desafiador.

1.2 QUESTÃO DE PESQUISA

Que processo é adequado, que promova o engajamento da equipe, a ser aplicado para controle dos projetos realizados no laboratório?

1.3 OBJETIVO DA PESQUISA

O objetivo deste trabalho é propor um processo gamificado para o gerenciamento dos projetos no laboratório. Alguns processos do PMBOK® serão aplicados de acordo com as necessidades atuais. A gestão será feita a partir da conciliação entre metodologias tradicionais e ágeis, que permita dinamismo e rapidez nas entregas. Técnicas de gamificação serão utilizadas com o intuito de engajar a equipe a aderir ao processo.

O processo de desenvolvimento de software, que atualmente é o único serviço oferecido pelo laboratório, deve receber uma reavaliação do panorama atual em busca de melhorias. Uma ferramenta mais simples deve ser disponibilizada aos usuários em busca da redução destes obstáculos encontrados pela equipe. Os professores desejam ter a informação sobre cada projeto em qualquer momento. Logo, esta ferramenta deverá contemplar esta necessidade.

As técnicas definidas na gamificação buscam dinamizar a execução dos processos no laboratório e facilitar seu controle. A ideia de pontuação baseado no progresso será significativa para os dois lados. Age a favor dos professores (por visualizar o status das tarefas efetivamente e não de forma intuitiva) e a favor dos colaboradores, no momento que vencem os desafios propostos e podem ser beneficiados por isto. Uma das finalidades deste projeto é proporcionar a partir da gamificação e métodos de gerenciamento de projetos (que serão definidas no futuro) uma eficiente gestão.

1.3.1 Objetivos específicos

Os objetivos específicos deste trabalho estão definidos a seguir.

- a) Análise sobre atual processo do laboratório;
- b) Estudo sobre gerência de projetos e metodologias ágeis;
- c) Estudo sobre gamificação;
- d) Análise de perfis de pessoas, baseado na gamificação;
- e) Proposta de um processo de gerenciamento de projetos baseado em Gamificação;
- f) Pesquisa e/ou desenvolvimento de uma ferramenta para utilização na aplicação do processo;
- g) Avaliação qualitativa do processo.

1.4 METODOLOGIA

Será realizada uma pesquisa qualitativa exploratória neste trabalho. O projeto foi dividido em nove fases que estão detalhadas a seguir.

a) Análise do processo atual do laboratório.

Executar visitas de acompanhamento e reuniões com os professores. O intuito desta tarefa é entender como o laboratório funciona atualmente de maneira detalhada e identificar possíveis pontos de mudança. Além disso, entender os objetivos a serem alcançados pela equipe.

b) Realizar estudo sobre gerenciamento de projetos.

Estudar autores direcionados para GP (Gerência de Projetos) com foco em metodologias ágeis, que possam embasar o processo aplicado no laboratório e reforçar a importância deste projeto. Examinar livros, e-books e artigos sobre o assunto.

c) Realizar estudo sobre gamificação.

Buscar aprendizado sobre metodologias de gamificação e aplicações dentro de ambientes empresariais. Localizar autores especializados em livros e artigos que relatam aplicações em casos reais.

d) Elaborar processo.

A partir dos estudos anteriores e o entendimento completo das operações, elaborar o processo gamificado que será aplicado no laboratório. Utilizar ferramentas que suportam a modelagem BPMN (*Business Process Model and Notation*) para desenhar o fluxo de atividades.

e) Apresentar processo proposto.

O processo deverá ser aprovado pelos professores do laboratório e caso necessário, apresentada aos colaboradores em uma reunião.

f) Levantar ferramentas que deverão suprir o processo proposto e/ou ver sobre desenvolvimento.

O processo proposto deverá ser ministrado por uma ferramenta que deve agir como um facilitador. Portanto, este estudo deverá ser minucioso e as atividades devem estar claras. Assim, a ferramenta escolhida ou o desenvolvimento a ser realizados serão adequados. Pesquisar por aplicações voltadas para gerência de projetos.

g) Escolher ferramenta ou iniciar desenvolvimento.

Após escolha das ferramentas ou optar por desenvolvimento, iniciar a implantação da mesma. No caso da necessidade de programação, a aplicação deverá ser finalizada e testada antes de ser disponibilizada. Instalar aplicações levantadas no passo anterior e testar de forma breve se suas opções suprem o processo proposto.

h) Avaliar resultados.

Realizar uma análise qualitativa dos resultados apresentados baseado em um questionário para a equipe. A partir destes dados, tomar decisões sobre como

prosseguir juntamente com o apoio dos professores. Esta análise deve ser realizada via questionário.

1.5 ESTRUTURA DO TRABALHO

O trabalho está apresentado em cinco grupos: introdução, referencial teórico, trabalhos relacionados, proposta de solução e considerações parciais.

No presente capítulo 1, é feita a introdução do contexto do problema, bem como os objetivos a serem atingidos com este projeto.

Os capítulos 2 e 3 constituem o referencial teórico. A seção 2 aborda a conceituação de projetos e as duas metodologias de interesse deste trabalho, tradicional e ágil. Cada estratégia constitui um subcapítulo, que possui ramificações tornando a informação mais didática. O capítulo 3 incorpora a teoria da gamificação e a sua relação direta com a motivação, baseado no perfil do indivíduo. Possui três sessões.

O capítulo 4 demonstra alguns trabalhos que estão relacionados ao objetivo desta monografia. Seção única.

O capítulo 5 apresenta de modo mais detalhado o contexto em que a proposta será aplicada, além do detalhamento do processo defendido. Nesta seção, o ambiente atual é detalhado nos dois primeiros subcapítulos (5.1 e 5.2). O fluxo a ser aplicado está presente na seção 4.3 e está dividido em quatro ramificações.

O capítulo 6 apresenta a avaliação de ferramentas que poderiam atender a proposta e a aplicação escolhida, bem como sua estrutura e as funcionalidades disponíveis. Além disso, apresenta a avaliação do processo e o veredito final.

A seção 7 relata as conclusões após a pesquisa, análise e implantação do processo proposto no trabalho.

2 GERÊNCIA DE PROJETOS

Nesta seção serão apresentados os conceitos voltados à gerência de projetos que são aplicados durante o decorrer do estudo.

2.1 GERÊNCIA DE PROJETOS TRADICIONAL

Neste capítulo será apresentado o conteúdo referente à GP tradicional. Recebe esta nomenclatura devido a ser o modelo que iniciou a gestão por projetos. É suportado hoje pelo órgão PMI (*Project Management Institute*).

2.1.1 Projeto

Diversos autores já divulgaram seu conceito de projeto. O PMI, através do seu guia PMBOK® (2013), classifica da seguinte forma: “projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo”. Um projeto não possui uma vertente em específico, ou seja, aplicar uma administração por projetos não se limita apenas a software. A aplicação correta das práticas e o gerenciamento conciso de todas as áreas de conhecimento, de acordo com o cenário e a estratégia da empresa, são as chaves de um projeto bem-sucedido.

Heldman (2009) atribui as seguintes características a um projeto:

- a) Não existem projetos iguais;
- b) Projetos possuem período, ou seja, início e fim;
- c) Pode-se dizer que um projeto é triunfante ao concluir as especificações das partes interessadas.

O autor cita que os projetos são únicos, cada um possui suas especificações que os diferenciam. Ou seja, por terem tratamentos diferentes de partes interessadas, custos e prazos diferentes, isto os torna exclusivos. Além disso, pode-se dizer que são temporários devido a possuírem uma data de início e uma data de término, que deve ser cumprida. Este é um dos principais desafios de um gestor de projetos, pois ao cumprir o prazo definido, na teoria, seu trabalho foi desempenhado com sucesso. Outra característica citada pelo autor explica que um projeto deve ter metas a completar. No descumprimento das mesmas, o projeto resultou em um insucesso e cabe ao gerente de projeto determinar as causas e evitá-las em outra ocasião. Além disso, o projeto pode não se tornar mais viável. Isto está diretamente atrelado ao gerenciamento de riscos, visto que uma adversidade rigorosa repentina no ambiente pode modificar completamente os próximos passos do projeto e caso não tenha saída,

o trabalho deve ser abortado. Por fim, um projeto deve atingir as expectativas das partes interessadas. Neste ponto, pode-se dizer que o mesmo é qualificado.

É importante destacar também a diferença entre projetos e operações. Podem possuir conceitos próximos, porém seus fins são totalmente diferentes. Um projeto é único, enquanto que as operações são repetitivas e uniformes. O PMBOK® (2013) cita que as operações são ligadas aos processos de uma organização. De outro modo, os projetos são fonte de dúvida no que tange o seu objetivo, seja produto, serviço ou processo. De certa forma, seus resultados podem se tornar imprevisíveis.

Além disso, Cavalcanti (2016) afirma que as operações caracterizam-se pela repetibilidade independentemente do cenário. Portanto, além de projetos, há operações relacionadas à produto, serviço e processo. Vários projetos podem conter operações em comum, porém, seus fins são diferentes e as outras variáveis, abordadas nas áreas de conhecimento, também serão únicas.

2.1.2 Gerenciamento de projetos

O gerenciamento de projeto eficiente baseia-se em planejamento. O sucesso da organização, direção e controle de todas as áreas exige que esta primeira fase seja executada com êxito, da melhor maneira possível. Um bom planejamento evita surpresas futuras (normalmente desagradáveis), além de ter conhecimento de como se portar em situações adversas.

O conceito de gerenciamento de projeto é amplo. Busca a harmonia entre conhecimento, habilidades e ferramentas para atingir os requisitos definidos pelas partes interessadas (PMBOK®, 2013). Além disso, cita também sobre a aplicação dos processos separados em cinco grupos, que serão detalhados no decorrer do estudo.

Larson (2016) categoriza que o gerenciamento de projetos corrobora com o incentivo ao relacionamento entre os integrantes da equipe. Além disso, Kerzner (2006) afirma que o gerenciamento de projetos, por envolver colaboradores com faculdades diversas que estão espalhadas por áreas de conhecimento, promove uma forma de comunicação horizontal. Isso significa que não há uma cadeia de comando (exceto o gerente de projeto que é responsável por todas as áreas). Esta maneira vai contra a maneira tradicional, chamada de vertical, que possui uma hierarquia de comando mais extensa. O autor complementa que a abordagem horizontal mostra-se mais competente que a tradicional. A ausência de uma autoridade contínua incentiva a produtividade, eficiência e eficácia (Kerzner, 2006).

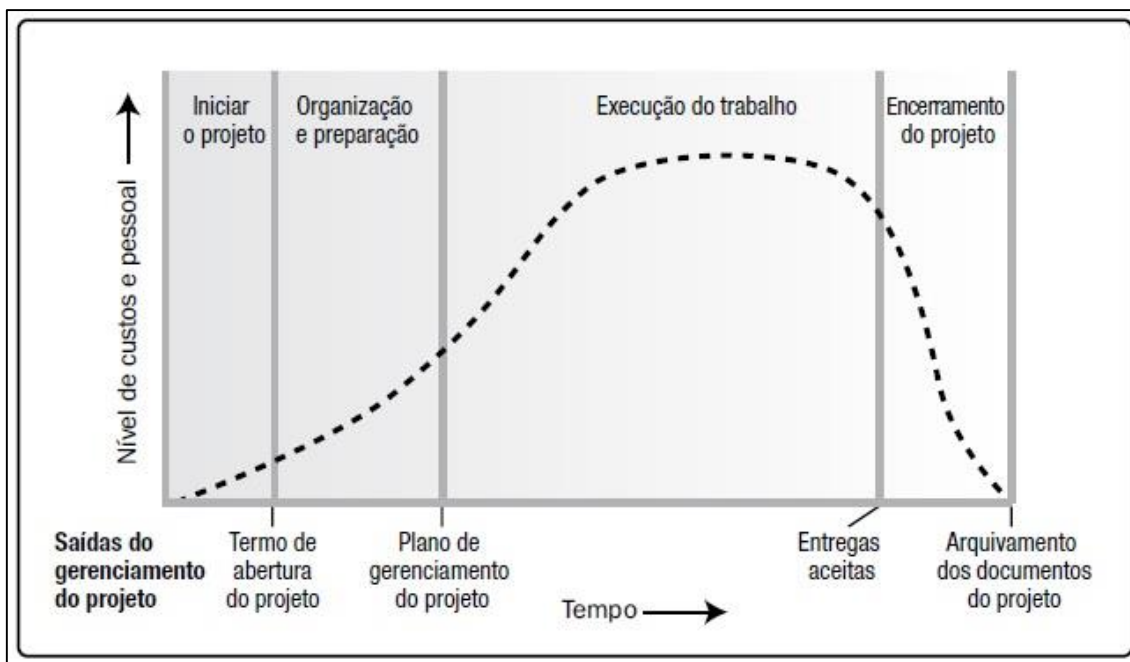
Os projetos também possuem ciclo de vida, processos de gerenciamento e áreas de conhecimento. Os três temas estão detalhados nos subcapítulos a seguir.

2.1.2.1 Ciclo de vida dos projetos

Todo projeto possui um ciclo de vida, ou seja, início, meio e fim. Este ciclo pode ser detalhado de acordo com um modelo adotado pela própria empresa. Alguns autores sugerem algumas situações como padrão e que acontecem independentemente do tipo de negócio. Estas situações estão exemplificadas conforme a Figura 1.

As quatro fases definidas exigem níveis diferentes de esforço da equipe e alocação de recursos. Esta curva representa um padrão, porém o PMBOK® (2011) reitera que isto não é uma lei, ou seja, um projeto pode ser custoso durante a sua organização tanto quanto na sua execução, por exemplo.

Figura 1: Ciclo de Vida dos Projetos



Fonte: PMBOK®, 2011, p. 39

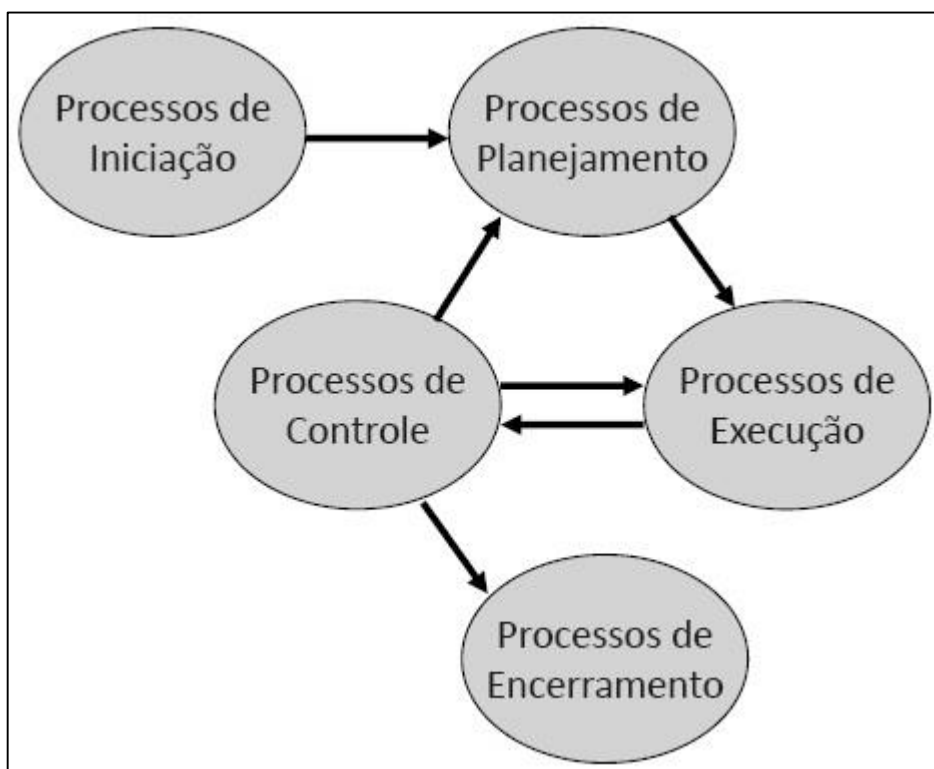
Para que o projeto evolua de estágio, existem marcos que classificam esta troca de fase. Ademais, Kanabar e Warburton (2012) afirmam que o ciclo de vida de um projeto pode gerar um produto ou então acompanhar o ciclo de vida do mesmo até o fim. Um produto também possui esta característica e o acompanhamento do projeto até o seu fim ocorre de acordo com o negócio da empresa. Normalmente em software, por exemplo, o projeto permanece ativo até a vida útil do aplicativo desenvolvido (KANABAR; WARBURTON, 2012).

A definição do ciclo de vida também tem o encargo de demonstrar o tempo empregado em cada fase. Em cenários de projetos sendo executados simultaneamente na empresa, torna-se fundamental o controle das tarefas que são executadas em cada projeto visto que cada um possui seu ciclo (LARSON, 2016).

2.1.2.2 Processos de gerenciamento de projetos

Os processos de gerenciamento de projetos consistem em um agrupamento dos controles a serem executados baseado na sua finalidade. Cada grupo pode comunicar-se diretamente com o outro, ou seja, há processos que influenciam as atividades de outra área presente em um grupo totalmente diferente. É importante salientar também o reforço do PMBOK® (2011) sobre explicar que diferentemente do ciclo de vida, os processos não são caracterizados como fases de projeto. Cada grupo pode ser aplicado em uma ou mais fases do projeto, portanto, não deve-se confundir os conceitos. Todas as categorias possuem a sua devida importância. A Figura 2 demonstra seu fluxo com clareza.

Figura 2: Grupos de processos de gerenciamento de projetos



Fonte: CAVALCANTI, 2016, p. 40

As categorias de processos estão descritas abaixo, conforme PMBOK® (2011).

a) Processos de Iniciação

Nesta fase, são realizados todos os processos necessários para dar andamento ao projeto ou parte do mesmo. Portanto, neste momento são reservados os recursos necessários (financeiros e pessoal) para que sejam atendidos os desejos das partes interessadas. Além disso, permite-se nestes processos que os interessados possam imaginar o futuro deste projeto no que tange o seu andamento e os seus resultados.

b) Processos de Planejamento

A partir dos processos de iniciação, a equipe pode planejar o destino dado a cada recurso que foi alocado para a sua área. Boa parte da documentação é realizada baseado nos processos de planejamento. Devido às mudanças que normalmente ocorrem em um projeto, estes processos são revistos à medida que são necessários. Este é um exemplo de grupo que pode ser consultado em diversas fases do trabalho em andamento.

c) Processos de Execução

Este grupo, conforme a denominação, compromete-se às atividades relacionadas à execução do projeto. Na aplicação destes processos, arrisca-se a visitar os processos de planejamento e inclusive de iniciação, caso algum imprevisto tenha ocorrido. Vale recordar que a execução deve ser feita em conformidade com as especificações e requisitos (TRENTIM, 2014).

d) Processos de Monitoramento e Controle

Os processos de monitoramento e controle são responsáveis por realizar o diagnóstico das áreas e recomendar atitudes a serem tomadas para aprimorar o projeto. Ou seja, avaliações devem ser feitas de maneira contínua de modo que a necessidade de mudança seja detectada de forma prévia. Porém, inevitavelmente pode suceder uma situação que seja necessária uma ação corretiva, também responsabilidade deste grupo de processos.

e) Processos de Encerramento

Ao finalizar todos os processos dos demais grupos e o projeto for concluído (com êxito ou não), são executadas as atividades que caracterizam seu encerramento. Estas atividades englobam o retorno ao cliente sobre os resultados alcançados, registro de lições aprendidas e levantamento do desempenho das equipes em cada área. Os frutos das demais áreas são examinados na execução dos processos deste grupo.

2.1.2.3 Áreas de Conhecimento

O guia também instrui sobre as dez áreas de conhecimento pertinentes para uma eficiente gestão de projetos. Cada uma possui a sua importância dentro da administração. De acordo com o PMBOK® (2015), uma área de conhecimento inclui diversos conceitos que compõe um campo de estudo da GP em específico. A Figura 3 as demonstra e nos próximos subcapítulos há o detalhamento de cada uma.

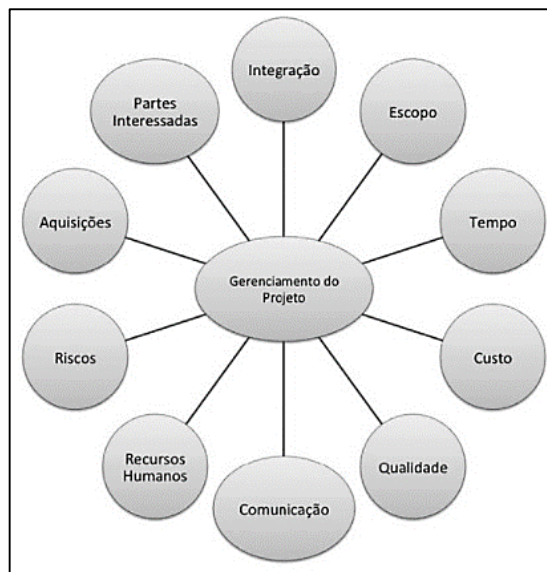
a) Gerenciamento de Integração do Projeto

O Gerenciamento de Integração é responsável por garantir o alinhamento entre as demais áreas. Permite a coordenação dos processos, portanto, é de suma importância para o andamento do projeto. Recebe a responsabilidade da resolução dos conflitos entre as demais áreas, garantir que a dependência entre as mesmas exista e seja monitorada, além de mensurar e distribuir os recursos. (PMBOK®, 2013).

b) Gerenciamento de Escopo do Projeto

O Gerenciamento de Escopo do Projeto determina como será dado o andamento das fases posteriores. Uma aplicação bem-sucedida desta área de conhecimento é essencial para a continuidade do projeto como um todo. De acordo com Trentim (2014) o escopo comporta todo o detalhamento do projeto. Uma das gerências mais importantes dentro das áreas de conhecimento, suas alterações atingem diretamente todas as demais áreas. Portanto, suas mudanças devem ser analisadas, controladas e previstas (se possível).

Figura 3: Áreas de conhecimento do PMBOK®



Fonte: TRENTIM, 2014, p. 56

c) Gerenciamento de Tempo do Projeto

Na atualidade, diante de um cenário que as respostas devem ser obtidas de forma mais ágil possível, o gerenciamento de tempo está entre as áreas mais importante dentro de um projeto. Neste momento são estabelecidas as estimativas, cronogramas e conseqüentemente o prazo que deve ser respeitado. Os cronogramas devem ser controlados durante o andamento do projeto. É de suma importância que um projeto cumpra com os prazos definidos. O cumprimento das metas à tempo é uma restrição presente em todos os projetos, portanto, o cronograma deve gerenciar a aplicação de todas estas atividades (VALERIANO, 2015).

d) Gerenciamento de Custos do Projeto

O gerenciamento de custos realiza a administração dos recursos financeiros oferecidos ao projeto. Quanto maior for a minuciosidade deste controle, menor é a chance de surpresa de gastos desnecessários. Os motivos para este tipo de acontecimento podem acontecer em todas as demais áreas de conhecimento, logo, realça ainda mais a importância deste domínio para as demais. O gerenciamento de custos centraliza todas as atividades que envolvem recursos, proporcionando o cumprimento do projeto e obedecendo o orçamento estipulado (PMBOK®, 2013).

e) Gerenciamento da Qualidade do Projeto

A maneira como esta gerência deverá ser aplicada varia de acordo com cada projeto. Valeriano (2015) cita a classificação dos aspectos abordados nesta área como intangíveis, por exemplo, atendimento ao cliente, e tangíveis, por exemplo, preço de uma reforma. Em suma, garantir a qualidade do projeto significa atender as expectativas do cliente e das partes interessadas.

f) Gerenciamento de Recursos Humanos do Projeto

Esta área de conhecimento lida com todos os aspectos relacionados às pessoas dentro do projeto, por exemplo, treinamentos e levantamento de especialidades. Deve garantir que os recursos humanos sejam utilizados da melhor forma possível, ou seja, buscando o melhor desempenho dos colaboradores em cada função específica. O gerente de projeto deve ser capaz de negociar e compreender as motivações de cada indivíduo (TRENTIM, 2014).

g) Gerenciamento de Comunicações do Projeto

As informações do projeto devem ser transmitidas a todos os membros da equipe de maneira concisa e clara. O gerenciamento das comunicações do projeto é responsável por propor maneiras de propagação da informação no ambiente sem distorcê-la. Falhas ocasionadas nesta gerência impactam diretamente nas demais, pois uma informação incorreta pode gerar custos necessários e alterações desnecessárias de cronograma. O contato entre as pessoas e o controle das

informações transferidas entre elas são de responsabilidade desta gestão (CAVALCANTI, 2016).

h) Gerenciamento de Riscos do Projeto

Os riscos de um projeto podem trazer benefícios ou malefícios para o mesmo. Caso identificados como oportunidades, cabe ao gerente de projeto avaliar como deve ser abordado e o que intervém nas demais áreas. Contudo, se o risco for classificado como ameaça, o projeto obriga-se a orientar como enfrentar o mesmo da melhor maneira possível. TRENTIM (2014) cita que o gerenciamento de riscos procura reduzir a possibilidade de efetivos negativos no projeto e favorecer os acontecimentos positivos.

i) Gerenciamento de Aquisições do Projeto

O relacionamento com os fornecedores também é tema de conhecimento dentro do gerenciamento de projetos. O controle eficiente dos contratos com prestadores de serviço e produto garante que não ocorra surpresas desagradáveis durante o andamento normal do projeto. “O gerenciamento de aquisições do projeto abrange os processos de gerenciamento de contratos e controle de mudanças que são necessários para desenvolver e administrar contratos ou pedidos de compra por membros autorizados da equipe de projeto” (PMBOK®, 2013).

j) Gerenciamento das Partes Interessadas do Projeto

Os *stakeholders* (ou partes interessadas) são as pessoas (física ou jurídica) que demonstram interesse e buscam ganhar proveito com o projeto de alguma forma. Como exemplo, um stakeholder pode ser o próprio cliente que solicitou o desenvolvimento de alguma aplicação ou a gerência da própria empresa. Esta gerência é responsável pelo tratamento das partes, ou seja, mantê-los informados, engajá-los, convencê-los. TRENTIM (2014) descreve que é possível aplicar estratégias específicas conforme os stakeholders, lembrando, sempre de acordo com o projeto.

2.2 GERÊNCIA DE PROJETOS ADAPTATIVA

Este capítulo busca conceituar e detalhar o funcionamento de algumas estratégias adaptativas (ou ágeis). A seguir, encontra-se a base teórica das metodologias ágeis e a explicação sobre o Scrum, LSD (Lean Software Development) e XP (Extreme Programming). As metodologias ágeis recebem este nome devido a serem mais dinâmicas que uma abordagem tradicional, menos burocrática e mais preocupada com o desenvolvimento do produto. Desta forma, evitando algumas práticas que poderiam prolongar a entrega.

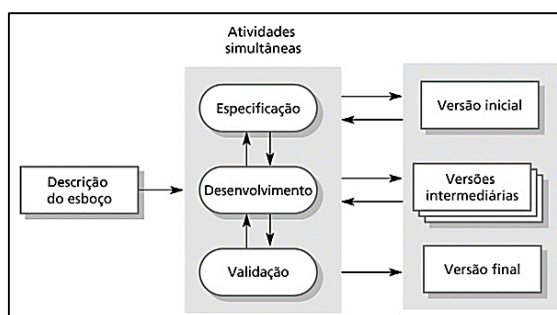
2.2.1 Metodologias ágeis

A abordagem adaptativa surgiu da necessidade das empresas em entregar seus projetos de maneira mais rápida e de acordo com os requisitos. A forma de desenvolvimento tradicional, normalmente em cascata, não apresentava a agilidade necessária. Na década de 90 os métodos tradicionais não satisfaziam as necessidades de desenvolvimento e com isso, novas ideias com o intuito de agilidade foram propostas (SOMMERVILLE, 2011).

Além disso, o que provocou o crescimento do uso deste tipo de metodologia é a constante mudança de requisitos que pode acontecer durante a execução de um projeto. A definição de requisitos é o início de todo e qualquer desenvolvimento de software. Nesta fase, são levantadas todas as obrigações do cliente que devem ser atendidas pelo software, portanto, ela é a base para todo o restante. Contudo, por mais eficiente que seja o analista de negócio que recebeu estas necessidades, os possíveis diálogos com o cliente podem modificar radicalmente os requisitos. Isto pode ser identificado durante o desenvolvimento ou pior, na entrega do software já finalizado.

As metodologias adaptativas utilizam-se do conceito de desenvolvimento incremental. Este método instrui que sejam feitas várias entregas parciais ao cliente ao invés do software já concluído em uma única apresentação. Este método permite que as mudanças nos requisitos não causem um atraso excessivo ao projeto, pois o mesmo está em constante desenvolvimento. Além disso, esta maneira também promove maior comunicação com o cliente, visto que cada incremento é entregue e apresentado ao cliente. Posteriormente é realizada a aprovação ou a reprovação. Neste momento cabe ao analista de sistemas dar andamento ao projeto baseado no retorno que recebeu da parte interessada. O desenvolvimento declara-se como finalizado no momento que todas as especificações definidas pelo analista de negócio e cliente estejam no último incremento aplicado. A Figura 4 apresenta como é o desenvolvimento incremental.

Figura 4: Desenvolvimento incremental



Fonte: SOMMERVILLE, 2011, p 22.

2.2.1.1 Manifesto Ágil

Na época, em meados da década de 90, surgiram diversos processos alternativos em oposto à metodologia tradicional, chamada também de prescritiva. As metodologias ágeis possuem em comum o desenvolvimento iterativo, que procura a evolução do produto em bloco conforme a colaboração entre cliente e equipe (PRIKLADNICKI, WILLI, MILANI, 2014).

O Manifesto Ágil trata-se da união de 17 especialistas em metodologias ágeis com conceitos diferentes, mas opiniões semelhantes. Deste encontro, foram definidas quatro premissas que embasam todas as formas de desenvolvimento ágil. São elas:

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano;

(BECK et al; 2001)

Estes quatro valores revolucionaram a maneira com que os projetos poderiam ser administrados na época. No momento que foi identificado que é possível gerenciar um trabalho sem necessidade de burocracia e processos que poderiam se tornar maçantes, muitas empresas aderiram ao movimento.

As ferramentas e processos jamais devem ser deixadas de lado. Porém, o relacionamento entre as pessoas passou a ser mais valorizada por ser a forma mais veloz de transmitir informação. Além disso, ao utilizar uma metodologia ágil é necessário promover a reflexão sobre o que deve ser realmente documentado. A entrega do projeto é o principal objetivo e qualquer registro que pode tornar-se desnecessário não deve tomar o tempo de um colaborador. A mais questionável de todas as premissas é a questão de relacionamento com o cliente. Muitas vezes o tratamento informal permite uma aproximação maior desta parte interessada com a equipe do projeto, logo, minimiza-se a chance de erros. Contudo, a segurança descartada nesta abordagem aumenta a probabilidade de atitudes antiéticas. A colaboração entre as partes, cliente e equipe, é essencial e não se indica criar mecanismos de proteção exacerbados (PRIKLADNICKI; WILLI; MILANI, 2014). Ademais, as mudanças imprevistas no decorrer do projeto podem ocorrer. Na aplicação incremental de uma metodologia ágil, são menos impactantes.

Os autores do manifesto ágil também elaboraram doze princípios que são de suma importância. São eles:

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- Software funcional é a medida primária de progresso.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

(BECK et al; 2001)

Os “doze mandamentos” do manifesto ágil dissertam de forma mais detalhada o que os quatro valores apresentam. Realçam principalmente a importância de uma equipe motivada e dedicada ao projeto e que um ambiente propício é essencial. Indivíduos unidos relacionam-se de forma positiva. Desta maneira, a comunicação torna-se mais clara e concisa, agilizando a conclusão dos objetivos. O cliente deve receber o produto que solicitou conforme seus requisitos, porém, estes podem ser modificados durante o projeto e isto não deve ser um empecilho, pelo contrário. Para os autores do manifesto ágil, simplicidade é eficiência. Vale lembrar que a utilização de uma metodologia ágil não significa enaltecer o caos, mas sim, ter este ambiente controlado (SBROCCO; MACEDO, 2012).

2.2.1.2 Scrum

O Scrum é a metodologia ágil que possui a maior disseminação dentro das organizações. Pode ser aplicada em diversos tipos de negócio, contudo, nas empresas de desenvolvimento de software é o seu destaque.

A ideia surgiu na década de 80 a partir dos japoneses Hirotaka Takeuchi e Ikujiro Nonaka, no artigo “*The new new product development game*”. O nome Scrum é baseado em uma jogada de Rugby, que não por acaso, é um esporte que depende totalmente de trabalho em equipe. A jogada Scrum tem o intuito de reunir os jogadores e combinar a próxima jogada. Assim como no gerenciamento de projetos, cabe ao gerente de projetos planejar de forma ágil o que deve ser feito e resolver os obstáculos encontrados da maneira mais veloz possível, com o apoio do seu grupo. Tempo depois, os autores Jeff Sutherland, Ken Schwaber e Mike Beedle documentaram as definições do Scrum, assim, tornando-a efetivamente uma metodologia ágil.

A aderência com o manifesto ágil embasa esta metodologia ágil. Promove a valorização do indivíduo, a simplicidade do trabalho, comunicação verbal e entregas incrementais do produto. O foco do Scrum está em ambientes com um alto grau de mudanças (PRIKLADNICKI; WILLI; MILANI, 2014). Ou seja, pode-se dizer que as mudanças são bem-vindas.

O Scrum pode ser caracterizado da seguinte forma, conforme as definições de PRIKLADNICKI, WILLI e MILANI (2014):

- Equipes multitarefas que se organizam para desenvolver incrementos em períodos curtos;
- O Scrum master deve garantir a aplicação correta do método e resolver os empecilhos, blindando a equipe;
- O backlog do produto deve ser revisado continuamente;

As definições destacam perfeitamente como a abordagem Scrum trabalha. Das características citadas, ao menos duas citam sobre o trabalho em equipe. A valorização do indivíduo é de suma importância na aplicação desta metodologia. Isto deve ser garantido pelo Scrum master (será explicado a seguir). Além disso, cada equipe possui certa liberdade de trabalho e a preocupação mútua é a entrega do que foi estipulado para aquele período (Sprint).

2.2.1.2.1 Papéis

Na metodologia Scrum, as pessoas envolvidas podem assumir três funções diferentes. Cada uma possui a sua importância dentro do projeto e o principal deve ser a atuação em conjunto entre todas. Isto é a chave para o sucesso de um trabalho administrado com esta abordagem ágil.

a) *Product Owner*

De acordo com a denominação, trata-se do “dono do produto”. COHN (2011) atribui ao dono do produto a responsabilidade de assegurar o andamento do projeto desenvolvido pela equipe para o caminho certo. Além disso, compromete-se a

fornecer os recursos e requisitos necessários para que o projeto seja construído (FOGGETTI, 2014).

O dono do produto é de suma importância para a execução do projeto. A sua aproximação com o Scrum master e a equipe definem durante e ao fim de cada Sprint se os requisitos que devem estar disponíveis no incremento estão de acordo. Vale lembrar também que o *Product Owner* não obrigatoriamente será um cliente externo. O Scrum pode ser aplicado para realizar um projeto para a própria empresa e o dono do produto ser o diretor ou o presidente, por exemplo.

COHN (2011) em seu estudo compartilha que o *Product Owner* deve fornecer a visão e os limites do projeto. Isso significa que o indivíduo que possui este papel tem a função de diagnosticar o ambiente a ser aplicado o trabalho e passar isto para a equipe. Isto atinge diretamente o backlog do produto e os sprints, devido às prioridades serem aplicadas de acordo com este contexto também. Além disso, os limites (ou restrições) do projeto carecem de ser esclarecidos pelo Product Owner também.

b) Scrum Master

O Scrum master é o responsável pela equipe. Responsabiliza-se pela aplicação correta do método, atuando como um gerente de projetos (FOGGHETTI, 2014). Orienta o *Product Owner* no que tange o desenvolvimento do backlog do produto, visando garantir a integridade da futura aplicação das técnicas Scrum (PRIKLADNICKI; WILLI; MILANI, 2014).

Persistindo com o estudo de SBROCCO e MACEDO (2012), os autores representam os papéis de Scrum master a partir de algumas características. Deve desempenhar o papel de líder e assumir as responsabilidades pela equipe, promover o engajamento, solucionar empecilhos identificados durante o Sprint e ajudar a desenvolver o *Product Backlog*. Além disso, filtrar as situações advindas externamente e garantir que o Scrum seja aplicado com louvor.

Os autores entram em acordo nas definições deste papel. O indivíduo deve ter conhecimento sobre aplicação do Scrum e o mais importante: saber trabalhar em equipe. Sua responsabilidade de treinar, engajar e motivar a equipe de maneira eficaz durante a execução do projeto torna-o a pessoa certa para a função.

c) Equipe

A equipe é constituída pelos colaboradores que devem executar as tarefas. A característica principal é a capacidade de multitarefa, ou seja, o time no desenvolvimento do projeto deve possuir diversas especialidades, não deve ser focado em apenas um tema (SBROCCO; MACEDO, 2012). Esta equipe deverá disponibilizar ao fim de cada interação um incremento do produto a ser entregue para o cliente em companhia da “Definição de Pronto” (PRIKLADNICKI; WILLI; MILANI, 2014). Esta definição de pronto trata-se das funcionalidades que devem ser entregues ao fim do incremento.

Uma equipe SCRUM possui a responsabilidade de realizar estimativas, detalhar as histórias e desmembrá-las em tarefas, desenvolver os incrementos do produto de maneira qualificada e apresentar os resultados ao cliente (SBROCCO; MACEDO, 2012).

2.2.1.2.2 Artefatos

A metodologia Scrum, assim como as outras abordagens ágeis, utiliza o mínimo de documentação possível. Porém, alguns artefatos obrigatoriamente devem ser gerados para que a aplicação seja feita de maneira efetiva. São eles:

a) Product Backlog

O backlog do produto recebe os requisitos definidos pelo Product Owner. Este artefato pode ser gerado a partir de diversas maneiras, porém, a forma mais conhecida baseia-se em histórias de usuário. Estas histórias são documentadas e cabe à equipe interpretá-las para transformá-las em produto. PRIKLADNICKI, WILLI e MILANI (2014) salientam que histórias não são o único método. Casos de uso e descrições textuais são bem-vindos, basta que estejam de acordo com o projeto. Não há uma maneira que é a ideal. Cabe à própria equipe identificar sua preferência e aplicá-la.

Além disso, o Product Backlog também recebe as prioridades de cada requisito para que seja designado corretamente nos sprints. Este artefato pode ser constantemente modificado, visto que o Scrum é aberto a mudanças e inclusive as prioridades podem mudar. O Scrum Master deve organizar os sprints conforme as mudanças no ambiente.

b) Sprint Backlog

O Backlog da Sprint tem o intuito de indicar à equipe o que deve ser produzido no Sprint, qual membro deve ser responsável por tal funcionalidade e a ordem de execução. A equipe deve respeitar esta ordenação para que atrasos não aconteçam. Como a ordem foi definida baseada na prioridade do cliente, caso a funcionalidade não seja entregue conforme o planejado, pode causar desconfiança ao cliente prejudicando o andamento de todo o projeto. O *taskboard* (que será detalhado a seguir) é o responsável pela representatividade destas tarefas, guiando os colaboradores para a execução correta. A Backlog do Sprint deve centralizar todo o trabalho a ser executado durante as sprints. Este documento embasa as sprints que serão executadas pelo desenvolvimento (PRIKLADNICKI; WILLI; MILANI, 2014).

c) Incremento

O incremento é o resultado gerado ao fim de cada sprint. As tarefas aplicadas durante o período transformam-se em produto. Este, deve ser apresentado ao cliente que deve validá-lo de acordo com o que foi apresentado no Sprint Backlog. Caso a

funcionalidade não esteja pronta neste sprint, deve retornar no próximo e sua prioridade deverá ser redefinida com as demais.

d) *Taskboard*

O *taskboard* centraliza todas as tarefas definidas a partir do Product Backlog. Além disso, este quadro possui identificações sobre as situações das tarefas e cabe à equipe alimentá-lo. Isso significa que caso a tarefa esteja incluída no sprint atual e seja finalizada, sua situação no *taskboard* deverá ser modificada para “Finalizada”, por exemplo. Estas situações podem ser definidas pelo Scrum master após conversa com a equipe, porém as sugestões apresentadas na definição do Scrum são *To Do* (a fazer), *Doing* (fazendo), *Done* (Feito) e *To Verify* (para verificar). SBROCCO e MACEDO (2012) explicam:

- *To Do* (a fazer): Tarefas definidas no sprint que ainda não foram iniciadas;
- *Doing* (fazendo): Tarefas que estão em andamento pela equipe;
- *Done* (feito): Tarefas concluídas;
- *To Verify* (para verificar): Tarefas que estão preparadas para serem executadas em ambiente de teste.

O fluxo deve ser executado na ordem citada acima. Uma tarefa “a fazer” nunca deve ser transferida diretamente para “feito”. O colaborador deve indicar à equipe que está em andamento, assim, evitam-se falhas na comunicação e dois integrantes executarem o mesmo procedimento. Todos devem estar de acordo com o processo para que não ocorram falhas indesejadas, portanto, cabe ao Scrum master instruir e engajar a equipe a serem os mais corretos possível neste quesito.

e) Gráfico Burndown

Este gráfico compromete-se a mostrar o desempenho atual da equipe comparado com o desempenho desejado. É composto por uma linha vertical que representa a quantidade de esforço a ser realizado e a horizontal que representa o tempo (SBROCCO; MACEDO, 2012). Ligando as extremidades apresenta-se uma linha vermelha. Caso o fluxo da equipe esteja abaixo da linha vermelha significa que o desempenho está acima do esperado. Outrora, acima da linha do fluxo ideal, a equipe está deixando a desejar e cabe ao Scrum master verificar o motivo da decadência no desempenho.

2.2.1.2.3 *Cerimônias e Sprint*

As cerimônias são encontros periódicos realizados pela equipe para esclarecimentos referente a um Sprint. O Sprint trata-se do período definido pelo Scrum master para entrega de um Incremento. Dentro deste período, as tarefas devem ser cadastradas a cada programador de acordo com os requisitos que carecem de ser atendidos no momento. A sprint conceitua-se como um tempo pré-determinado

em que o desenvolvimento das atividades é executado (SBROCCO; MACEDO, 2012). Os encontros são:

a) Revisão de planejamento de Sprint

Esta reunião define o que será produzido no próximo sprint. É ministrada pelo Scrum master após a definição do Product Backlog e prioridades de cada requisito. A cerimônia deverá responder às seguintes perguntas (FOGGETTI, 2014):

- O que será entregue no incremento desta sprint?
- Como faremos para entregar o incremento desta sprint?

Para o andamento do trabalho este encontro é relevante. Neste momento são definidos os fluxos de uma ou mais sprints. Nesta reunião deve ser definida uma meta de entrega ao fim do sprint pela equipe e como a sprint apresenta as partes a serem produzidas para cada requisito (PRIKLADNICKI; WILLI; MILANI, 2014).

b) Scrum Diário

A reunião Scrum diária tem o intuito de indicar a posição atual da equipe no sprint. Neste momento os colaboradores devem apresentar o que estão desenvolvendo no momento, o que mudou desde a última reunião e se há algum empecilho no seu trabalho (FOGGETTI, 2014). Este compromisso é ministrado novamente pelo Scrum master e sua duração obrigatoriamente deve ser curta. Cabe ao responsável gerenciar o tempo.

Recomenda-se que seja realizada ao lado do Taskboard para que o acompanhamento seja feito de forma mais eficiente (SBROCCO; MACEDO, 2012).

c) Revisão de Sprint

Esta cerimônia é realizada ao fim de cada período de sprint. Os interessados são a equipe do projeto, o Scrum master e o Product owner. Neste momento são apresentados os resultados que estão presentes no incremento. Além disso, devem ser levantados os problemas ocorridos durante o sprint e as soluções aplicadas (PRIKLADNICKI; WILLI; MILANI, 2014). Todas as perguntas devem ser respondidas.

Por ser uma reunião cujo propósito é expor o decorrente do tempo avaliado, o Scrum master e a equipe devem definir também o percentual de andamento e estimativa de conclusão. O Product Owner deve receber o incremento e avaliá-lo para verificar se está de acordo.

d) Retrospectiva do Sprint

Buscando a melhoria contínua do processo, a retrospectiva da sprint tem o objetivo de identificar possíveis aspectos a serem aprimorados para uma próxima sprint. Os itens levantados podem tratar de diversos assuntos, tais como o trabalho em equipe e ferramentas utilizadas, por exemplo.

O Scrum master e todos os membros da equipe devem participar deste encontro. O Product Owner pode ser convocado também, contudo não faz-se necessário. Cabe ao responsável reunir todos os pontos levantados, sejam bons ou ruins, priorizá-los adequadamente e tomar atitudes de mudança (SBROCCO; MACEDO, 2012).

2.2.1.3 Lean Software Development

O LSD (Lean Software Development) segue as premissas da Lean direcionada à produção de outros produtos. Respeita o conceito de produção enxuta e a redução de desperdício. Seus procedimentos promovem o engajamento da equipe e podem ser utilizados em qualquer organização (SBROCCO; MACEDO, 2012). A aplicação de uma cultura ágil é indicada para o sucesso do método.

Baseando-se no manifesto ágil e nos movimentos conseguintes, o LSD respeita sete princípios, conforme POPPENDIECK e POPPENDIECK (2013).

a) Evitar desperdício

Eliminar o desperdício significa tornar o projeto o mais minimalista possível. Ou seja, deve ser realizado apenas o que foi acordado com o cliente, sem maiores caprichos e de forma simples. Desta forma, mais garantida é a qualidade oferecida ao cliente e a manutenção do código, caso necessário, também é mais clara. Cada funcionalidade desenvolvida deve estar ligada a um requisito e nada deve estar em processo na entrega final. Além disso, POPPENDIECK e POPPENDIECK (2013) esclarecem também sobre alguns desperdícios capitais, como evitar a troca de responsável das tarefas e atrasos.

b) Garantir a qualidade

Um produto gerado de forma ágil não significa que perde a qualidade. Pelo contrário, a LSD por meio de sua entrega incremental característica das metodologias ágeis garante a qualidade nos testes executados e na apresentação das funcionalidades para o cliente ao fim de cada bloco de tarefas. Ademais, FOGGETTI (2014) salienta a importância de realizar corretamente a funcionalidade na primeira vez, ou seja, o código deve ser qualificado na primeira entrega.

c) Criar conhecimento

Cabe ao responsável pela equipe convencer a equipe a documentar o conhecimento aprendido. Na leitura destes registros, o indivíduo deve ser capaz de identificar o que houve em cada momento e tomar alguma medida necessária referente ao projeto, caso seja necessário.

d) Adiar comprometerimentos

Adiar comprometimentos consiste em permitir que as funcionalidades tenham alternativas de reversão. Caso seja necessário executar alguma mudança no que foi feito, o código deve permitir esta alteração com o mínimo de retrabalho possível (FOGGETTI, 2014). Neste momento o controle de tempo deve ser feito de forma inteligente, visto que adiar certos compromissos pode atrasar o projeto e assim, perder a confiabilidade do cliente.

e) Entregar rápido

A entrega dos incrementos e, por conseguinte, do produto final deve ser feita o mais rápido possível. POPPENDIECK e POPPENDIECK (2013) introduzem o conceito de tempo de ciclo, ou seja, quais são os prazos possíveis para entrega de cada parte do software? Dentro do LSD, incentiva-se a investigação sobre mitigação do processo na questão de tempo e custo.

f) Respeitar as pessoas

A premissa de respeitar as pessoas envolve o relacionamento entre todos da equipe. Logo, todos os colaboradores devem estar com o mesmo objetivo em mente e empenhados a alcançá-lo. Para que isto seja executado com êxito, é primordial que o líder desta equipe seja respeitado por todos e que gerencie a equipe de forma que todos sintam-se livres a opinar. O responsável também deve ter características propícias para o Lean, ou seja, prático e simples.

Além disso, deve-se levar em conta as habilidades técnicas de cada colaborador e usufruí-las da melhor maneira possível de acordo com as demandas do projeto (FOGGETTI, 2014).

g) Otimizar o todo

Assim como no Lean convencional, que originou o específico para software, a melhoria contínua também é uma premissa para esta ramificação. A equipe deve identificar as possibilidades de melhoria e aplicá-las, de modo que seu processo seja otimizado na medida do possível e por conseguinte, aprimorando a qualidade do produto.

Os sete conceitos acima são essenciais para que a LSD seja aplicada de forma eficiente e eficaz. Porém, é importante lembrar que, assim como as demais metodologias ágeis, o relacionamento e tratamento com as pessoas são chaves para o sucesso. Portanto, a mentalidade da equipe em prol do alcance dos objetivos baseado na estratégia é indispensável.

Além disso, cada organização possui particularidades em que no momento da análise para aplicação de um método de gerencia de projeto, devem ser levadas em conta. Uma metodologia ágil, de acordo com a sua equipe atual e seu negócio, pode não ser a melhor forma de gerenciamento. No entanto, uma abordagem tradicional pode trazer burocracias que são dispensáveis.

2.2.1.4 Extreme Programming

A metodologia de desenvolvimento de software eXtreme Programming, popularmente conhecida como XP, foi criada pelo engenheiro de software Kent Beck e aprimorada juntamente com Ward Cunningham, desenvolvedor na época. O método foi difundido em meados da década de 90. Contudo, sua especificação foi documentada em 2000 pelo mesmo Kent Beck em sua obra *Extreme Programming explained: embrace change*.

2.2.1.4.1 Princípios

A aplicação do XP consiste em quatro valores principais, conforme determinado por BECK (2000). Estão detalhados a seguir.

a) Comunicação

As falhas de comunicação acontecem em qualquer projeto e podem ser causas de custos e perda de tempo irreparáveis. Portanto, o XP trata a comunicação de modo que sua relevância ao projeto seja significativa. Ao levantar as falhas no projeto, a comunicação com lapsos em boa parte das ocasiões podem ser a causa de tais acontecimentos (BECK, 2000).

A comunicação valorizada pelo XP encoraja a prática tanto de maneira externa, ou seja, desenvolvedor e cliente, ou interna, a conversa entre membros da mesma equipe. Durante o uso da metodologia a comunicação pode trazer itens a serem alterados no processo. Caso esta não aconteça, essas mudanças nunca ocorrerão (AUER; MILLER, 2002).

b) Simplicidade

O *Extreme Programming* busca simplificar o que for possível. De acordo com a metodologia, a simplicidade pode reduzir custos que uma abordagem mais detalhada poderia trazer com funcionalidades que poderiam ser descartadas (KENT, 2000). Além disso, reduz a chance de ocorrência de imperfeições e minimiza a execução de testes, outra prática reforçada pelo XP.

c) Feedback

O retorno sobre o estado do projeto e as funcionalidades em específico é um valor de suma importância para o XP. Além disso, o feedback não aplica-se somente ao tratamento cliente e desenvolvedor, mas sim em todas as maneiras possíveis de retorno. Um feedback de um teste automatizado para o programador deve ser feito da forma mais ágil possível para economizar tempo. As histórias criadas pelo cliente

devem estar concisas e legíveis para proporcionar o desenvolvimento, logo, cabe ao desenvolvedor comunicar ao cliente sobre a situação destes documentos (BECK, 2000).

Quanto menor o tempo de feedback, mais eficiente e eficaz torna-se o projeto. Assim, caso ocorra alguma implementação que foi feita de maneira errônea, a correção pode ser feita à tempo (PRIKLADNICKI; WILLI; MILANI, 2014).

d) Coragem

As atitudes de coragem tratam-se de não refutar perante uma situação ruim. Não se deve desistir quando as circunstâncias não são favoráveis (AUER; MILLER, 2002). Ao aplicar a metodologia XP, o indivíduo não deve demonstrar sinais de fraqueza ou indecisão que levem muito tempo. Ou seja, caso seja necessário tomar uma atitude drástica (excluir todo o código, por exemplo), faça. Uma ideia inovadora surgiu, a equipe concorda que pode render frutos, porém leva tempo ou não há vontade. Isto não pode acontecer.

Os quatro princípios juntos aplicados na metodologia XP são a base para o sucesso neste ambiente. Todos necessitam um do outro para que sejam usados efetivamente. A comunicação e feedback juntos, por exemplo, atinge diretamente o cliente no momento que está recebendo a situação das tarefas de forma ágil e clara. Alguns autores citam o Respeito como um princípio considerado pela *Extreme Programming* também. Este ocorre no momento que a equipe consegue aplicar corretamente o método, construindo a união entre os colaboradores e valorização desta metodologia (PRESSMAN, 2011).

2.2.1.4.2 Práticas

A Extreme Programming, além de seus valores que praticam uma forma de desenvolvimento e planejamento diversificada perante uma metodologia tradicional, apresenta dez princípios que a embasam (SOMMERVILLE, 2011). São eles:

a) Planejamento Incremental

O XP, assim como no Scrum, sugere que os requisitos de usuário sejam definidos por meio de histórias escritas pelos mesmos. Nesta situação, são entregues cartões CRC (classe, responsabilidade e colaboração) que recebem estas informações. Cabe ao analista, desenvolvedor e cliente assumirem o entendimento de cada funcionalidade a ser concebida no sistema. Estas determinações devem ser priorizadas corretamente. A partir destas condições, deve ser entregues incrementos que comportam partes dos requisitos apresentados. Um conjunto de incrementos compõem um sistema. A comunicação, um dos valores da XP, é fundamental para o sucesso da aplicação desta prática. Desenvolvedores e clientes devem entender-se,

caso contrário, não é possível estimar e realizar o projeto (ASTELS; MILLER; NOVAK, 2002).

b) Versões pequenas e frequentes

A geração de releases deve ser contínua e conter as tarefas que tornam o incremento gerado adequado (SOMMERVILLE, 2011). O cliente deve aprovar cada versão gerada em prol de diagnosticar se está válida ou se algo deve retornar para o desenvolvimento. É possível o cliente determinar o valor do software o quanto antes possível, não deixando-o desinformado (AUER; MILLER, 2002).

c) Simplificar ao máximo

O objetivo do projeto é atender a demanda do cliente. Não devem ser realizados caprichos ou outras funcionalidades que não foram definidas. Cada tarefa desenvolvida deve ter embasamento, caso contrário, são testes desnecessários a serem feitos e possibilidade de erros de codificação (BECK, 2000).

d) Testes antes do desenvolvimento

Um dos elementos mais importantes da metodologia XP, os testes automatizados antes do início de qualquer desenvolvimento garantem qualidade ao software e segurança ao programador. O principal empecilho encontrado neste momento é o convencimento do indivíduo a aplicar esta forma de desenvolvimento, visto que grande parte dos programadores julgam a execução de testes como desnecessária. Esta mentalidade não é acolhida pelo eXtreme programming.

Os testes dentro da metodologia XP podem ser classificados de duas maneiras, segundo AUER e MILLER (2002): unitário e de aceitação. A automação de testes deve ser aplicada ao primeiro, respectivamente. Os programadores devem compor seu código de forma que ao passar pela bateria de testes automatizados, seu código seja efetivamente executado. Assim, seu sistema proporciona segurança, estabilidade na codificação e que as alterações atuais não irão atingir o que já foi realizado anteriormente (ASTELS; MILLER; NOVAK, 2002). Os testes de aceitação são aplicados pelas partes interessadas e tem o propósito de validar se as funcionalidades desenvolvidas estão em pleno acordo com os requisitos.

e) Otimizar o código

O código deve ser refatorado sempre que possível. Caso o desenvolvedor identifique esta possibilidade, deve efetuar este aprimoramento imediatamente. BECK (2000) cita que mesmo tornando o trabalho mais árduo, isto reflete nas funcionalidades que venham a ser desenvolvidas no futuro. Os novos desenvolvimentos presentes em um ambiente otimizado tendem a executar com a chance de falha reduzida. Portanto, o XP além da segurança proporcionada nos testes, promove a otimização do código como uma prática a ser aplicada sem retorquir, garantindo assim um sistema ainda mais seguro.

f) Programação em Pares

A comunicação está presente nesta prática, instigando a conversa entre dois desenvolvedores de uma mesma funcionalidade. O ditado “duas cabeças pensam melhor do que uma” resume a ideologia promovida por esta prática. Devido ao custo para a organização de manter dois indivíduos trabalhando em uma mesma funcionalidade, esta técnica seja a menos aplicada dentre as demais. Os autores defendem arduamente sua realização. Os principais argumentos são a revisão em tempo real, identificação e resposta aos impasses de forma ágil (PRESSMAN, 2011).

g) Qualquer programador pode prestar manutenção em qualquer módulo;

Em contraposição às organizações que designam e formam especialistas em partes específicas do seu produto, o XP indica que toda a equipe deve conhecer e prestar manutenção em todo o software. Todos são responsáveis pelo código. O intuito da metodologia nesta prática é evitar a atribuição de culpa a um indivíduo e a dependência com o mesmo para a resolução de um problema em específico (AUER; MILLER, 2002).

h) Integrar tarefa após finalizar

A integração realizada pelo programador ao término de uma tarefa permite agilidade na execução dos testes e detecção de possíveis erros. ASTELS, MILLER e NOVAK (2002) citam que em certos ambientes a integração é um dos piores momentos da equipe no processo devido aos problemas encontrados. Contudo, defendem que caso a integração fosse feita continuamente, estes erros seriam resolvidos de forma mais tranquila e evitando os conflitos que acontecem durante este período.

i) Não realizar trabalho extra

A metodologia XP defende que o trabalho deve ser realizado durante as horas contratadas de trabalho. Não há hora extra. A estimativa de tempo deve ser baseada no que cada indivíduo pode trabalhar. Colaboradores sem condições físicas e mentais de trabalho não executam suas tarefas da melhor forma possível e sua probabilidade de erro aumenta (AUER; MILLER, 2002).

j) Disponibilidade do cliente

O contato contínuo com o cliente é essencial para a execução com êxito do trabalho usando a XP. O cliente faz parte da equipe, responsabiliza-se pelos requisitos e pela aceitação dos mesmos (SOMMERVILLE, 2011). Além disso, as premissas de respeito e comunicação devem ser aplicadas de forma ainda mais intensa, visto que um bom relacionamento com o cliente é ideal em qualquer projeto.

A metodologia eXtreme Programming, quando respeitados os princípios e as práticas, pode ser um processo de sucesso no seu ambiente. Contudo, devem ser

examinadas as circunstâncias do seu contexto atual e se este modelo cabe para a organização.

2.3 TRADICIONAL X ADAPTATIVO

Com o passar dos anos e as novas demandas de clientes à procura de software, metodologias ágeis começaram a ganhar espaço dentre os gerentes de projetos. Sua proposta de rapidez nas respostas e tratamento das mudanças tornou-se o seu diferencial diante das técnicas tradicionais.

As tendências ágeis estão à contrapartida dos métodos tradicionais (ou prescritivos), visto que estas abordagens buscam veemente o planejamento do projeto e produto como um todo desde o início. As metodologias ágeis defendem o processo incremental, ou seja, as entregas são feitas em blocos de modo que as partes interessadas participem de forma mais ativa no processo. MARTINS (2007) justifica que decisões podem ser tomadas e que atingem o projeto como um todo. Em abordagens tradicionais, seria necessário refazer todo o planejamento, enquanto que em um pensamento ágil o nível de impacto seria reduzido.

Além disso, a gerência de projetos tradicional pode tornar-se burocrática e em alguns casos certos controles são desnecessários. LARSON (2016) destaca que os métodos adaptativos estão preparados para atuar em um ambiente com mudanças, imprevisíveis, enquanto que metodologias convencionais são aplicadas em cenários sem transformações. Neste quesito, a abordagem ágil se destaca devido à imprevisibilidade ser natural em um ambiente natural. O PMBOK® (2011) ressalta que métodos adaptativos devem ser aplicados em ambientes frequentemente mutáveis e com dificuldades no levantamento de requisitos, assim sendo, sem definição de um escopo completo.

Alguns temas são divergentes entre ambas as metodologias. Amaral et al. (2011) destacam alguns que estão apresentados a seguir.

a) Autogestão

O conceito de autogestão promove a autonomia do indivíduo em tomar as decisões e realizar o que julga correto. Há um responsável pela equipe, porém, todas as decisões não são tomadas por este indivíduo. Normalmente em equipes ágeis os colaboradores são especialistas e esta proatividade é apreciada durante o processo (AMARAL et al., 2011). No entanto, os métodos tradicionais preocupam-se com a documentação e que as decisões devem ser tomadas por um responsável. Em certo ponto a segurança na execução das atividades é maior, porém o processo é mais lento e burocrático.

A valorização do indivíduo perante a equipe e a organização também distingue a tradicional da ágil. Em uma abordagem prescritiva, em que a documentação é

essencial, as pessoas são dispensáveis pois teoricamente as informações e conhecimento estariam registrados. Nas adaptativas a documentação é a mais enxuta possível e o indivíduo é valorizado acima de qualquer burocracia, ou seja, o relacionamento entre as pessoas é mais importante que um registro.

b) Incremento e simplicidade

Além do tratamento com a equipe, uma grande discrepância entre ambos os métodos são as entregas do projeto. Na abordagem tradicional, defende-se a entrega de um fim (produto, serviço ou processo) após sua produção por completo. Nas ágeis, são entregues partes pequenas e em tempos pré-definidos.

Uma metodologia tradicional bem sucedida sofre poucas mudanças durante o andamento do projeto e há menos comunicação com as partes interessadas se comparado com um método adaptativo. Vale lembrar que os incrementos devem estar de acordo com o contexto que o cliente realizou a solicitação, portanto, nada deve ser desenvolvido sem a justificativa de que deve estar disponibilizado no produto (SBROCCO; MACEDO, 2012).

A qualidade do produto representa o ponto que ambas as metodologias convergem. Em ambas um produto é qualificado no momento que atende os requisitos das partes interessadas. Nos métodos adaptativos, isto deve ser feito da maneira mais simples possível e com o mínimo de documentação necessária. Nas tradicionais, o desenvolvimento de toda a fundamentação implica em tornar o processo e as entregas mais lentas.

c) Comunicação com o cliente

A interação com o cliente é realizada de forma mais ativa nas metodologias ágeis. Devido a obrigatoriedade de estar presente na aprovação dos resultados de cada sprint, as partes interessadas também podem indicar a direção que o projeto tende a seguir (SBROCCO; MACEDO, 2012). Nas abordagens tradicionais, o projeto cumpre o planejado do início ao fim. O cliente participa ativamente na entrega final e pode ser consultado eventualmente.

Entretanto, o cliente não deve atribuir uma comunicação excessiva a ausência de planejamento ou falta de qualidade técnica. O cliente deve ter conhecimento sobre o funcionamento da abordagem ágil e entender a importância do relacionamento ativo entre ambas as partes.

Não existe uma metodologia perfeita. Cabe ao gerente de projeto analisar o seu contexto organizacional e o projeto recebido.

3 GAMIFICAÇÃO

A gamificação (ou ludificação) é um tema que tende a ganhar cada vez mais força com o passar dos anos. Hoje, em meio à tecnologia ao nosso redor, jogar tornou-se ainda mais fácil e competitivo. Este assunto resume-se à ideia de aplicar o conteúdo de uma atividade gamificada para resolver problemas usuais, do cotidiano.

O conceito origina-se da palavra *game* (jogos, partida, em inglês) e lúdico (refere-se a jogos, brinquedos) de acordo com o dicionário HOUAISS (2009). Segundo VIANNA et al. (2013), o termo foi abordado pela primeira vez a partir do programador Nick Pelling, contudo, somente tempo depois este conceito realmente foi trazido à tona. Os jogos fazem parte de toda vida, desde crianças até a melhor idade. BURKE (2015) salienta que a gamificação atinge principalmente os jovens adultos nascidos entre o início da década de 80 e fim da década de 90. As pessoas que nasceram neste período cresceram em paralelo à evolução dos games e a facilidade de acesso aos mesmos, portanto, possivelmente já viveram uma experiência semelhante ao proposto na ludificação. O elemento da competitividade, resultados, metas, solução de problemas e trabalho em grupo, pontos cruciais de um jogo, são o que tornam esta temática fascinante.

A gamificação concentra-se principalmente no fator motivacional, inclusive, que é o gatilho para o uso deste conceito neste estudo. ALVES (2014) relembra o fato de que muitos de nós já tiveram a experiência de passar horas durante um jogo sem parar. Em um ambiente empresarial, busca-se implantar este mesmo conceito em busca de resultados. Superficialmente, a ideia de atingir uma meta para receber uma recompensa aplicaria-se a um contexto de acordo com a estratégia da organização. Consequentemente, para atingir o sucesso, o engajamento dos colaboradores é fundamental. Contudo, BURKE (2015) ressalta que a gamificação não é retribuir o funcionário, mas sim motivá-lo a conquistar a recompensa de maneira gamificada.

É importante lembrar que gamificação não está ligada diretamente a *videogames*. Jogos digitais podem ser utilizados como elementos para um fim específico, todavia, este conceito abrange todo tipo de método que envolva o conceito de jogos para alcançar um objetivo.

3.1 JOGOS

Conforme dito anteriormente, os jogos estão cada vez mais presentes no nosso cotidiano. Com a comodidade do smartphone, a qualquer momento é possível competir e distrair-se com um aplicativo. HUIZINGA (1999) destaca que a brincadeira é natural ao ser humano e até mesmo aos animais, no momento que há uma atividade com regras (mesmo que sejam implícitas). O autor explica também que jogo se

caracteriza como uma atividade que imersa seus participantes a uma nova realidade, diferente do ambiente rotineiro.

Geralmente estão presentes nos jogos três atributos principais: *points* (pontos), *badges* (comprovação de conquistas, níveis) e *leaderboards* (classificações).

Pontos são requisitos principais de qualquer jogo, independentemente do seu tema. São utilizados para definir a sua posição atual dentro do ambiente, diagnosticar suas recompensas, dentre outras finalidades. Em suma, demonstra o seu desempenho durante o *game*. ZICHERMANN e CUNNINGHAM (2011) classificam o sistema de pontuação em cinco tipos: experiência, resgatáveis, de técnica, karma e reputação.

- a) Pontos de experiência: são utilizados para demonstrar a sua evolução dentro do ambiente virtual. Normalmente estão ligados a um ranking, que realiza a comparação entre os jogadores a fim de determinar o jogador com o nível mais alto.
- b) Pontos resgatáveis: podem ser obtidos a partir de uma moeda virtual do *game*. Estas moedas são atreladas ao jogador por meio de atividades cumpridas dentro do ambiente ou com dinheiro real.
- c) Pontos de técnica: estão atrelados a uma ação específica dentro do jogo, por exemplo, um golpe que pode evoluir conforme os treinos realizados pelo personagem.
- d) *Karma Points*: raramente são utilizados nos jogos. Consistem em uma recompensa por iniciarem a atividade gamificada, ou seja, você ganha pontos por apenas abrir o jogo, além do seu desempenho.
- e) Pontos de reputação: consistem em meios de avaliar o jogador em todas as atividades que realiza, a fim de diagnosticar se o indivíduo é confiável.

Badges são conquistas realizadas dentro do jogo e podem gerar uma recompensa ao jogador. Normalmente o jogo tem a intenção de engajar o jogador a alcançar estes objetivos. Esta característica também valoriza o ato de colecionar e atingir metas (KUMAR; HERGER, 2013).

As classificações, ou *leaderboards*, tem o propósito de comparar os jogadores e exibir quem possui a maior pontuação, quem realizou mais conquistas do *game*, dentre outras formas de comparação. ZICHERMANN e CUNNINGHAM (2011) ressaltam que em grande parte dos casos os jogadores não se preocupam com o que realmente compõe a classificação. Contudo, isto pode variar de acordo com o perfil do jogador.

Um jogo pode ser classificado em diversas categorias. Por faixa etária, gênero e plataforma são alguns exemplos. Contudo, todos eles compartilham de quatro itens que são base para qualquer atividade lúdica, conforme VIANNA et al. (2013): meta, regras, sistema de feedback e participação voluntária. A meta é o desafio, missão que o usuário deve cumprir. Ao concluir com êxito, o jogador pode receber uma recompensa tangível ou elevar de nível. No ambiente empresarial este termo pode ser

aplicado da mesma forma, ao receber um vale para jantar com a família ou com a sua equipe de trabalho (Introduction to Gamification, 2014). Além disso, o atingimento da meta pode proporcionar ao colaborador atingir um nível superior, ou seja, receber uma promoção. As regras são necessárias para que o atingimento das metas seja feito garantindo o mesmo ambiente para todos sem discriminação, tornando a conquista legítima. A ausência de regras provoca instabilidade e não caracteriza um ambiente gamificado (ARRUDA, 2014). Além disso, o jogador deve ter conhecimento do seu desempenho a partir de um sistema de feedback, ou seja, o atingimento da meta não deve ser uma surpresa. VIANNA et al. (2013) lembram que esta também é uma poderosa ferramenta de motivação, visto que o acompanhamento do desempenho pode engajar o colaborador a atingir seus objetivos o quanto antes possível. Por último, a participação voluntária exige o consenso da equipe perante as regras, o sistema de feedback e as metas a serem atingidas.

Os jogos hoje são utilizados para diversos fins. Uma ferramenta de interação lúdica na área da educação seria útil para ensinar algum conteúdo que tornaria-se maçante em classe. O aplicativo *Duolingo*, que oferece o aprendizado de diversos idiomas, é um bom exemplo desta ocasião. Além disso, um jogo pode ser uma ótima escolha caso o intuito seja a divulgação de um evento. Os resultados no aplicativo podem lhe trazer benefícios dentro da festa.

3.2 ANÁLISE DE PERFIS

Durante uma atividade gamificada, seja uma partida de futebol ou um videogame, expressa-se a personalidade do indivíduo com as atitudes e escolhas no momento. Outrora, o mesmo pode simular um personagem na tentativa de ludibriar seus adversários a seu favor. Em uma partida de poker, por exemplo, as partidas são decididas no momento que um dos jogadores manifesta-se de forma questionável ao fazer um blefe. Retirar-se na cadeira é sinal de insegurança e caso seja detectado por um adversário, é informação que pode decidir o duelo. Isto está contemplado na análise de perfis. BARTLE (1996) em sua obra caracteriza os jogadores em quatro tipos distintos: *killers* (predadores), *achievers* (realizadores), *explorers* (exploradores) e *socializers* (socializadores). Os perfis estão detalhados após a Figura 5.

a) Predadores

Indivíduos com a personalidade caracterizada como predador são competitivos ao extremo. Desconsideram o ambiente do jogo e a “convivência” em prol do cumprimento dos seus objetivos e satisfação pessoal. VIANNA et al. (2013) destacam os métodos adotados por estes tipos de jogadores. Muitas vezes, procuram intimidar os adversários de modo que sejam afetados psicologicamente ou tecnicamente durante o jogo. Além disso, estes jogadores alegram-se ao identificar a derrota alheia (KUMAR; HERGER, 2013). Ambos autores ressaltam que apenas próximo a 1% dos jogadores possuem este perfil.

b) Realizadores

Os realizadores, assim como os predadores, possuem apreço pela vitória. Contudo, a maneira radical de alcançar as metas não é de natureza deste perfil. Sua busca pelas conquistas são feitas de maneira leal. Em qualquer estilo de jogo, almejam o topo. Caso esta meta não seja cumprida, o *achiever* desinteressa-se pelo jogo (ZICHERMANN; CUNNINGHAM, 2011). Atender esta fatia de público é um desafio para qualquer desenvolvedora de atividade lúdica, visto que o fascínio pelo jogo pode ser perdido facilmente. Além disso, obrigatoriamente deve apresentar elementos de competitividade entre jogadores. De acordo com VIANNA et al. (2013), indivíduos enquadrados nesta personalidade consistem em cerca de 10% do público.

c) Exploradores

Este jogador busca conhecer todos os aspectos do jogo nos mínimos detalhes. É de suma importância investigar o ambiente apresentado em busca de elementos novos que possam aprimorar a jogabilidade. Para o explorador, a experiência da vivência durante o jogo é mais importante que a vitória (ZICHERMANN; CUNNINGHAM, 2011). O destaque que procura receber dos adversários é o reconhecimento sobre o seu conhecimento da atividade gamificada e não necessariamente a vitória. Este perfil corresponde a aproximadamente 10% do público (VIANNA et al.,2013).

d) Socializadores

Os *socializers* não buscam a vitória acima de tudo ou a experiência oferecida pelo jogo. Neste perfil de jogador, as interações sociais são o principal atrativo e o que importa para o indivíduo. Jogos com mecânicas e modos de jogo em grupo são os preferidos e não necessariamente precisam promover algum tipo de competição. Conforme VIANNA et al. (2013), representam cerca de 80% do público.

Figura 5: Perfis definidos por Richard Bartle



Fonte: adaptado de KUMAR; HERGER, 2013, p. 44

Na análise destes tipos de perfis é possível diagnosticar a ligação que há entre a gamificação e o ambiente empresarial. Ao replicar as características de cada personalidade para o dia a dia de trabalho, deve-se tomar cuidado principalmente com os predadores. De acordo com o perfil os jogadores apresentam resultados, mas nem sempre de maneira legítima. Isto pode causar problemas para a equipe no momento em que a trapaça utilizada traga algum malefício geral. Por exemplo, em uma equipe que responde ao seu responsável baseado no desempenho por horas registradas. O predador é o membro da equipe com a média de horas por dia mais alta, porém, o registro foi feito de forma ética?

Os indivíduos socializadores tendem a ter um melhor relacionamento com as partes interessadas, seja o cliente ou a empresa. Contudo, ter uma boa relação com todos não deve ser confundido com praticar este contato na maior parte do tempo. Ou seja, o socializador deve ter consciência do seu trabalho e que possui metas a cumprir. Sua sede por atingir o objetivo não é a mesma de um colaborador realizador ou predador, todavia, isto não lhe deixa sem responsabilidades. Este perfil caracteriza-se também por manter um ambiente saudável na equipe. Como preza pela comunicação e vivências sociais, os conflitos não são bem-vindos. Não se deve confundir conflito com discussão.

Empregando um universo lúdico, a personalidade realizadora é a mais condizente com a liderança da equipe. Seu empenho está dirigido aos objetivos e não abstém de relações sociais para que as metas sejam atingidas. Vale lembrar o quão penoso é o fracasso para este perfil. Portanto, uma equipe bem-sucedida deve conter indivíduos de diversas individualidades, tornando cada ser único. O explorador pode identificar uma vertente a ser seguida que não foi detectada pelo realizador. O socializador, no momento que está alinhado com a equipe, auxilia no engajamento da equipe na retomada do rumo. O predador deve ser controlado, porém compartilhar sua sede de vitória com a equipe é uma ótima ideia. Em síntese, não há uma personalidade perfeita, mas sim uma união que traz frutos baseado em um processo que esteja condizente com a gamificação.

3.3 MOTIVAÇÃO

Para os gestores contemporâneos, manter seus funcionários motivados é a chave para o sucesso das equipes, conseqüentemente, alcançar os objetivos da estratégia. Contudo, esta não é uma tarefa fácil. Com o passar dos anos e a mudança de gerações, conquistar o colaborador torna-se cada vez mais árduo. As empresas que conseguem engajar seus funcionários tendem a ser mais competitivas e sobretudo, visionada pelas pessoas. Ao passo que a organização se prova um local atraente para se trabalhar, conseqüentemente a disputa pelo espaço pode ser também um fator motivacional.

A motivação, segundo MARINS e MUSSAK (2013), são as conclusões individuais que levam o ser a fazer suas escolhas. Portanto, um indivíduo motivado possui a vontade de tomar estas decisões e alcançar um objetivo. Os motivos em grande parte dos casos advêm de necessidades. (SILVA; RODRIGUES, 2007). A organização pode ser a criadora destes desejos, no momento que é capaz de alinhar seu pensamento para o futuro com as expectativas do funcionário. Ambos devem crescer juntos e este pensamento também pode motivar o indivíduo.

Este conceito é detalhado em duas vertentes, conforme VIANNA et al. (2013): intrínseca e extrínseca. Resumidamente, segundo o autor, a motivação intrínseca baseia-se na curiosidade, vontade de aprender coisas novas, a novidade. Isto é a motivação do indivíduo, sua recompensa é o alcance do objetivo. Na extrínseca, KUMAR e HELGER (2013) evidenciam os exemplos que provocam esta motivação de acordo com fatores externos, por exemplo bens materiais e sentimento de culpa ou derrota, provocados por alguma responsabilidade que não foi alcançada.

Em gamificação nas organizações, almeja-se a motivação intrínseca nos membros da equipe. BURKE (2015) comprova salientando que as recompensas a nível emocional (intrínsecas) são mais eficientes que a nível transacional (extrínsecas). Entende-se por transação algo que possui um período, ou seja, a satisfação é temporária. O sentimento de prazer ao cumprir um objetivo próprio ao invés de visar apenas a recompensa é um desafio para qualquer gestor de projetos ou de RH (recursos humanos). Em seus estudos, LAZZARO (2004) destaca os quatro principais “sentimentos” (ou tipos de divertimento) que incentivam o usuário a jogar. São eles: *Hard Fun* (jogar utilizando a estratégia, desafiar-se), *Easy Fun* (explorar o jogo e aventurar-se), *Altered States* (sensação de alívio, sentir-se bem consigo mesmo) e *The People Factor* (socializar com os jogadores). Todo indivíduo possui um certo perfil que pode se encaixar de uma melhor forma dentro da ludificação.

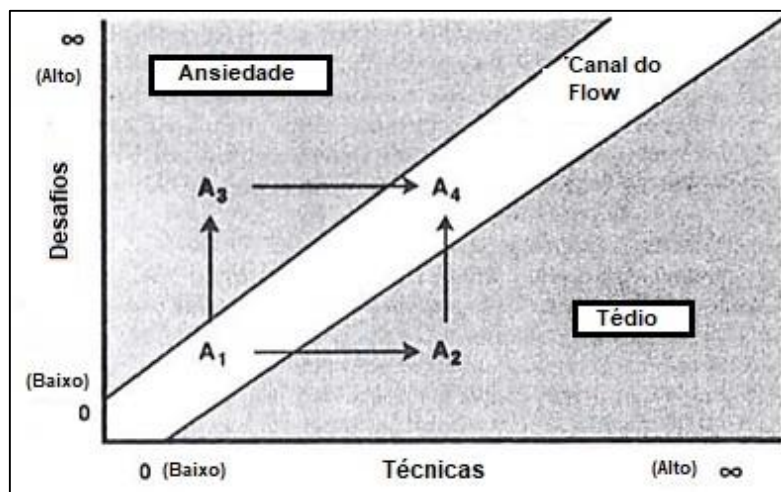
Dentro da concepção de motivação, BURKE (2015) classifica em três elementos. São eles: autonomia, domínio e propósito. A autonomia caracteriza-se pela permissão ao jogador de tomar as próprias decisões para alcançar as suas metas. Como diz o nome, domínio trata-se de ter o controle de algo que nos é relevante. Sentir-se motivado por executar uma atividade cujo controle total está na mão do jogador. Além disso, o propósito conceitua-se em agir por uma vontade “acima de si mesmo”. Isto significa que o objetivo a ser alcançado é mais relevante do que o próprio indivíduo.

A teoria do Flow também é relevante ao estudarmos a ludificação e a motivação. Seu idealizador foi Csikszentmihaly (1990) e trata basicamente da relação entre *anxiety* (ansiedade) e *boredom* (tédio). A Figura 6 demonstra este relacionamento.

Em suma, a teoria descreve que as tarefas que tenham um nível muito alto tendem a causar uma maior ansiedade no indivíduo caso seu conhecimento de como resolver a mesma seja baixo. Em contra-partida, no momento que o indivíduo tem

estudo e experiência para resolver os problemas, ele necessita de desafios para manter-se motivado e não executar apenas tarefas com nível baixo, portanto, pouca ansiedade. Quando há um meio termo entre as sensações, a pessoa está no ambiente desejado.

Figura 6: Diagrama flow



Fonte: adaptado de CSIKSZENTMIHALY, 1990, p. 74

Os jogos são atividades que estão presentes no cotidiano a tempos. Contudo, a gamificação é um conceito atual pois está alinhado com as personalidades das próximas gerações, que possuem maior contato e facilidade com tecnologia, atividades gamificadas e por fim tornam-se mais competitivas. Portanto, a motivação é o elemento principal a ser almejado ao aplicar um processo lúdico em uma equipe, neste caso, o laboratório.

4 TRABALHOS RELACIONADOS

Diversos trabalhos já foram publicados e que possuem relação com esta pesquisa. O cuidado das empresas em como controlar os seus projetos é o que pode garantir o cumprimento dos prazos e evitar surpresas no gerenciamento de custos. Neste caso, mesmo na utilização de uma metodologia ágil estas premissas devem ser respeitadas. Vários trabalhos contemplam análises, comparações e junções das abordagens prescritivas com as ágeis.

Na pesquisa de CONFORTO (2009), o autor busca apresentar uma solução para gestão de escopo e tempo a partir de um processo desenvolvido pelo próprio. O fluxo elaborado baseia-se em GAP (gerenciamento ágil de projetos). Por fim, realiza a análise para diagnosticar se o processo proposto foi efetivo.

A integração entre os processos do PMBOK® e as técnicas do Scrum são tema de pesquisa de outros autores. KARDEC (2012) realizou esta pesquisa comparando a metodologia tradicional com a adaptativa e mostrando como cada área de processo do PMBOK® é abordada dentro do Scrum. Por fim, são comparadas as entregas finais dos dois métodos.

Levando em conta o caso de KARDEC (2012), BOEING e SOUZA (2013) estudaram um método semelhante de integração entre ambas as metodologias, conforme o mesmo autor que embasa a proposta de solução deste trabalho. Os autores identificaram que é possível aplicar em paralelo os conceitos destas duas sistemáticas sem que percam suas características de agilidade e controle. Além disso, VARGAS (2016) também estudou o uso do Scrum simultaneamente com uma metodologia tradicional. A autora citou além do PMBOK® a integração com o método PRINCE2, utilizado na Europa. Lembrando que ambas as publicações focam no desenvolvimento ágil de software, mesmo que proponham a associação com métodos não ágeis.

Em relação à aplicação da agilidade proposta no Scrum, CARVALHO (2009) utilizou-se das técnicas em uma organização desenvolvedora de software motivado por, segundo o autor, a produção de software nesta empresa ser totalmente desorganizada. Ao término do processo, uma avaliação deve ser realizada para diagnosticar os resultados da mudança.

CRISTAL, WILDT e PRIKLADNICKI (2008) salientam também a motivação de utilizar o Scrum em virtude de evitar documentações que julgam desnecessárias, burocráticas. A equipe reitera que os registros realizados seguidamente acabam tornando-se maçantes.

AUDY (2015) disserta sobre a resposta das equipes ao modificar sua forma de trabalhar para o método Scrum, semelhante ao que é proposto neste trabalho. O autor destaca também a necessidade da análise do grupo no que tange a capacidade de auto-gestão.

O foco deste trabalho destina-se também à integração com técnicas de gamificação com o processo unificado entre Scrum e PMBOK®. LORIGGIO, FARIAS e MUSTARO (2013) em sua publicação destacaram a aplicação da ludificação como motivação dos indivíduos de uma equipe. Os elementos presentes em jogos buscam tornar a aprendizagem e aplicação do Scrum mais atraente, promovendo o engajamento da equipe. McCLEAN (2015) durante seu estudo reforça esta teoria e no seu caso, as técnicas gamificadas devem ser iniciadas em uma equipe. Mediante o sucesso, será disseminado para as demais.

A aplicação da gamificação em ambiente empresarial é uma abordagem relativamente nova. HAMARI e KOIVISTO (2015) destacam os motivos pelos quais este tipo de serviço pode ser utilizado. Justificam que a ludificação promove motivações intrínsecas que provocam o indivíduo a participar da aplicação do método. Acima de qualquer recompensa oferecida neste tipo de método, a gamificação procura valorizar o fator pessoal, seja na relação com a equipe como a valorização do próprio indivíduo em engajá-lo a alcançar suas metas.

Estas pesquisas estão relacionadas com o conteúdo abordado no referencial teórico e na proposta de solução deste trabalho. Cada trabalho, de certa forma, contribuiu para a continuação desta monografia.

5 PROPOSTA DE SOLUÇÃO

O processo apresentado deve ser aplicado ao laboratório de criação e aplicação de software, presente dentro da Universidade de Caxias do Sul. A seguir, descreve-se a situação atual do laboratório e detalhes do processo atual. Além disso, de forma detalhada está apresentada a proposta de solução para os problemas apresentados.

A análise proposta baseia-se na obra de Fábio Cruz denominada Scrum e PMBOK®: unidos no gerenciamento de projetos. O livro propõe a integração entre a metodologia tradicional e a adaptativa, de modo que ambas poderiam trabalhar juntas em uma organização. Em certos casos é necessário que sejam feitas entregas rápidas, contudo, alguns controles descartados pelo Scrum são resgatados dentro deste processo de Cruz. Inclusive, o autor cita que esta metodologia ágil não é complexa como os processos do PMBOK®, porém, está bem elucidado de acordo com as técnicas que propõe (CRUZ, 2013).

Este trabalho não propõe a aplicação do método do autor em sua totalidade, mas sim estudar e adequar sua proposta de acordo com o cenário em que este será praticado. Muitas soluções propostas por Cruz, devido ao reduzido número de pessoas e algumas restrições de tempo (toda a equipe não está no laboratório em tempo integral), não serão executadas.

Além disso, para promover a motivação da equipe no uso do processo e engajar novos alunos a ingressar no laboratório, durante as fases serão aplicadas técnicas baseadas em gamificação.

5.1 APRESENTAÇÃO DO CENÁRIO

A Universidade de Caxias do Sul (UCS) foi criada em 1967 e atende algumas cidades do estado do Rio Grande do Sul. Oferece cursos de graduação e pós-graduação em áreas de diversas finalidades, humanas e exatas. O laboratório de criação e aplicação de software está localizado no campus-sede em Caxias do Sul, no bloco 71, direcionado para os cursos de tecnologia. Atualmente a universidade oferece, na área de computação, os cursos de Ciência da Computação, Sistemas de Informação, Análise e Desenvolvimento de Sistemas, Tecnologias Digitais e Engenharia da Computação.

Hoje, a equipe do laboratório é composta por um corpo permanente de três professores (pode variar), um funcionário, cinco alunos em Trabalho de Conclusão de Curso, além das turmas das graduações que são acompanhadas pelos professores e podem executar certas demandas. No segundo semestre de 2016, sete turmas foram envolvidas nos projetos. Cada professor possui suas especialidades e baseado nisto, assumem certo papel dentro dos projetos. Nesta equipe multidisciplinar há

especialistas em projetos, modelagem de software, programação para dispositivos móveis, web e aplicações tradicionais para desktop. A universidade busca a contratação de alunos para atender as necessidades do laboratório e em contrapartida, oferece horas complementares pelo tempo aplicado em prol dos projetos.

Os projetos recebidos podem ser classificados entre Acadêmico, Mercado e Startup. A partir da necessidade de atendimento dos projetos acadêmicos o laboratório foi criado. Contudo, com o passar dos anos e o sucesso dos projetos, demandas classificadas como Mercado e Startup também surgiram. Os trabalhos realizados e que exigem um investimento do cliente (baseado nas horas da equipe), são classificados como Mercado. Projetos Startup são suportes a projetos que tem o intuito de tornarem-se produto de uma futura empresa. Vale lembrar também que não há restrição de temática das demandas recebidas, ou seja, o laboratório atende todos os tipos de serviço, desde necessidades da Agronomia até jogos relacionados à bolsa de valores.

O controle do processo atual e os documentos gerados é feito pela professora responsável pelos projetos e contato com o cliente. Dentro deste contato estão contemplados o levantamento de requisitos, possíveis conversas durante o desenvolvimento e a apresentação do produto final.

O desenvolvimento pode ser remunerado aos alunos, dependendo do projeto. Além disso, conforme citado anteriormente, pode ser tema de projetos de disciplina nas graduações e em trabalhos de conclusão de curso, pesquisa e estágio. Em síntese, o laboratório age como um portal que recebe as necessidades de criação de software e canaliza para desenvolvimento acadêmico. Este desenvolvimento é realizado pelo corpo de alunos, orientado pelos professores.

O laboratório hoje já possui um controle dos projetos, porém, há a necessidade de aprimorá-lo. Devido à equipe ser composta de alunos muitas vezes voluntariados, a rotatividade dos colaboradores é alta e os únicos fixos são os professores. Outro problema citado é a falta de tempo dos professores para ser aplicado somente ao laboratório. A demanda é alta para uma equipe pequena.

5.2 PROCESSO ATUAL

O processo atual segue o fluxo apresentado nas figuras a seguir. A Figura 8 representa o início do sequenciamento das atividades. Na Figura 9, há a continuação baseado nos resultados do processo antecessor.

É possível conferir que há um fluxo a ser seguido, portanto, há um controle das operações por parte do Gerente de Projeto. Este processo consiste na etapa antes da produção, ou seja, avaliação da possibilidade de desenvolvimento, levantamento de requisitos, arquitetura e definição das equipes.

Ao iniciar o projeto, o coordenador do laboratório incumbe-se de realizar uma avaliação prévia e definir os requisitos (feitos utilizando o formato de histórias), juntamente com o diagrama de caso de uso. Além disso, deve esboçar uma possível arquitetura e o diagrama de classes que contemplariam o projeto. Futuramente em reunião, estes artefatos serão discutidos e readequados conforme a necessidade e a opinião do professor especialista no assunto do projeto. Não está apresentado no processo, porém, neste momento é gerado o Termo de Abertura do Projeto, que oficializa o seu início.

Com a definição confirmada dos artefatos, cabe ao coordenador do laboratório criar e apresentar o plano de projeto. Este documento deve ser aprovado por ambas as partes após concluir se existirá uma integração com outro sistema. Na necessidade desta integração, o plano de projeto deve ser atualizado com os requisitos obrigatórios a serem atendidos para que a comunicação seja realizada.

No momento que o projeto está definido e as possíveis integrações são de conhecimento do cliente e da equipe do laboratório, cabe ao coordenador do laboratório designar os colaboradores que formarão a equipe de desenvolvimento deste projeto. Vale lembrar que, conforme descrito na Figura 9, os envolvidos podem ser alunos de disciplinas em andamento, futuros formandos realizando seu trabalho de conclusão de curso, estagiários, etc.

A partir da Figura 10, iniciam-se as atividades de desenvolvimento do projeto. O laboratório atualmente usufrui do sistema baseado em incrementos. Previamente, deve-se garantir a disponibilidade dos recursos de tecnologia e infraestrutura, essenciais para prosseguir as atividades. Todas as versões do software passam pelo fluxo de análise, codificação e teste unitário. As ocorrências encontradas nas validações são registradas.

O incremento gerado após o fluxo deve ser homologado pelo coordenador do projeto e caso tenha integração com outro sistema, o cliente deve indicar se a comunicação está de acordo e indicar as falhas, caso ocorram. Neste momento, o processo retorna à fase de análise e os erros são corrigidos. Por fim, o cliente deve realizar uma atividade chamada “Aceite final”, que caracteriza a aprovação do produto desenvolvido neste projeto. Após a validação, é gerado o Termo de Encerramento do projeto. Assim sendo, o projeto foi entregue e está finalizado.

As tarefas são designadas a cada colaborador por meio de post-its em um quadro. A utilização de um software para este fim não foi bem aceita pela equipe, devido à dificuldade na usabilidade.

Neste modelo, as colunas referem-se aos projetos e as linhas, os colaboradores da equipe. Os nomes foram apagados para que o sigilo seja mantido. Os post-its contém a lembrança do que deve ser realizado neste projeto pelo indivíduo. Os prazos eram controlados por uma ferramenta, porém, não foi bem aceita pelos colaboradores e não é utilizada da melhor forma.

O controle do andamento do projeto é feito por outro quadro, semelhante a um Kanban, que possui as etapas definidas pela gerente de projeto. Está apresentado na Figura 8.

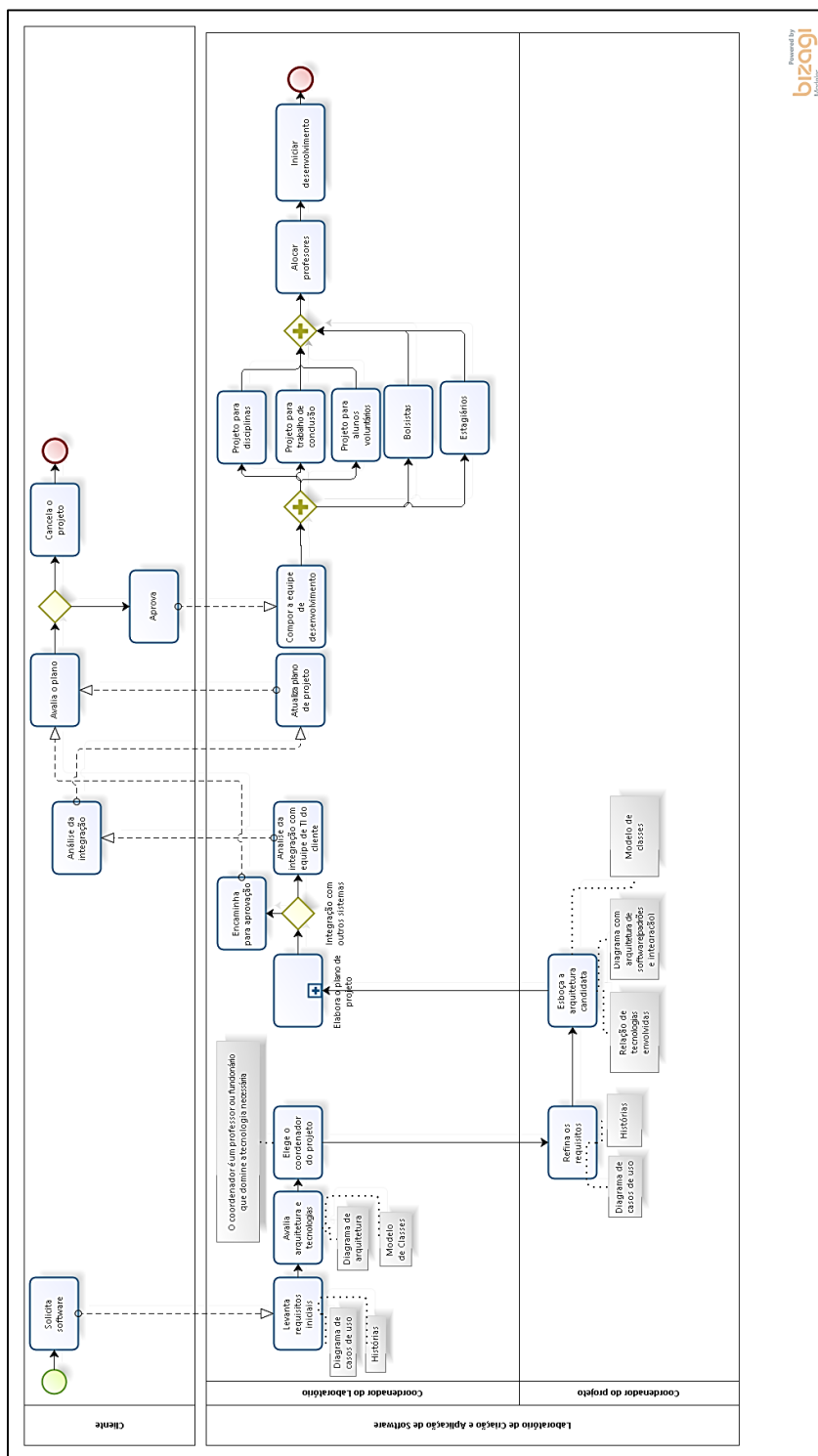
O quadro possui cinco classificações respectivamente: suspenso, prospecção, concepção, construção e entregue. Projetos suspensos não estão mais em andamento. Prospecção está em fase de negociações, portanto, nenhuma modelagem e desenvolvimento foi feita neste momento, contudo, há a possibilidade de tornar-se um projeto. Concepção significa que o trabalho foi aprovado com o cliente e está sendo analisado e modelado. Em construção, o projeto está modelado e as tarefas foram repassadas aos programadores. Entregue, está no cliente em fase de testes ou em modo produção. Este processo hoje é feito totalmente de forma manual. As únicas aplicações auxiliares utilizadas hoje são software para modelagem UML, software para modelagem BPMN, o software que recebe as tarefas (atualizado hoje apenas pelo Gerente de Projeto) e o Google Drive para armazenamento de arquivos.

Figura 7: Situações dos projetos



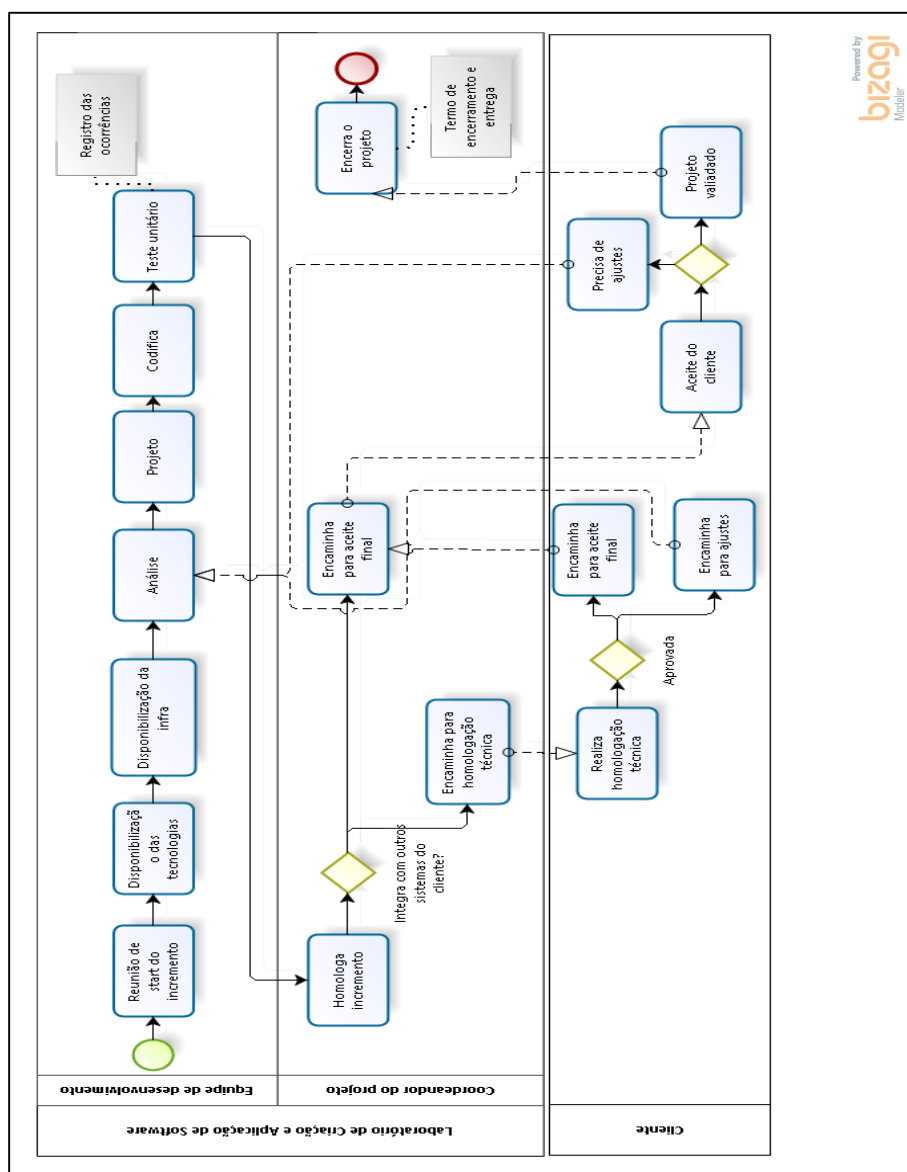
Fonte: Próprio autor.

Figura 8: Processo de planejamento atual



Fonte: DE CARLI, Iraci Cristina da.Silveira.; NOTARI, Daniel Luis; ZAMPIERI, Andre, 2015

Figura 9: Execução do projeto atual

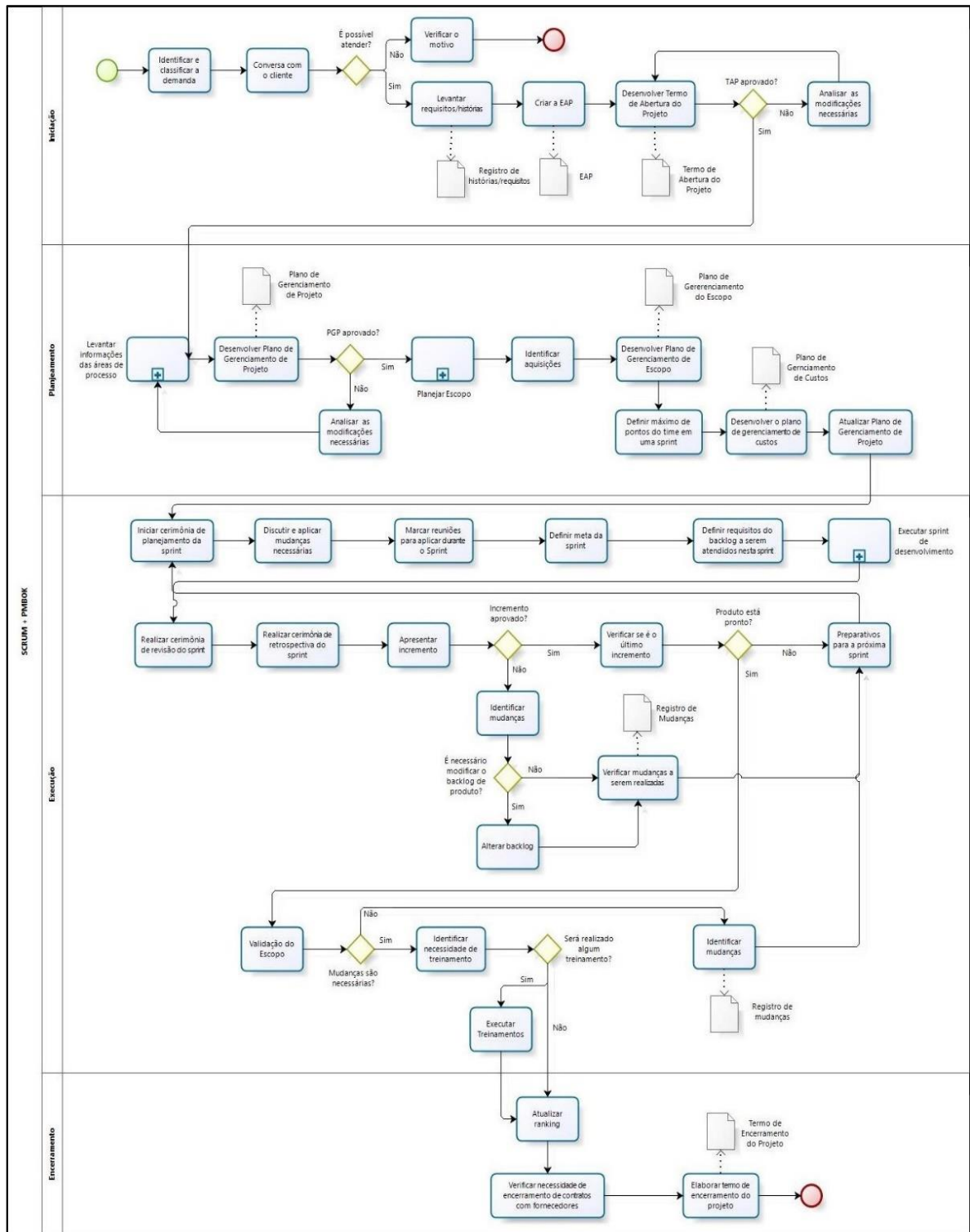


Fonte: DE CARLI, I.C.S.; NOTARI, Daniel Luis; ZAMPIERI, Andre, 2015

5.3 PROPOSTA DE ADAPTAÇÃO DO PROCESSO

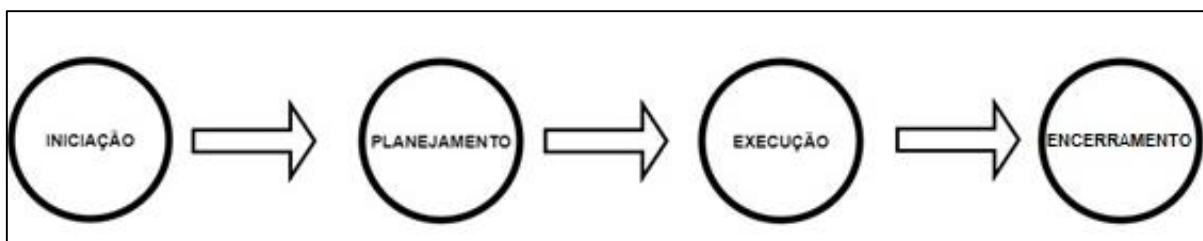
As alterações no processo de gerenciamento de projetos do laboratório foram elaboradas baseado nas atividades propostas por CRUZ (2013). O autor desenvolveu uma forma de integração entre a metodologia Scrum e os processos definidos no guia PMBOK®. O sequenciamento das quatro fases está apresentado na Figura 11. As atividades de cada fase estão demonstradas na Figura 10.

Figura 10: Processo completo



Fonte: Próprio autor.

Figura 11: Fluxo proposto



Fonte: Próprio autor.

O grupo de iniciação implica nas atividades relacionadas às primeiras comunicações com o cliente e o acerto para a futura geração do produto. O planejamento comporta todo o processo que embasará a execução, ou seja, elaboração do escopo, documentos para o controle do projeto e levantamento de requisitos. A execução baseia-se no fluxo proposto pelo Scrum, porém com algumas alterações de acordo com o contexto atual do laboratório. O encerramento faz o levantamento das mudanças, caso necessárias, após a apresentação do incremento final ao cliente. Promove o encerramento das atividades de controle proposta nas etapas anteriores. O processo completo demonstrado na Figura 10 é detalhado nos capítulos seguintes.

5.3.1 Iniciação

A fase de iniciação tem o intuito de formalizar o contato com o cliente e promover o início do projeto. A Figura 12 apresenta o fluxo neste grupo de forma mais clara, a partir de um recorte da Figura 10.

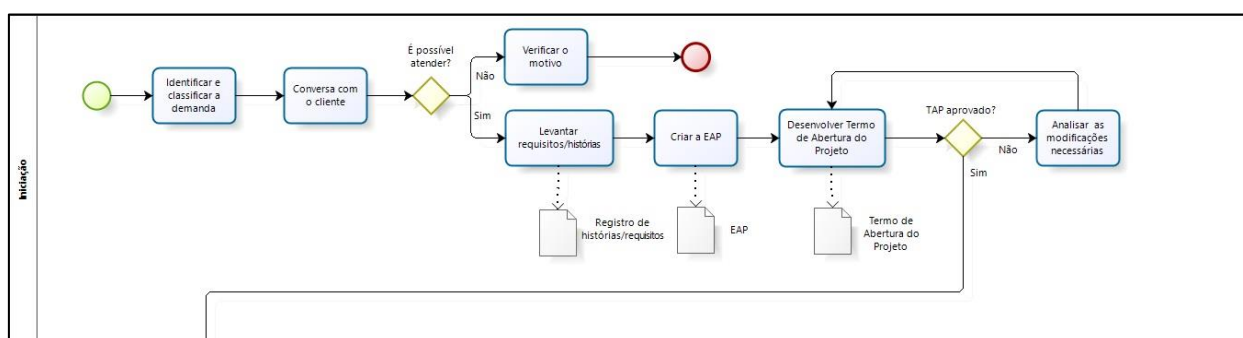
Comparando com o processo anterior, algumas atividades são mantidas. A identificação e classificação da demanda será feito da mesma maneira, ou seja, os projetos ainda podem receber a classificação de Acadêmico, Mercado e Startup. O tipo do projeto é indiferente para as fases seguintes.

Após a identificação da necessidade de demanda deste cliente e esta já ter recebido a sua classificação, deve ser realizada uma conversa com a parte interessada para melhor entendimento do que deve ser desenvolvido. Esta conversa ainda não é um levantamento de requisitos, porém, o gerente de projeto/analista de negócio (no caso do laboratório, o mesmo professor executa as duas funções) deve ter em mente o problema a ser solucionado e uma breve solução que poderá ser adotada ou não. As atividades relacionadas à gerência de projetos desde professor relacionam-se com a administração geral do projeto, tais como o cuidado com a equipe, as aquisições necessárias, etc. As funções de analista de negócio dirigem-se

ao levantamento do que o software deverá atender, no momento, sem pensar em como isto será desenvolvido. Neste momento, sugere-se o uso da prática 5W2H.

A partir desta conversa, cabe ao gerente de projeto/analista de negócio definir se a estrutura do laboratório é capaz de atender esta solicitação. Ou seja, deve analisar as competências da equipe, a complexidade do projeto e diagnosticar a possibilidade de progredir com o desenvolvimento. Caso prossiga, o primeiro passo é o encontro com o cliente para a definição dos requisitos (ou histórias, deve ser definido pelo GP e equipe qual a melhor maneira para este projeto). Neste momento, todas as necessidades que o produto deve atender são declaradas pelo cliente e devem ficar claras para o analista, visto que posteriormente estas informações serão repassadas à equipe. Não há um documento formal que registre estas necessidades, porém, o cliente deve analisar o registro feito e concordar com o levantamento.

Figura 12: Fase de iniciação



Fonte: Elaborado pelo autor.

O registro embasará a criação do backlog do produto. Este documento será a base da equipe durante os sprints de desenvolvimento, pois contém todos os requisitos a serem atendidos. A Estrutura Analítica de Projeto (EAP) deve ser desenvolvida também para tornar a visualização mais clara do desenvolvimento de cada requisito.

Após a conclusão da atividade precedente, devem ser documentados os dados relacionados ao Termo de Abertura de Projeto (ver Apêndice A), que são:

- a) Situação atual;
- b) Critérios de sucesso do projeto;
- c) EAP realizada com o levantamento das histórias;
- d) Marcos;
- e) Partes interessadas;
- f) Restrições;
- g) Premissas;

CRUZ (2013) cita que é recomendado identificar o Termo de Abertura de Projeto independente da sua complexidade. Após isto, o projeto é passado para a fase de Planejamento. Caso não seja possível acolher este projeto, o fluxo é finalizado.

5.3.2 Planejamento

O ciclo de planejamento é iniciado após o sucesso da fase de iniciação. A equipe pode atender esta demanda e o Termo de Abertura de Projeto foi aprovado. As atividades executadas neste período estão detalhadas na Figura 13.

Nesta fase, cabe ao gerente de projetos e a equipe levantar o detalhamento necessário para a execução das sprints do Scrum, como o software será desenvolvido e os controles propostos pelo PMBOK®. Portanto, o primeiro passo a ser dado neste fluxo é determinar oficialmente quem são as partes interessadas, as formas de comunicação utilizadas, como deve ser feito o controle de qualidade e os possíveis riscos do projeto. Estas informações estarão presentes no Plano de Gerenciamento de Projeto. Este fluxo está definido na Figura 14, retirado da Figura 10.

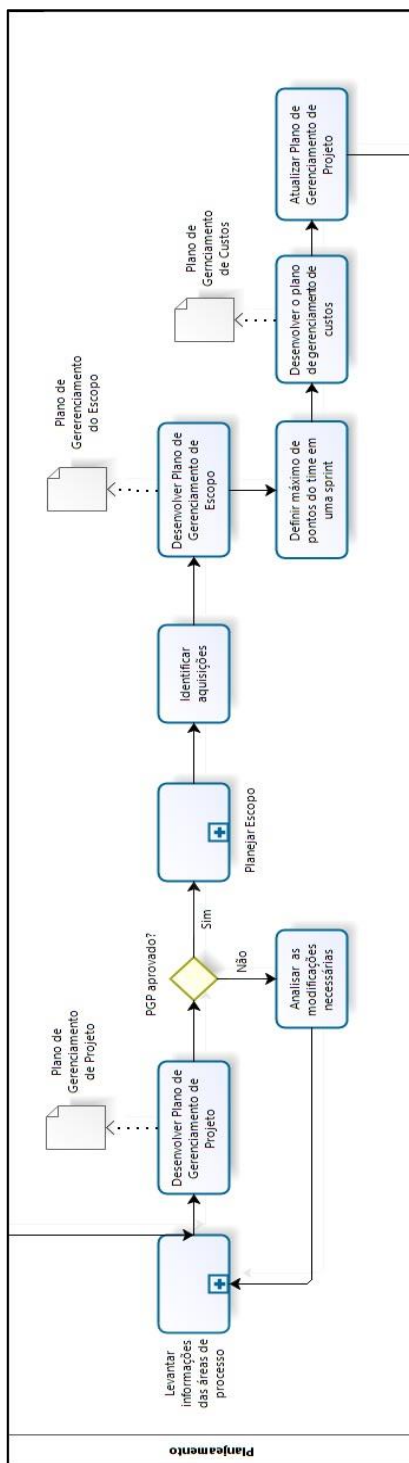
Baseado nas informações definidas na atividade anterior e no termo de abertura do projeto, o gerente de projetos deve documentar as informações que caracterizam o PGP (Plano de Gerenciamento de Projeto). O modelo está apresentado no Apêndice B. Este registro deve contemplar todos os dados necessários para a administração deste trabalho e deve ser disponibilizado a todos da equipe. São eles:

- a) Registro de requisitos/histórias
- b) EAP
- c) Registro de partes interessadas
- d) Registro de membros da equipe e funções
- e) Registro de riscos
- f) Registro de formas de comunicação
- g) Registro de requisitos de qualidade
- h) Registro de aquisições
- i) Arquitetura e diagramas UML

A aprovação por todas as partes interessadas e o time também é obrigatória. Esta validação existe para minimizar a chance de falta de algum aspecto.

A próxima atividade é o planejamento do escopo. Este momento é crucial para a próxima fase e deve ser realizada com cautela e precisão. A Figura 15 apresenta como este subprocesso deve ser aplicado. Com os requisitos/histórias declarados e a EAP confirmada, é possível desenvolver o backlog do produto, registrando as histórias de forma técnica. O gerente de projeto, juntamente com a equipe e o Dono do Produto devem definir a prioridade de desenvolvimento dos itens estipulados. No futuro, isto apoiará a criação das sprints de desenvolvimento.

Figura 13: Fase de planejamento



Fonte: Elaborado pelo autor

Neste momento o primeiro elemento da gamificação será aplicado. Para determinar a ordem de desenvolvimento será aplicada a estratégia do Planning Poker, que resumidamente consiste em atribuir pontos a cada necessidade baseado na complexidade e na importância. Com os frutos desta reunião o backlog de produto deve ser atualizado com as prioridades. Além disso, o gerente de projetos pode elaborar o MDP, que identifica as relações entre os requisitos/histórias.

Mediante o esclarecimento do problema que deve ser resolvido pela equipe, cabem ao analista e aos desenvolvedores estabelecer qual será a arquitetura lógica e física utilizada no desenvolvimento. Por exemplo, será aplicado em uma plataforma WEB ou será um aplicativo para dispositivo móvel? Deve ter as duas possibilidades? Baseado nisto, qual será o banco de dados que armazenará os dados? Visto que o esqueleto do produto será definido neste momento, esta análise é de suma importância para o decorrer do projeto. O analista de sistemas deve documentar a modelagem necessária utilizando UML e esta fundamentação deve ser revisada e aprovada.

Definidos os requisitos e a forma de desenvolvimento, o gerente de projeto deve verificar se há necessidade de o laboratório adquirir alguma ferramenta ainda não disponível.

Antes da fase de execução o time deve identificar qual é a pontuação máxima a ser atendida dentro de uma sprint, ou seja, qual é a velocidade de execução das tarefas. No início, devido a este processo nunca ter sido aplicado, a estimativa pode ser falha. Contudo, CRUZ (2013) ressalta que o importante é a estimativa existir, pois mais que não seja totalmente correta. Além disso, é obrigatório conferir o Plano de Gerenciamento de Projeto e atualizá-lo caso tenha alguma mudança, por exemplo, se alguma necessidade de aquisição foi identificada ou algum risco novo foi encontrado.

Por fim, o gerente de projeto deve levantar os recursos utilizados e desenvolver o Plano de Gerenciamento de Custos (modelo apresentado no Apêndice C), que receberá o valor dos recursos empregados no trabalho. Deve conter as informações a seguir.

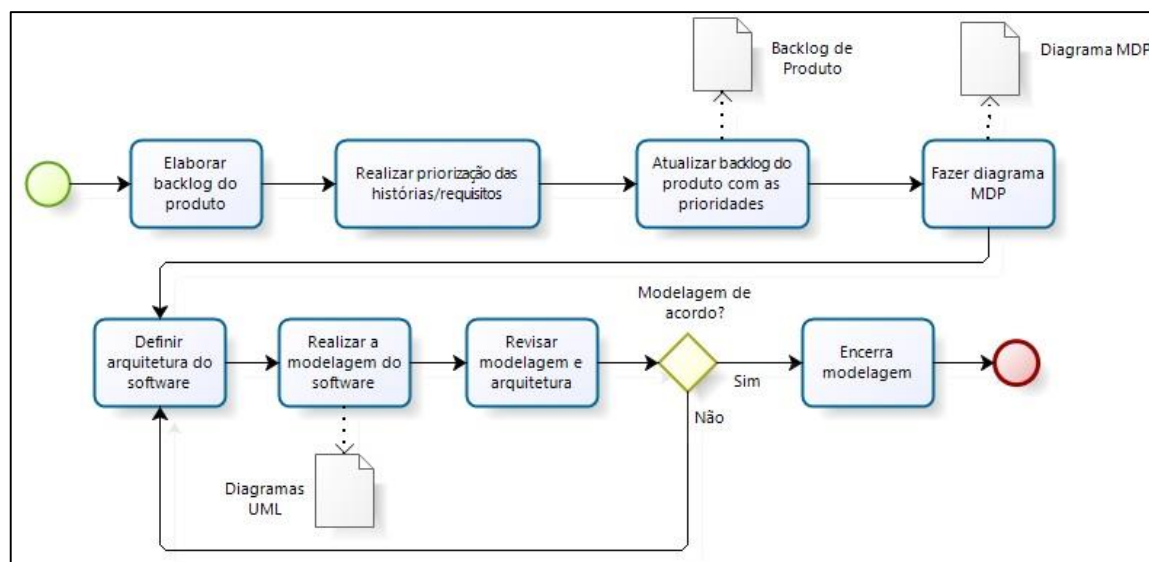
- a) Lista de Aquisições realizadas e o valor acertado nos contratos;
- b) Estimativa de custos baseada na EAP (valor/hora para desenvolver cada requisito levantado para atendimento a este projeto);

Figura 14: Levantar informações das áreas de processo



Fonte: Elaborado pelo autor

Figura 15: Planejar escopo



Fonte: Elaborado pelo autor

Devido à grande rotatividade da equipe, os alunos egressos recentemente no laboratório ter a mesma pontuação máxima de desenvolvimento, independentemente da sua capacidade. Após um período de três meses de experiência, este tempo deve ser revisto para otimizar o cadastro das sprints.

Baseado nas demais informações levantadas até o momento, o Plano de Gerenciamento de Projeto deve ser consultado e atualizado com as informações relacionadas diretamente ao software, tal como a arquitetura utilizada, linguagem de programação e banco de dados. Durante todas as atividades praticadas após o Levantamento das informações das áreas de processo, alguns itens podem ser identificados e há a necessidade de atualizá-los no PGP. É de suma importância que este documento disponha de todas as informações desde projeto para fins de rastreabilidade futuras.

5.3.3 Execução

A fase de execução do projeto começa após a conclusão das priorizações dos requisitos/histórias e definição da arquitetura. Neste momento é aplicado todo o conceito trazido pelo Scrum, no que tange as sprints com tempo pré-determinado e geração do produto de forma incremental. Além disso, neste momento a gamificação

é aplicada efetivamente, buscando o engajamento da equipe. O fluxo desta fase está definido na Figura 16, advindo da Figura 10.

O ciclo inicia-se na cerimônia de planejamento da sprint, cujo intuito é a definição dos requisitos/histórias que serão atendidos no período definido. Os resultados desta reunião são originados do backlog do produto já priorizado, conforme realizado na fase de planejamento. Após isto, estes requisitos devem ser especificados em tarefas, que serão desenvolvidas pela equipe. Nesta reunião também é realizada a atualização do ranking de pontuação de acordo com as histórias/requisitos já atendidos (ver capítulo 5.4).

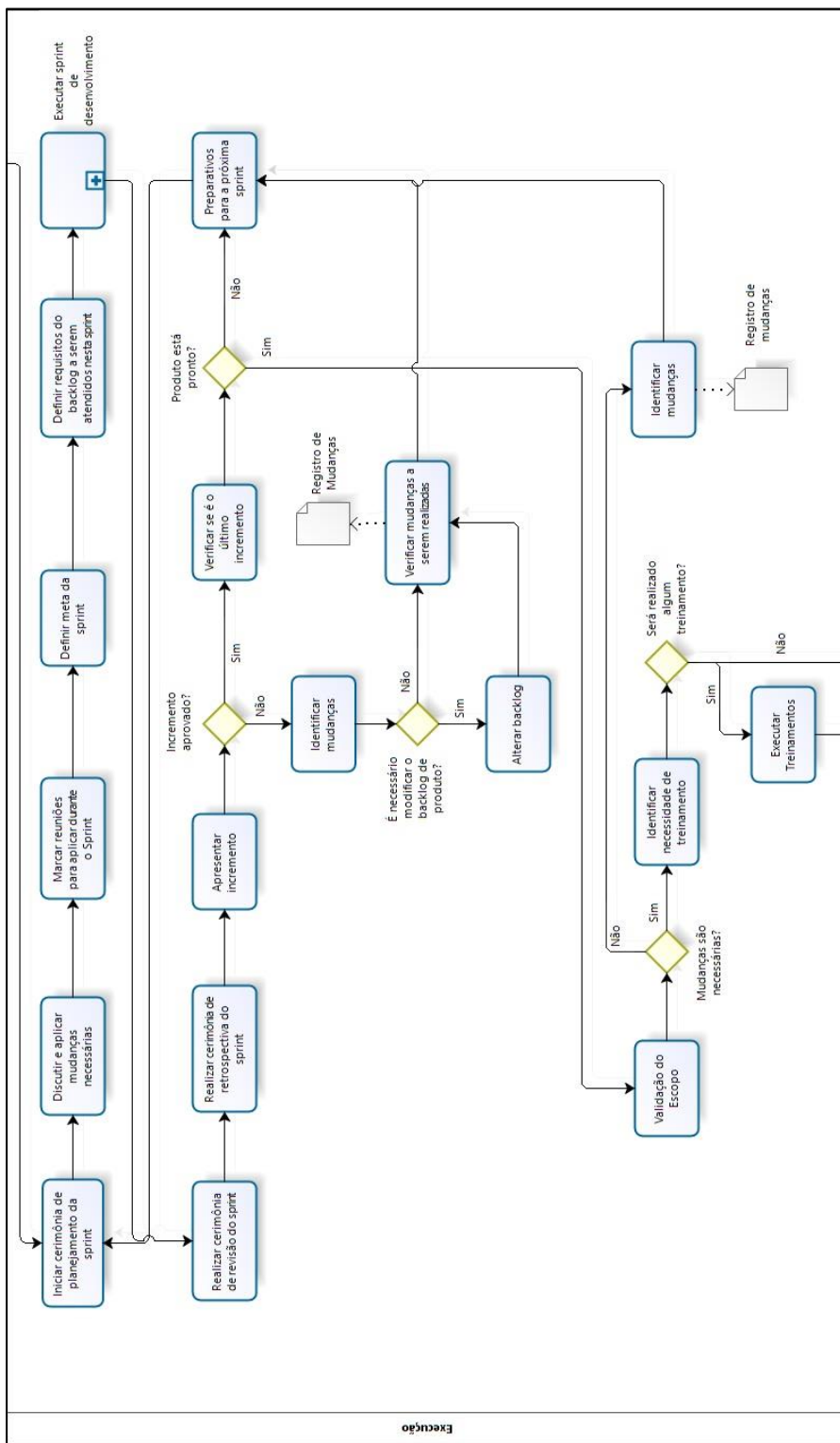
Além disso, deve ser definido pela equipe e o Dono do Produto uma meta do Sprint, ou seja, ao fim deste “jogo”, quais devem ser os resultados. A meta, além do sistema de classificação, é outro elemento advindo da gamificação. Colaboradores com espírito competitivo tendem a trabalhar de maneira mais árdua para alcançar o objetivo definido. Nesta cerimônia devem ser marcados encontros durante o período para conversas breves sobre o projeto. Como os professores e alunos não dedicam todo o tempo para o laboratório, infelizmente não há possibilidade da aplicação da prática de Reunião Diária. Porém, alguns encontros são relevantes e devem acontecer. Finalizada a cerimônia de planejamento, a equipe engaja-se a realizar a sprint. Este processo deve seguir as seguintes atividades, conforme a Figura 17.

As tarefas devem ser registradas no taskboard, assim, a equipe tem conhecimento da situação de cada demanda e em requisitos TI (Término para Início), identifica-se quando é possível iniciar o desenvolvimento que dependia de outra tarefa.

O quadro deve ser semelhante a um *Kanban*, ou seja, os requisitos/histórias aguardando desenvolvimento, em andamento, finalizados e aguardando validação. O colaborador deverá desenvolver a tarefa considerando a arquitetura lógica e física definida no planejamento. Isto é importantíssimo para que a qualidade futura seja garantida. Ao fim do desenvolvimento, é gerado um incremento do software e a tarefa é modificada para a situação “Resolvida”. Ao fim do sprint, todo o desenvolvimento deve gerar uma versão final que será apresentada ao Dono do Produto. Além disso, duas cerimônias são executadas. A Revisão do Sprint determina o que foi desenvolvido ao longo do período e o que será apresentado. Deve garantir juntamente com a equipe a qualidade deste incremento, ou seja, se foi desenvolvido de acordo com a modelagem e atende a proposta dos requisitos/histórias. A retrospectiva tem o objetivo de diagnosticar possíveis imprevistos e problemas encontrados durante o sprint. Antes do início do próximo ciclo, os itens encontrados pela equipe devem ser propostos durante a cerimônia de planejamento e ideias para solução são propostas.

A apresentação da versão poderá ser aprovada ou não. Caso o produto esteja em desacordo, cabe ao time identificar as mudanças necessárias, compará-las com o backlog do produto e remanejar as sprints seguintes. Novamente, não há um documento padrão que receba os requisitos de alteração.

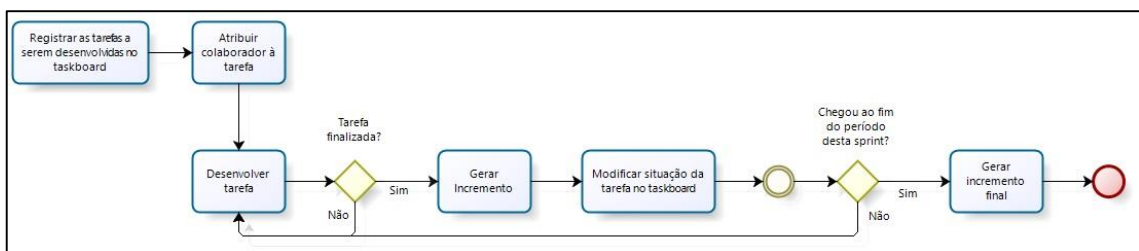
Figura 16: Fase de execução



Fonte: Elaborado pelo autor

Caso o sprint executado tenha sido o último, o incremento gerado será o produto final aguardando a validação. Portanto, esta apresentação validará se o produto atende as expectativas do cliente. As validações de qualidade durante o processo Scrum tornam baixa a probabilidade de reprovação. Da mesma forma, caso mudanças sejam necessárias, uma nova sprint deve ser elaborada para o desenvolvimento das soluções. No momento que o incremento final é aprovado, o gerente de projetos deve diagnosticar a necessidade de treinamentos do produto desenvolvido, agendá-los e executá-los.

Figura 17: Sprint

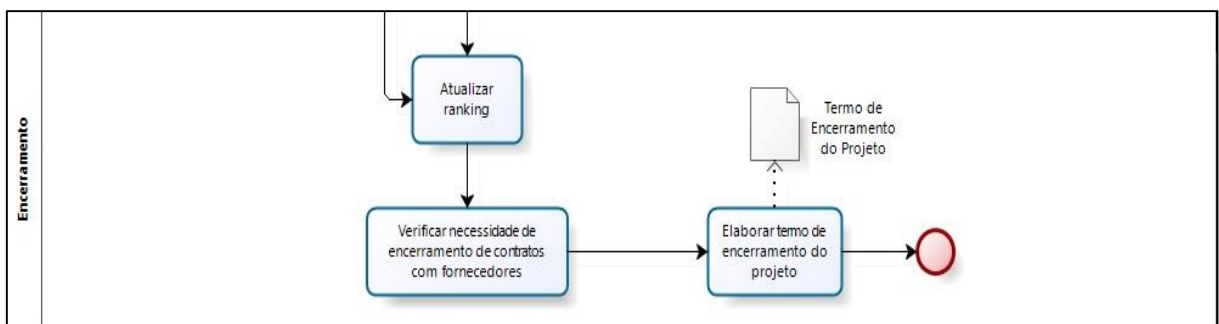


Fonte: Elaborado pelo autor

5.3.4 Encerramento

Ao passo que os treinamentos foram finalizados e o produto entregue com a validação do cliente, o projeto chega à fase de Encerramento. Nesta altura da aplicação do processo proposto, as seguintes atividades devem ser executadas (ver Figura 18, originada da Figura 10).

Figura 18: Fase de encerramento



Fonte: Elaborada pelo autor

A finalidade da fase de encerramento, em suma, é atualizar a pontuação dos colaboradores neste projeto, encerrar contratos em aberto com fornecedores e formalizar o término juntamente com as partes interessadas. Ambas as partes devem aprovar o Término de Encerramento do Projeto (ver Apêndice D). Neste, o gerente de projeto e a equipe devem definir as lições aprendidas durante todo o processo com o intuito de promover a melhoria contínua.

5.4 FORMAS DE ENGAJAMENTO

Conforme salientado na apresentação do cenário, o laboratório de criação e aplicação de software possui uma rotatividade alta de colaboradores. Pelo aluno não ter uma ligação formalizada com o local (não há contrato), possui a liberdade de desligar-se quando quiser, inclusive em meio a um projeto. Portanto, uma das finalidades deste trabalho, além de propor o aprimoramento da coordenação do processo, é engajar os atuais presentes na equipe e cativar mais alunos a produzir e principalmente aprender no laboratório.

Para isso, a gamificação aparece como a metodologia ideal para o ambiente. Devido ao laboratório estar presente no bloco de tecnologia da universidade, há uma grande chance dos alunos identificarem-se com a proposta gamificada pelo seu perfil já acostumado com jogos. Além disso, propõe-se uma competitividade entre os integrantes no que tange a sua produção dentro do laboratório e o favorecimento do relacionamento com colegas da área.

As atividades a seguir devem ser aplicadas no laboratório como modo de ludificação:

a) Pontuação e Ranking

O método Scrum concentra-se no desenvolvimento em ciclos fechados de tempo. Estes fluxos contêm tarefas a serem desenvolvidas com certa pontuação baseado no esforço e na importância. Esta atividade propõe o armazenamento dos pontos produzidos por cada colaborador, ou seja, o nível das atividades que cada indivíduo produziu de acordo com o planejado. Ao passo que o colaborador está engajado com o projeto, atingindo as metas do sprint por realizar os objetivos, pontos são acumulados. A aplicação deste método influencia principalmente os indivíduos identificados com os perfis Predador e Realizador. A motivação intrínseca que os remete a almejar a vitória e portanto, teoricamente, produzirão mais. A competitividade é o prazer.

O sistema de ranking apresenta os colaboradores que teoricamente são mais eficientes, visto que estão pontuando devido a alcançar os objetivos do planejamento. Além disso, é possível identificar se o indivíduo está colaborando com o projeto, sendo uma ferramenta para o gerente de projeto identificar a necessidade de motivação de um integrante da equipe. A atualização deste ranking está presente no início de cada

sprint, ou seja, baseia-se na aplicação do colaborador no ciclo anterior. Na cerimônia de planejamento do sprint há o *feedback* da etapa precedente, portanto, o colaborador receberá a pontuação somente caso tenha concluído sua pendência em tempo. Isto determina o desempenho do indivíduo, além de auxiliar no possível recálculo do máximo de pontos atendidos pelo time.

Além disso, o colaborador pode elevar sua pontuação a partir de duas outras maneiras. Conforme citado anteriormente, a falta e rotatividade de pessoal é um problema enfrentado hoje pelo laboratório. Logo, faz-se a necessidade de cativar voluntários para compor a equipe. A metodologia oferece pontuação bônus aos alunos que indicarem colegas a participarem do projeto, incentivando os indivíduos com perfil de jogador Socializador. Portanto, seu desejo é uma equipe que esteja unida em prol do objetivo. Caso consiga engajar seus colegas, sua motivação intrínseca tende a crescer cada vez mais. Além disso, no momento que o componente da equipe propõe uma mudança válida a ser aplicada no processo, recebe pontuação bônus também. O colaborador Explorador é ideal para este fim, pois durante seu processo de desenvolvimento procura perceber o todo ao seu redor e terá facilidade no reconhecimento de possíveis mudanças.

b) Níveis

Os colaboradores da equipe, baseado no sistema de pontuação, também recebem níveis. Por ser localizado no bloco de TI e projetos voltados à tecnologia, sugere-se a utilização de níveis e uma evolução gradativa. Os alunos devem buscar os níveis conforme seus avanços nos projetos Scrum.

A recompensa oferecida pelo laboratório e conseqüentemente pela UCS são horas complementares. No término da graduação, é obrigatório o cumprimento de uma carga horária de atividades além dos estudos da disciplina, portanto, o tempo destinado aos projetos destina-se à conclusão do curso. Além disso, um bom desempenho e competências registradas neste sistema mostra a eficiência do aluno e favorece para possíveis oportunidades no mercado identificadas pelos professores.

6 DESENVOLVIMENTO DA SOLUÇÃO

O presente capítulo demonstra o desenvolvimento da solução aplicada ao caso do laboratório de criação e aplicação de software. A seção apresenta a avaliação das ferramentas e a implementação da proposta.

6.1 AVALIAÇÃO DA FERRAMENTA

Para aplicação do processo para o laboratório, cogita-se algumas ferramentas que possam atender o fluxo proposto. Estas ferramentas devem possuir as funcionalidades que o sequenciamento de atividades propõe, os elementos de gamificação, além de ser atrativo para o usuário.

Neste caso, três ferramentas são avaliadas para indicar o caminho a seguir. Há a possibilidade destas aplicações serem utilizadas nos trabalhos do laboratório, portanto, esta fase de avaliação é muito importante. Assim sendo, alguns critérios de avaliação devem ser levantados para garantir o êxito deste julgamento. A ISO/IEC 9126 em conjunto com a ISO/IEC 14598 normatizam esta análise, conforme explícito na seção 6.1.1.

6.1.1 NBR ISO/IEC 9126 e 14598

As normas NBR ISO/IEC 9126 (2001) e 14598 (1999) indicam como uma avaliação de software deve proceder para garantir que atenda o que é proposto, ou seja, seu nível de qualidade. Neste trabalho não está elucidado detalhadamente como as normas estão constituídas. Em suma, estes documentos são utilizados para avaliação e auxílio para comparação de software baseado em critérios de qualidade. A ISO/IEC 9126 (2001) declara como levantar requisitos de qualidade e a ISO/IEC 14598 (2001), como realizar o processo de avaliação.

A ISO/IEC 9126 (2001) define um modelo de qualidade baseado em requisitos de qualidade interna, externa e em uso. A qualidade interna refere-se à constituição do software e formas de avaliações internas, tais como a revisão de código e testes automatizados. A qualidade externa aplica-se ao que é visível e sentido durante os testes do avaliador, critérios como desempenho e defeitos apresentados após a geração da versão do software, por exemplo. Por fim, a qualidade esperada pelos usuários que utilizam o produto é chamada de qualidade em uso.

Foram levantadas cinco subcaracterísticas julgadas como mais relevantes na avaliação da ferramenta, devido a serem obrigatórias para o produto final. O software deve possuir estas características para ser considerado adequado ao laboratório. No

grupo Funcionalidade, a Adequação e a Acurácia estão sendo avaliados. Além disso, no ramo Usabilidade, a Operacionabilidade e Apreensibilidade são critérios de julgamento. A Coexistência, subcaraterística de Portabilidade, também será medido. A seguir estão esclarecidos os significados de cada subcaracterística.

- a) Adequação: aptidão do software para atender as necessidades do usuário;
- b) Acurácia: capacidade do software de demonstrar resultados exatos, diante do que foi acordado com as partes interessadas;
- c) Operacionabilidade: competência do software para permitir o usuário utilizá-lo corretamente;
- d) Apreensibilidade: tendência do software a possibilitar que o indivíduo aprenda-o;
- e) Coexistência: Acessar o produto em diferentes sistemas operacionais mas com o mesmo ambiente.

Baseado nos critérios acima, a norma ISO/IEC 14598 (1999) orienta a levantar métricas que possam ser medidas e gerar valores possíveis de avaliação. Neste trabalho, as métricas serão definidas a partir de requisitos para cada subcaracterística. Os requisitos estão definidos a seguir:

- a) Adequação:
 - 1. Capacidade de registrar tarefas detalhadamente;
 - 2. Capacidade de registrar o que foi feito em cada tarefa;
 - 3. Comportar vários usuários para acessar o projeto;
 - 4. Gamificação baseado em resultados;
 - 5. Registro de documentos;
 - 6. Armazenar backlog do produto;
 - 7. Software deve ser gratuito;
- b) Acurácia:
 - 1. Capacidade de apresentar o que foi realizado em cada tarefa;
 - 2. Capacidade de demonstrar o andamento do projeto;
 - 3. Permitir visualizar os resultados da pontuação baseado na gamificação;
- c) Operacionabilidade:
 - 1. Capacidade do software de indicar o caminho que o usuário deve seguir nos primeiros acessos;
 - 2. O software possui manual de operações;
 - 3. O software apresenta avisos legíveis de erros nas operações;
 - 4. O produto possui validações de tipos de valores inseridos nos campos;
 - 5. O software possui o idioma adequado para a equipe;
- d) Apreensibilidade:
 - 1. O software possui manual de instruções;

2. Número de cliques para o cadastro da primeira tarefa sem necessitar de auxílio e sem consulta ao manual;
 3. O software utiliza a gamificação para guiar o usuário durante o uso;
- e) Coexistência:
1. O software está presente em mais de um sistema operacional;
 2. É possível acessar suas funcionalidades a qualquer momento;

A partir destes requisitos, três produtos serão avaliados utilizando o cálculo de média ponderada. Cada requisito poderá receber uma nota de um a cinco, que deve ser aplicada logo em seguida em uma média ponderada. Neste método, as subcaracterísticas recebem pesos de acordo com a sua importância e o impacto da mesma no projeto.

No capítulo 6.1.2, são demonstrados os resultados da avaliação aplicada baseado nas métricas.

6.1.2 Resultados da avaliação

Os resultados estão apresentados na Tabela 1. As ferramentas avaliadas foram os softwares Asana³, Redmine⁴ e Wrike⁵. Os produtos escolhidos foram determinados a partir de um levantamento prévio que avaliou softwares de gestão de projetos. Além disso, poderiam ser usados em um método ágil. Outras ferramentas foram observadas mas que não estão nesta avaliação, tais como o Producteev, Trello e JProject-open[.

O software para gestão de projetos Asana foi escolhido devido à sua facilidade no aprendizado, sua portabilidade para outros sistemas operacionais e por oferecer uma interface amigável. Além disso, permite o acesso multiusuário ainda na versão gratuita, o que torna mais interessante ao laboratório. A respeito da sua integração com outros sistemas operacionais, a possibilidade de baixar o aplicativo para o smartphone e acompanhar o andamento do projeto a qualquer momento pondera o software positivamente dentre boa parte das ferramentas avaliadas. Para comprovar a eficiência do Asana, os autores SMITS e MOGOS demonstram o uso da aplicação no *European Conference On Information Systems* (2013). Utilizaram-se desta ferramenta na sua pesquisa sobre o impacto das mídias sociais no desempenho dos negócios. A ferramenta foi aplicada no setor de marketing e por ela foi possível monitorar as tarefas de cada usuário. Desta forma, a pesquisa dos autores foi bem-sucedida usufruindo dos artifícios desta aplicação.

³ Link: <https://asana.com/>

⁴ Link: <http://www.redmine.org/>

⁵ Link: <https://www.wrike.com/pt-br/>

A ferramenta open-source Redmine também está dentre as escolhidas para a avaliação principalmente pela sua possibilidade de customização, mesmo que não esteja dentre os requisitos avaliados. Isto é necessário pois caso não seja possível encontrar uma ferramenta que atenda o processo proposto, uma adequação deve ser realizada e o Redmine está disponível em código aberto, caso necessário. O idioma português suportado pela aplicação também é um quesito diferencial dentre os demais. Possui também outras aplicações que integram diretamente com o sistema. Durante o Malaysian Conference in Software Engineering (2013), apresentou-se a possibilidade do uso do Redmine na aplicação de um processo ágil, demonstrando a sua adaptabilidade a ambientes diversos. Sua flexibilidade comparada a um software específico para métodos ágeis, conforme demonstrado no artigo elaborado neste evento, torna-o interessante para utilização no fluxo indicado neste trabalho.

Por último, a aplicação Wrike surge como o principal concorrente do Asana e promove-se citando que sua usabilidade é consideravelmente melhor. Na avaliação, o principal critério a favor desta ferramenta é a sua maior integração com o formato ágil, contudo, perde pontos por sua versão gratuita ser disponibilizada por apenas um mês. NASCIMENTO (2017) destaca a possibilidade de geração de relatórios demonstrando a situação dos projetos e o uso de métricas automáticas para levantamento de desempenho. Este último quesito pode ser aproveitado durante a aplicação do processo abordado neste trabalho.

Antes de apresentar a tabela, alguns detalhes são cruciais para o entendimento da avaliação. Os requisitos determinados com peso dez são obrigatórios para o projeto, portanto, no momento que a ferramenta não atende a necessidade a mesma não é ideal para o processo. A avaliação foi feita utilizando uma técnica de Observação Simples. GIL (2008) cita que este método pede ao menos um controle sobre os dados, que está apresentado na Tabela 1. Esta observação foi realizada baseado na opinião do autor. Cada requisito pode receber uma nota de 0 a 5, conforme a sua adequação com o item avaliado.

A Tabela 1 está apresentada a seguir.

Tabela 1: Resultados da Avaliação

Requisito		Peso	Asana	Redmine	Wrike
Adequação					
1	Capacidade de registrar tarefas detalhadamente	10	4	5	5
2	Capacidade de registrar o que foi feito em cada tarefa	9	2	3	5

	Requisito	Peso	Asana	Redmine	Wrike
3	Comportar vários usuários para acessar o projeto	10	5	5	5
4	Gamificação baseado em resultados	10	0	0	0
5	Registro de documentos	7	3	5	5
6	Armazenar backlog do produto	8	0	0	5
7	O software deve ser gratuito	10	5	5	0
Acurácia					
8	Capacidade de apresentar o que foi realizado em cada tarefa	9	2	3	5
9	Capacidade de demonstrar o andamento do projeto	8	5	3	5
10	Permitir visualizar os resultados da pontuação baseado na gamificação	10	0	0	0
Operacionabilidade					
11	Capacidade do software de indicar o caminho que o usuário deve seguir nos primeiros acessos	7	5	0	3
12	O software possui manual de operações	6	5	5	5

Requisito		Peso	Asana	Redmine	Wrike
13	O software apresenta avisos legíveis de erros nas operações	6	5	5	5
14	O produto possui validações de tipos de valores inseridos nos campos	5	5	3	3
15	O software possui o idioma adequado para a equipe	7	3	5	3
Apreensibilidade					
16	Número de cliques para o cadastro da primeira tarefa sem auxílio e consulta ao manual	7	4	3	5
17	O software utiliza a gamificação para guiar o usuário durante o uso	8	0	0	0
Coexistência					
18	O software está presente em mais de um sistema operacional	9	5	5	5
19	É possível acessar suas funcionalidades a qualquer momento	10	5	5	5
MÉDIA PONDERADA			2,99	2,97	3,37

Fonte: Próprio autor.

Diante da avaliação das ferramentas perante cada requisito, a ferramenta que se mostrou mais adequada com a proposta de adequação do processo do laboratório foi o software Wrike. Suas utilidades mais próximas à gestão Ágil favoreceram-no neste caso.

Todavia, na avaliação dos requisitos obrigatórios (peso dez), nenhuma das ferramentas atendeu as expectativas de algumas necessidades. Os itens quatro e dez não foram atendidos por nenhuma das quatro aplicações e o item sete não foi atendido pelo próprio Wrike. Ou seja, o software vencedor se destacou das demais em vários quesitos com peso menor, porém, em elementos obrigatórios o mesmo foi inferior.

A avaliação diagnosticou que nenhuma ferramenta está preparada para atender o processo alternativo. Portanto, devido à sua capacidade de manipulação por ser código aberto, foi optado por uma adaptação ao software Redmine. Esta ferramenta poderia receber uma nota maior devido a este ponto, mas a possibilidade de customização não foi avaliada.

6.2 EXECUÇÃO DA SOLUÇÃO

Alguns plugins para o software Redmine foram selecionados a fim de adequarem-se ao processo. Devido ao fluxo propor a gestão ágil com motivações gamificadas no decorrer das atividades, foi necessário realizar a pesquisa de artifícios que proporcionam este ambiente e que sejam adequados com a plataforma. Assim sendo, o detalhamento da execução da ferramenta selecionada para gerir o processo está descrita nos capítulos seguintes.

6.2.1 Extensões

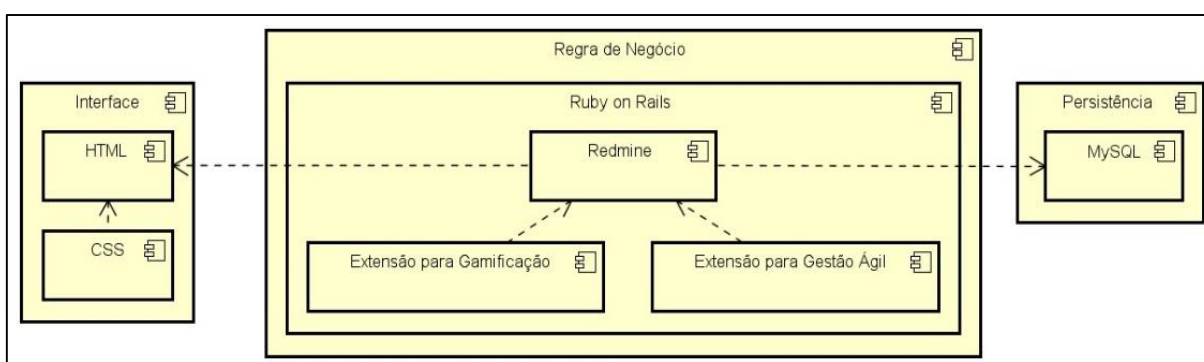
O Redmine, por ser software livre e disponível publicamente, é incrementado todos os dias por usuários ao redor do globo. A aplicação oferece a possibilidade de incrementos do programa por meio de extensões ou por integração via API.

Neste trabalho, é proposto o uso de dois plugins trabalham que integrados diretamente ao código do Redmine, a partir dos métodos de integração com as tarefas, projetos e demais funcionalidades do sistema. Também é indicado que não seja alterado o código principal do Redmine, apenas evoluí-lo com as ideias propostas pelos desenvolvedores. Na instalação, o Ruby oferece a possibilidade de “migrar” o código do plugin para a aplicação principal, portanto, incluindo as chamadas para as telas da extensão e a criação das novas tabelas no banco de dados.

Diante do cenário proposto, uma aplicação que permita uma gestão ágil e visual das tarefas é crucial para a aplicação do processo. As métricas citadas durante a avaliação também foram consultadas neste momento.

Por tratarem-se de aplicações independentes, não há uma integração direta entre as extensões. Durante a análise foram simuladas algumas situações baseado em um ambiente real e testado o uso da aplicação. Visto que o uso de ambas em paralelo é possível sem readequar alguma fase do processo. A estrutura representativa desta integração está definida no centro da Figura 19. O Redmine disponibiliza a plataforma para uso dos dois *plugins*, que oferecem as funcionalidades para aplicação do processo. A ferramenta opera em ambiente Web e respeita a arquitetura conforme a Figura 19. Cada extensão está descrita nos itens a seguir.

Figura 19: Arquitetura Lógica e Física



Fonte: Próprio autor.

a) *Agile Light*

A extensão melhor avaliada para a adequação à metodologia ágil foi a aplicação disponibilizada gratuitamente pelo grupo RedmineUP⁶. O *plugin* conta com todas as técnicas essenciais para uma eficiente aplicação do método. Possibilita a criação do Backlog do Produto, permitindo o armazenamento de todos os requisitos a serem atendidos no projeto. Atende o controle de situações de cada tarefa e a apresentação por meio de um Taskboard.

A ferramenta possibilita outras formas de análise, por exemplo, o gráfico Burndown para análise de desempenho. Além disso, é possível aplicar medições de diferentes formas para cada tarefa, utilizando Scrum Points e prioridades.

Permite armazenar as atividades aplicadas a cada necessidade, viabilizando no futuro uma análise de desempenho que pode ser feita em conjunto com o ranking das atividades gamificadas. Sua aderência com o Scrum em conjunto com as funcionalidades existentes do Redmine encaixa com a proposta deste trabalho.

⁶ Link: <https://www.redmineup.com/pages/plugins/agile>

b) Gamification Plugin

A extensão que propicia a gamificação dentro do Redmine foi feita pelo desenvolvedor Kazuto Horita, que disponibilizou seu código no GitHub⁷, sem fins lucrativos. Seu *plugin* permite a atribuição de pontos ao usuário por tarefa, por edição das tarefas e por atualizações na Wiki do projeto (em certas demandas, isto pode tornar-se muito útil principalmente no laboratório, em que o fluxo de colaboradores é alto). Além disso, é possível qualificar cada indivíduo participante do projeto externamente às tarefas, uma forma de elogio pelo trabalho executado no projeto. Todas estas informações são apresentadas em forma de ranking. Esta demonstração inclusive encontra-se em uma das etapas do processo proposto, comprovando a conexão entre a extensão e a integração tradicional-ágil em pauta.

6.2.2 Instalação, Implantação e Treinamento

Conforme salientado na explicação da arquitetura lógica e física, o sistema opera a partir de um ambiente web. Portanto, para visualização das interfaces e interação com a aplicação, é necessário um navegador de preferência do usuário.

Neste projeto o Redmine foi aplicado a partir de uma solução disponibilizada pela organização Bitnami⁸, denominado de “Bitnami Redmine Stack”. O formato disponibilizado oferece a instalação semelhante a qualquer software Windows, ou seja basta selecionar o diretório de destino e o instalador faz todo o trabalho. Esta instalação contempla o uso do Apache (servidor Web), a aplicação Redmine, o banco de dados (MySQL) e a ferramenta auxiliar para interagir com a base de dados (phpMyAdmin).

Após a conclusão da instalação, o próximo passo é a integração dos plugins. Para isso, o Bitnami Redmine Stack oferece um console que permite a interação diretamente com o Ruby on Rails. A obtenção de cada um é feita de forma diferente. A extensão para gamificação está disponível diretamente no site do GitHub e é possível efetuar o download do arquivo compactado. A extensão para gestão ágil é enviada por e-mail pela RedmineUP. Nesta mensagem há o link que permite a descarga do arquivo compactado deste plugin.

A migração dos arquivos para o Redmine é feita através do console citado anteriormente. Os seguintes passos devem ser executados para que a integração seja concluída com êxito.

1. Os arquivos compactados devem estar presentes no diretório `apps\redmine\htdocs\plugins`;
2. Abrir o console do Bitnami Redmine Stack;

⁷ Link: https://github.com/mauricio-camayo/redmine_gamification_plugin

⁸ Link: <https://bitnami.com/stack/redmine>

3. Digitar o comando: *bundle install --without development test --no-deployment*;
4. Digitar o comando: *rake redmine:plugins:migrate RAILS_ENV=production*;
5. Parar o serviço do Bitnami Redmine Stack e iniciar o serviço novamente;
6. Abrir o Redmine localizado na própria máquina usando um navegador. Digitar o endereço: *localhost/redmine*.
7. O sistema irá realizar a migração dos plugins e o Redmine levará um breve período para finalizar o carregamento. Está pronto.

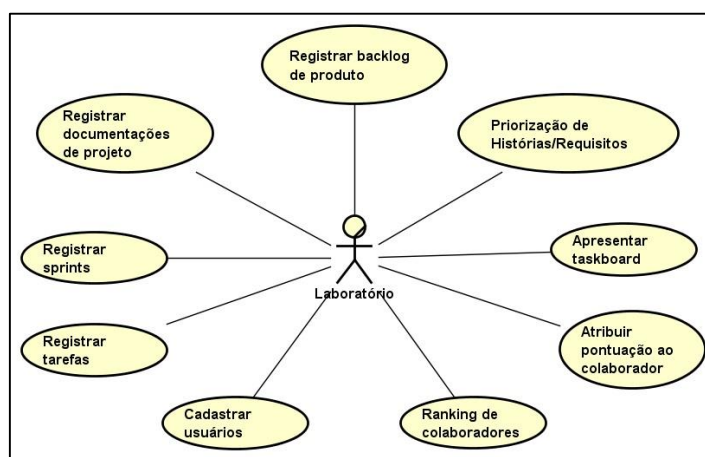
Com a ferramenta instalada e disponível para uso, basta que a rede permita localizar o servidor do Redmine e com uso a conexão das estações de trabalho.

Os treinamentos devem ser realizados em duas etapas: os acessos da gestão e os acessos dos alunos. Os gestores devem receber o treinamento de todo o sistema, incluindo as configurações gerais do Redmine, as parametrizações dos plugins, relatórios e as funcionalidades usadas pelos alunos. Os colaboradores da equipe devem ser capacitados apenas no que é de sua competência. Relatórios gerenciais e controles de versão, por exemplo, não são incumbências de alunos. Devido a ser uma ferramenta simples e que possui um guia amplo de uso, estima-se que a primeira etapa do treinamento leva em torno de cinco horas, incluindo os testes de uso. A fase que aplica-se aos alunos é mais curta. Duas horas seriam suficientes para explicar as funcionalidades que são de sua alçada e acompanhamento dos testes de usabilidade.

6.2.3 Funcionalidades

A ferramenta adaptada deve contemplar os seguintes requisitos, conforme a Figura 20. No decorrer da seção está explicado como operar cada determinação.

Figura 20: Funcionalidades



Fonte: Próprio autor.

6.2.3.1 Cadastrar usuários

Cada colaborador deve ter acesso à ferramenta por um usuário próprio. As tarefas são atreladas a um participante do projeto. Dentro da ferramenta há controle de múltiplos projetos e cada projeto terá seus membros em específico. Ou seja, certos usuários podem ter acesso a mais de um projeto simultaneamente, caso façam parte do mesmo.

Para isso, faz-se necessário a criação do usuário e atrelá-lo ao projeto. O passo a passo está descrito a seguir:

1. Abrir o cadastro de usuários: para cadastrar um usuário no Redmine, deve-se localizar a Administração da aplicação e selecionar a opção Utilizadores, conforme Figura 21.

Figura 21: Cadastrar usuários - Administração do Redmine



Fonte: Próprio autor.

2. Novo utilizador: No menu Utilizadores, localize a opção “Novo Utilizador”. A aplicação disponibilizará a tela de acordo com a Figura 22. Neste momento, os dados de acesso para o colaborador devem ser preenchidos conforme a necessidade.

- Vincular usuário ao projeto: abrir o painel dos projetos e localizar o trabalho cujo colaborador deve iniciar sua participação. Ver a Figura 23. Após selecionar o projeto, localize a guia Configurações – aba Membros. Nesta opção, clicar no botão “Novo membro”. O Redmine apresenta a tela conforme a Figura 24. Marque os usuários participantes deste projeto.

Figura 22: Cadastrar usuários - Novo utilizador

Utilizadores » Novo utilizador

Informações

Nome de utilizador *

Nome *

Apelido *

E-mail *

Língua Portuguese (Português)

Administrador

Autenticação

Palavra-chave *

Deve ter pelo menos 8 caracteres.

Confirmação *

Gerar palavra-chave

Deve alterar a palavra-chave no próximo início de sessão

Enviar dados da conta para o utilizador

Notificações por e-mail

Apenas para tarefas que observo ou em que estou envolvido

Não quero ser notificado de alterações feitas por mim

Preferências

Esconder endereço de e-mail

Fuso horário

Mostrar comentários Em ordem cronológica

Avisar-me quando deixar uma página com texto por guardar

Fonte para áreas de texto Default font

Fonte: Próprio autor.

Figura 23: Cadastrar usuários - Vincular usuário ao projeto - Administração



Fonte: Próprio autor.

Figura 24: Cadastrar usuários - Vincular usuário ao projeto - Novo membro

Novo membro

Procurar utilizador ou grupo:

aluno aluno

Matheus Concatto concatto

UserName LastName Admin

Utilizadores não membros

Utilizadores anônimos

Fonte: Próprio autor.

6.2.3.2 Registrar tarefas

O registro das tarefas de maneira correta dentro da aplicação é crucial para o funcionamento como um todo. Alguns dos demais requisitos dependem deste cadastro para garantir sua plena operação.

As tarefas contemplam o que deverá ser desenvolvido por um ou mais indivíduos da equipe designada para o projeto. Pode contemplar correções de erros, desenvolvimento de funcionalidades ou apenas gerir demandas do suporte que eventualmente poderão ocorrer. Além disso, os professores devem encarregar-se de monitorar o uso da ferramenta e estes registros, para garantir que está sendo usada de forma efetiva.

O registro de tarefas é feito por projeto, portanto, deve-se entrar no mesmo menu que foi apresentado no requisito anterior a partir da Figura 23.

Dentro do projeto, localizar a guia Tarefas – botão Nova Tarefa. O sistema irá apresentar os campos segundo a Figura 25. Cada campo possui o significado a seguir.

1. Tipo: pode ser classificado entre Bug, Funcionalidade e Suporte;
2. Assunto: breve título da demanda a ser atendida nesta tarefa;
3. Descrição: detalhamento da necessidade a resolver na tarefa. Deve ser o mais claro e conciso possível para evitar o retrabalho do responsável pela mesma. É importante lembrar que neste momento não devem haver mais dúvidas sobre o que deve ser feito, portanto, o esclarecimento eficiente com o cliente é essencial;
4. Estado: pode assumir os estados a seguir:
 - a. Novo: a tarefa foi criada e ainda não foi iniciada;
 - b. Em curso: a tarefa está em andamento;
 - c. Resolvido: a tarefa está concluída e necessita de testes;
 - d. Feedback: a tarefa necessita de um feedback do usuário;
 - e. Fechado: a tarefa foi testada, validada e já está pronta;
 - f. Rejeitado: a tarefa não foi executada.
5. Prioridade: pode receber os valores Baixa, Normal, Alta, Urgente e Imediata. Cabe ao analista que incluiu as tarefas designar a prioridade de um baseado no Product Backlog priorizado;
6. Atribuído a: responsável pelo desenvolvimento da tarefa;
7. Versão: Ciclo de desenvolvimento da tarefa. Na criação de uma tarefa no início de projeto, lembrar que faz parte do Backlog do Produto;
8. Ficheiros: anexos pertinentes à tarefa;
9. Observações: indivíduos do projeto interessados na conclusão desta tarefa (por exemplo, o professor);
10. Tarefa principal: preenchido caso esta tarefa faça parte de uma demanda maior, com um conjunto de tarefas;
11. Data de Início e Fim: início da tarefa e previsão de conclusão;

12. Tempo estimado: horas estimadas para conclusão;
13. % Completo: Percentual de conclusão;
14. Story Points: Pontos desta funcionalidade. São levantados a partir da priorização do backlog.

Figura 25: Registrar tarefas - Cadastrar tarefa

The screenshot shows a web form for creating a new task in the LASIS UCS system. The form is titled 'Nova tarefa' and is part of a navigation menu that includes 'Visão geral', 'Atividade', 'Planejamento', 'Tarefas', 'Tempo gasto', 'Cont', 'Agile', 'Calendário', 'Notícias', 'Documentos', 'Wiki', 'Fóruns', and 'Configurações'. The form is divided into several sections:

- Tipo:** A dropdown menu set to 'Funcionalidade'.
- Assunto:** A text input field.
- Descrição:** A large text area for describing the task.
- Estado:** A dropdown menu set to 'Nova'.
- Prioridade:** A dropdown menu set to 'Normal'.
- Atribuído a:** A dropdown menu for assigning the task to a user.
- Verdade:** A dropdown menu set to 'Verdade'.
- Tarefa principal:** A dropdown menu.
- Data de início:** A date input field set to '09/10/2017'.
- Data de fim:** A date input field set to '01/01/2018'.
- Tempo estimado:** A text input field.
- % Completo:** A text input field set to '0 %'.
- Story points:** A text input field.

At the bottom of the form, there are sections for 'Filtros' (with a dropdown set to 'Escolher qualquer') and 'Observações' (with a dropdown set to 'Nenhuma'). A note below the observations section reads: 'Necessário por observações para solução'.

Fonte: Próprio autor.

Para cada tarefa, o colaborador também deve registrar o tempo usado para seu desenvolvimento. Futuramente em relatórios é possível diagnosticar o tempo utilizado para todo o projeto e comparar com o tempo estimado.

O Redmine oferece a possibilidade de preenchimento das horas utilizadas por tarefa a partir dos botões em destaque, conforme a Figura 26.

Figura 26: Registrar tarefas - Tempo de trabalho

Funcionalidade #6

Demonstração da Tarefa
Adicionado por aluno aluno há 1 minuto atrás. Alterado há menos de 1 minuto atrás.

Estado: Novo
Prioridade: Imediata
Atribuído a: aluno aluno

Data de início: 02/11/2017
Data de fim:
% Completo: 100%
Tempo estimado: 2,50 h
Tempo gasto:

Story points: 20

Sub-tarefa

Tarefas relacionadas

Adicionar
Adicionar

Historico
Atualizado por aluno aluno há menos de 1 minuto
* % Completo alterado de 0 para 100
Finalizada tarefa.

#1

Tempo de trabalho

Tempo de trabalho

Também disponível em: Atom | PDF

Fonte: Próprio autor.

A partir dos botões, o sistema apresentará a tela conforme a Figura 28. É permitido informar o tempo apenas em horas, deve-se selecionar entre Desenvolvimento e Planejamento e o comentário é opcional. Ao utilizar o botão “Editar”, demonstrado com destaque na Figura 26, é possível incluir uma Nota que estará vinculado ao histórico de alteração desta tarefa. Este campo está demonstrado na Figura 27.

Figura 27: Registrar tarefas - Incluir tempo

Tempo de trabalho

Tempo gasto Horas

Atividade --- Selecione --- ▾

Comentário

Notas

B I U ~~S~~ C H1 H2 H3 [Ícones de formatação]

Notas privadas

Fonte: Próprio autor.

6.2.3.3 Registrar backlog de produto

O Redmine permite criar o backlog do produto em conjunto com as sprints a serem desenvolvidas no decorrer do projeto. Todas as tarefas devem ser criadas anteriormente e definidas para que estejam presentes na versão Backlog do Produto. Para que isto seja possível, deve-se seguir os passos descritos a seguir.

1. Criar versionamento para Backlog do Produto: o Redmine oferece o controle dos sprints de desenvolvimento por meio de versões. Antes de participarem de um ciclo, as tarefas devem estar presentes em uma versão “zero” chamada Backlog do Produto. Para criar a versão, localize a página do projeto conforme a Figura 23. Após, localize a opção Nova Versão, de acordo com a Figura 28. Na tela em destaque, deve-se efetuar a criação do Backlog do Produto com o nome conforme desejado pelo Administrador. Uma demonstração está apresentada na Figura 29.
2. Incluir tarefas no Backlog: ao incluir uma tarefa, esta demanda deve ser classificada na versão criada a partir do item um deste capítulo. Assim, nas

- etapas seguintes estará visível para o planejamento do desenvolvimento do projeto. Ver capítulo 6.2.3.2.
3. Consultar Backlog do Produto: Para verificar todas as tarefas que estão vinculadas ao Backlog, a extensão para o uso do *Agile* oferece o menu *Version Planning*, que apresenta um taskboard com as tarefas e seu estado atual. A Figura 30 demonstra este item de forma clara. Deve-se localizar o menu “*Agile*”, dentro do projeto em questão. Na barra lateral direita, a opção “*Version Planning*” deve ser selecionada para guiar ao próximo menu.
 4. *Version Planning*: o *taskboard* apresentado na tela permite a mudança de tarefas entre versões, portanto, nesta tela também será feito o controle de tarefas por Sprint. Uma coluna demonstrará todas as tarefas presentes no backlog do produto, que serão distribuídas no andamento do trabalho. Consulte a Figura 31 para maiores detalhes.

Figura 28: Registrar backlog de produto - Nova versão



Fonte: Próprio autor.

Figura 29: Registrar backlog de produto: Confirmar nova versão

A imagem mostra o formulário 'Nova versão' no sistema LASIS UCS. O formulário contém os seguintes campos e opções:

- Nome ***: Backlog do Produto
- Descrição**: Concentra todas as tarefas para desenvolvimento no projeto.
- Página da Wiki**: (campo vazio)
- Data**: 05/11/2017
- Partilha**: Com árvore do projeto (menu suspenso)
- Versão por defeito**:

Um botão 'Criar' está localizado na base do formulário.

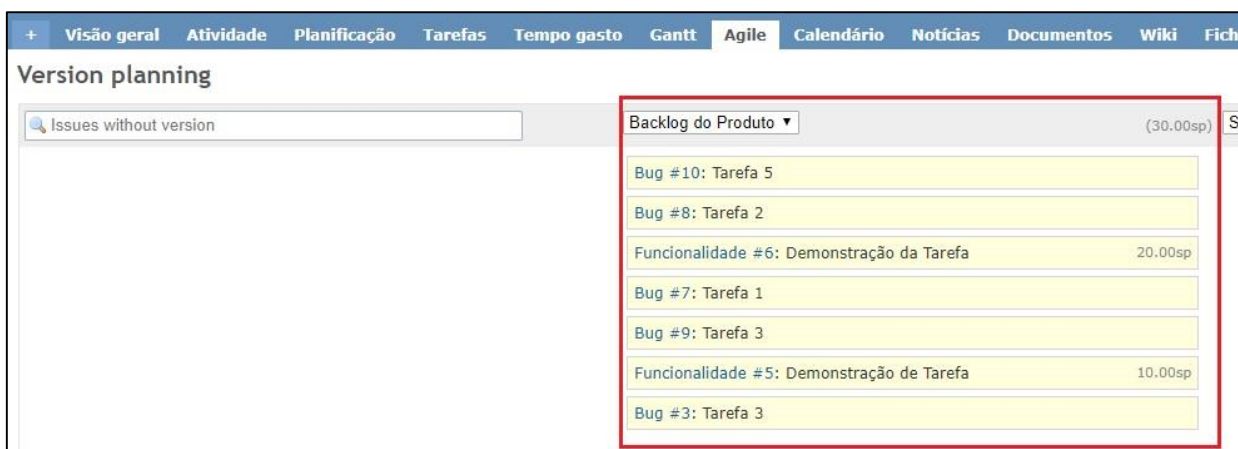
Fonte: Próprio autor.

Figura 30: Registrar backlog de produto - Menu Version Planning



Fonte: Próprio autor.

Figura 31: Registrar backlog de produto: Taskboard Version Planning



Fonte: Próprio autor.

6.2.3.4 Registrar sprints

Para registrar as Sprints a serem realizadas durante o projeto pelos colaboradores, o Redmine em conjunto com a extensão para metodologia ágil oferece a criação de Versões.

Cada versão deve ser criada conforme informado no capítulo 6.2.3.3, nomeando-as conforme o número do ciclo. Após isto, ao localizar o menu de controle de versões (Figura 30), é possível mover a tarefa cadastrada no Backlog do Produto para a sua Sprint correspondente.

Em cada reunião de planejamento da Sprint é importante definir o total de pontos a serem atendidos durante o período. A ferramenta oferece o controle mostrando o total de pontos de cada tarefa diretamente no taskboard e a soma dos pontos de todas as tarefas em cada Sprint. A Figura 32 mostra de forma mais clara este raciocínio. Os pontos são definidos para cada tarefa no campo *Story Points* (ver seção 6.2.3.2).

Figura 32: Registrar sprints - Taskboard

Sprint 1	(30.00sp)
Funcionalidade #6: Demonstração da Tarefa	20.00sp
Funcionalidade #5: Demonstração de Tarefa	10.00sp

Fonte: Próprio autor.

6.2.3.5 Priorização de histórias/requisitos

A priorização deve ser realizada por meio dos Story Points, registrados para cada tarefa. Cabe à análise do professor e da equipe para detalhar a complexidade de cada demanda e atribuir a pontuação adequada. Esta relação de pontos pode ser consultada por meio do Taskboard Version Planning (Figura 32).

6.2.3.6 Apresentar taskboard

A apresentação da situação das tarefas no atual Sprint deve ser a mais clara possível. Desta maneira, dentro dos artefatos do Scrum determina-se o uso de um

Taskboard que demonstre as tarefas por situação. A ferramenta deve oferecer esta funcionalidade, que no Redmine é denominada *Agile Board*. Ver a Figura 33.

Figura 33: Apresentar taskboard - Taskboard



Fonte: Próprio autor.

O Agile Board oferecido pela extensão do Redmine propõe quatro situações a serem mostradas para as tarefas. Além disso, a ferramenta permite mover as tarefas entre situações no próprio quadro, tornando-a sua usabilidade muito mais simples e eficaz. Ao clicar na tarefa, o sistema mostra a tela de detalhes da tarefa (ver Figura 26).

Para localizar o *Taskboard* na aplicação, deve-se entrar no projeto (Figura 23) e localizar a guia *Agile*.

6.2.3.7 Atribuir pontuação ao colaborador

Diante da metodologia proposta, o colaborador recebe a pontuação de acordo com as tarefas que executa. O ranking é atualizado após o retorno das Sprints. Na extensão para o Redmine usada neste projeto, além das tarefas o colaborador pode receber incentivos a partir de outras situações, tais como o preenchimento da ajuda do sistema. Além disso, a proposta de evolução de níveis também é atendida por este *plugin*, como é mostrado a seguir, na Figura 36.

Para acessar as configurações do plugin e verificar as possibilidades oferecidas pelo mesmo, deve-se abrir o Administrador do Redmine e localizar a opção Extensões, conforme Figura 34. Na opção Extensões, apresenta-se os *plugins* auxiliares do Redmine, já descritos na seção 6.2. Para configurar a extensão para gamificação, deve-se selecionar a opção “Configurar”, conforme a Figura 30.

A extensão oferece as seguintes configurações para proporcionar a evolução do usuário dentro da ferramenta.

1. *First level point*: número de pontos para alcançar o primeiro nível;
2. *Level multiplier*: taxa de evolução dos níveis na aplicação;
3. *New issue*: pontuação recebida pelo colaborador por criar uma tarefa;
4. *Edit issue*: pontuação recebida pelo colaborador por editar uma tarefa;
5. *Edit wiki*: pontuação recebida pelo colaborador por colaborar com a ajuda do projeto;
6. *Set Score for status*: pontuação recebida pelo colaborador por modificar o estado da tarefa. É possível pontuar caso a tarefa evolua para a situação posterior ou retroceda;
7. *Set score for tracker*: pontuação por tipo de tarefa;
8. *Closing points*: pontuação para o usuário que fechou a tarefa;

Figura 34: Atribuir pontuação ao colaborador - Administração do Redmine



Fonte: Próprio autor.

Figura 35: Atribuir pontuação ao colaborador - Acessar configurações



Fonte: Próprio autor.

Figura 36: Atribuir pontuação ao colaborador - Configurações

Extensões » Gamification plugin

Give points to previous user or the new user
 Give points to previous user
 When changing status AND user at the same time, shall we assign the points to the previous user or the new user?

First level point
 First level point
 Number of points in order to get to the first level

Level multiplier
 Level multiplier
 Ration in which every level increases

New issue
 Use gamification plugin with New Issue
 Score for new issue:
 Score for new issue within a project:
 This two add up if the creator is an active member of the project

Edit issue
 Use gamification plugin with Edit issue
 Score for update issue:
 Score for update issue within a project:
 This two add up if the updater is an active member of the project

Edit Wiki
 Use gamification plugin with Edit Wiki
 Score for update Wiki:
 Score for update issue within a project:
 This two add up if the updater is an active member of the project

Set score for status:

Status name	Going Forward	Going Backward
Novo	<input type="text"/>	<input type="text"/>
Em curso	<input type="text"/>	<input type="text"/>
Resolvido	<input type="text"/>	<input type="text"/>
Feedback	<input type="text"/>	<input type="text"/>
Fechado	<input type="text"/>	<input type="text"/>
Rejeitado	<input type="text"/>	<input type="text"/>

Set score for tracker:

Tipo	Score
Bug	<input type="text" value="3"/>
Funcionalidade	<input type="text" value="5"/>
Suporte	<input type="text" value="3"/>

Closing points
 Closing point to updater
 When a user close (see status) an issue, this points go to the updater (not the assignee)

Fonte: Próprio autor.

Percebe-se a gama de variedades possíveis dentro do *plugin* para motivação dos usuários, além do desenvolvimento da própria tarefa. Um atrativo desta extensão é a pontuação para a atualização da ajuda do projeto (*Wiki*). Como já descrito anteriormente, a rotatividade de alunos dentro do laboratório é um problema conhecido e enfrentado pelo relatório. No momento que o projeto possui registros

relacionados ao mesmo e acesso compartilhado a todos, tempo pode ser economizado acessando estes arquivos ao invés de deslocar um professor para fazer estas explicações. Além disso, o aluno que contribuir para esta ideia também é recompensado com pontos, portanto, o benefício é mútuo. Esta pontuação é apresentada no *ranking* de colaboradores da ferramenta, que integra todos os projetos. Esta etapa está esclarecida no capítulo 6.2.3.8. Além disso, os professores podem atribuir elogios aos alunos sem relação com tarefas. Desta forma, os colaboradores que possuem uma eficiência menor mas que provam sua dedicação ao projeto também podem ser qualificados.

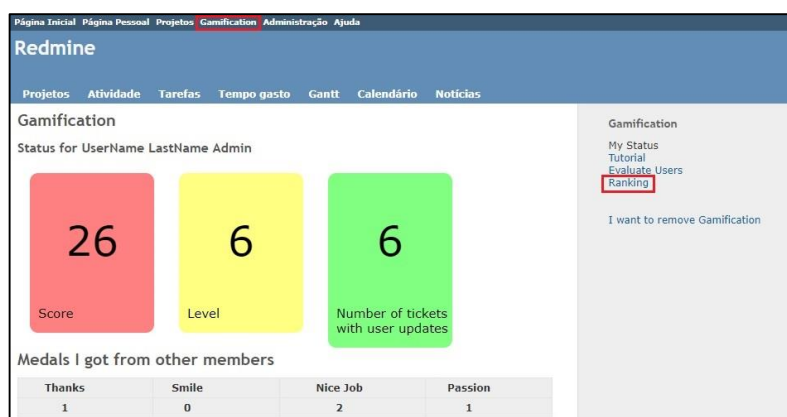
O único revés encontrado foi o idioma Inglês, que pode dificultar na interpretação da configuração do plugin. Conversando com os professores, isto não foi considerado um empecilho.

6.2.3.8 Ranking de colaboradores

A pontuação de todos os colaboradores é demonstrada através de um sistema de *ranking*. Esta lista demonstra além dos pontos, as gratificações dadas pelos outros usuários sem relação com tarefas, outra funcionalidade que promove a motivação presente na extensão para gamificação.

O ranking pode ser acessado através do menu superior *Gamification* e na opção *Ranking*, localizada à direita. Ver a Figura 37.

Figura 37: Ranking de colaboradores - Acessar ranking



Fonte: Próprio autor.

O ranking é apresentado em forma de tabela e usando a ordenação decrescente por score. Está demonstrado na Figura 38.

Figura 38: Ranking de colaboradores - Ranking

Ranking					
Ranking	Avatar	User name	Score	Level	Medals
1		UserName LastName Admin	26	6	4
2		aluno aluno	20	6	0
3		Anônimo	0	1	1
4		Matheus Concatto concatto	0	1	1

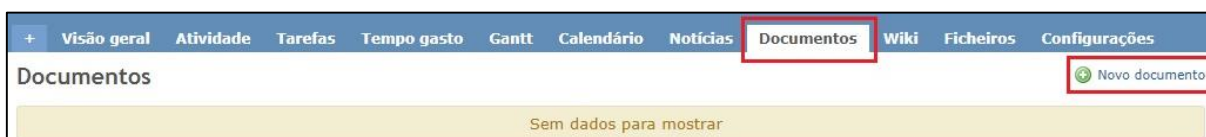
Fonte: Próprio autor.

6.2.3.9 Registrar documentações de projeto

O Redmine permite carregar os documentos por projeto para futura consulta. Além disso, a aplicação permite classificar os tipos de arquivos incluídos no projetos. Neste trabalho, vamos utilizar o padrão já existente denominado “Documentação Técnica”.

O primeiro passo é localizar a tela de gestão do projeto (Figura 23). Após isto, deve-se localizar a aba Documentos, conforme Figura 39. Pressione o botão “Novo Documento”.

Figura 39: Registrar documentações de projeto - Menu documentos



Fonte: Próprio autor.

O sistema irá apresentar a tela que permite caracterizar o arquivo a ser carregado e registrado no projeto. A interface está na Figura 40. Vale lembrar que o tipo deve ser escolhido como Documentação Técnica. Os campos Nome e Descrição devem ser utilizados para identificação de cada documento, portanto, ao importar o arquivo “Termo de Abertura do Projeto”, o documento deve estar denominado a caráter.

Vale lembrar que este requisito também é utilizado para carregar demais documentações pertinentes ao projeto, tais como o modelo da EAP, o primeiro registro das histórias e a documentação UML avulsa (está presente também no PGP).

Figura 40: Registrar documentações de projeto - Novo documento



Fonte: Próprio Autor.

6.2.3.10 Considerações finais

Na seção 6.2 foram apresentados o detalhamento da implantação da solução, bem como os requisitos principais que a ferramenta deveria atender para o uso no laboratório. Contudo, alguns pontos não foram levantados até o momento.

Algumas fases do projeto não são supridas diretamente na ferramenta mas devem ser executadas da mesma forma. No que tange a fase de iniciação, o primeiro levantamento de requisitos/histórias não é registrado no software. Neste momento, o que foi identificado ainda não está completamente validado. Nas fases posteriores, ao transformar estas histórias em backlog de produto, este terá seu local específico dentro da ferramenta. Além disso, o Redmine não possui uma aplicação própria para elaboração da EAP, portanto, a mesma deverá ser composta por outro sistema e inserida nos documentos obrigatórios do projeto. A Estrutura Analítica de Projeto pode ser carregada conforme os passos do capítulo 6.2.3.9, caso seja de desejo do professor guardar este arquivo em um ficheiro isolado.

Na fase de planejamento diversas atividades pedem o uso de documentos auxiliares, gerados por softwares em paralelo. Por exemplo, a geração do MDP deve ser feita em uma outra aplicação que permita registrar as histórias e relacioná-las. No Redmine é possível definir tarefas pai das demais, contudo, não há uma forma visual de consultar todas estas relações. Além disso, todos os artefatos padrão UML também pedem uma ferramenta à parte, visto que o Redmine é o gerenciador do projeto e não contempla artifícios para análise e modelagem de software.

Durante a fase de execução, as mudanças identificadas devem ser registradas, mas não há um padrão de documentação para este caso. Independente da forma que as mudanças sejam catalogadas, em seguida devem ser registradas como histórico das tarefas a partir de notas, conforme demonstrado na Figura 27.

Encerrando o processo, vale lembrar que para o preenchimento de todos os documentos pertinentes ao projeto faz-se necessário o uso de um editor de texto

conforme a preferência do professor. O arquivo gerado deve ser carregado no Redmine conforme citado na seção 6.2.3.9.

6.2.4 Avaliação

Definida a ferramenta a ser utilizada e os requisitos que a mesma deve atender, cabe ao laboratório indicar se a proposta como um todo está adequada. Portanto, foi definido com a administração do laboratório que uma avaliação é necessária para um veredito final do trabalho proposto. Esta avaliação levantará dados de forma qualitativa, conforme a proposta inicial. Este parecer deve ser advindo de uma forma de levantamento de informações com embasamento teórico que comprova a sua eficiência. Neste caso, foi definido que a avaliação qualitativa seria mais adequada e não há necessidade da complexidade de levantamento de dados de forma quantitativa. Este parecer deve ser dado por um professor do laboratório, que julga o processo. A decorrência da avaliação determina a possibilidade de execução do método. Neste trabalho não foi feito uma avaliação utilizando um projeto piloto, contudo, este deve ser o próximo passo. A qualificação do fluxo será realizada a partir da avaliação.

Na modalidade escolhida há diversas formas de avaliação possíveis. GIL (2008) destaca a observação, questionário e entrevista como técnicas para levantamento de dados. Cada uma possui vantagens e desvantagens, portanto, é papel do idealizador da pesquisa identificar qual é o método que melhor se adequa ao seu propósito.

Diante deste contexto, a alternativa escolhida para realizar a amostragem e confirmar a eficiência do processo foi o questionário. Assim sendo, GIL (2008) define alguns pontos que são cruciais para a elaboração das perguntas, tanto na definição da própria pergunta como nas opções oferecidas como respostas. Examinando a literatura e possíveis maneiras de iniciar o desenvolvimento das questões, definiu-se que todas as perguntas deveriam ser de caráter fechado. Isto significa que o interrogado terá opções fixas de respostas, assim, facilitando a análise da informação. Caso queira, pode realizar um adendo à sua resposta inserindo uma observação, contudo, a alternativa escolhida é usada para fins de análise.

A avaliação deve ser precisa, visto que qualquer inconformidade dependendo de sua complexidade pode determinar a mudança do processo. BROOKE et al. (1996) elaborou um modelo que é utilizado até hoje para qualificar softwares em torno do mundo, chamado de SUS (System Usability Scale). O autor elaborou as perguntas baseando-se na ISO 9241-11, que define três critérios principais para medição: eficácia, eficiência e satisfação (ISO, 1998). Neste contexto, algumas questões criadas por Brooke foram utilizadas como base e outras foram determinadas pelo autor deste projeto, para definir se o processo atende a necessidade do laboratório. As perguntas foram divididas pelas fases orientadas no capítulo 5.3. A Tabela 2

demonstra o questionário. Além disso, todas as questões são explicadas nas seções seguintes conforme a fase em destaque. Cada pergunta possui a sua importância. O método SUS indica uma forma de calcular o resultado final, contudo, neste caso será optado por uma forma de avaliação diferente.

Tabela 2: Questionário

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
Iniciação					
O processo pode ser respeitado pelo laboratório.					
O processo não é desnecessariamente complexo.					
Todas as atividades do processo são necessárias para a conformidade do todo.					
O processo pode ser entendido rapidamente.					
O termo de abertura de projeto é adequado para seu propósito.					
O processo atende a proposta da fase de iniciação.					
Planejamento					
O processo pode ser respeitado pelo laboratório.					
O processo pode ser respeitado pelo laboratório.					
O processo não é desnecessariamente complexo.					

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
Todas as atividades do processo são necessárias para a conformidade do todo.					
O processo pode ser entendido rapidamente.					
O plano de gerenciamento de projeto é adequado para seu propósito.					
O plano de gerenciamento de custos é adequado para seu propósito.					
A integração entre o PMBOK e o Scrum é perceptível.					
A integração entre o PMBOK e o Scrum é adequada.					
O processo atende a proposta da fase de planejamento.					
Execução					
O processo pode ser respeitado pelo laboratório.					
O processo não é desnecessariamente complexo.					
Todas as atividades do processo são necessárias para a conformidade do todo.					
O processo pode ser entendido rapidamente.					

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
A integração entre o PMBOK e o Scrum é perceptível.					
A integração entre o PMBOK e o Scrum é adequada.					
As etapas do Scrum estão bem definidas e podem ser executadas pelo laboratório.					
Todas as cerimônias propostas podem ser executadas pelo laboratório.					
A proposta de gamificação é perceptível.					
A proposta de gamificação é adequada.					
Os alunos serão engajados a participar do laboratório.					
O uso de um método ágil é bem aceito pelo laboratório.					
O processo atende a proposta da fase de execução.					
Encerramento					
O processo pode ser respeitado pelo laboratório.					
O processo não é desnecessariamente complexo.					

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
Todas as atividades do processo são necessárias para a conformidade do todo.					
O processo pode ser entendido rapidamente.					
O termo de encerramento de projeto é adequado para seu propósito.					
O processo atende a proposta da fase de encerramento.					

Fonte: Próprio autor.

Nos próximos capítulos, estão detalhados os resultados da avaliação proposta para cada fase do método.

6.2.4.1 Iniciação

A Tabela 3 detalha os resultados do questionário para a etapa de Iniciação.

Tabela 3: Fase de iniciação

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo pode ser respeitado pelo laboratório.					✓

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo não é desnecessariamente complexo.				✓	
Todas as atividades do processo são necessárias para a conformidade do todo.				✓	
O processo pode ser entendido rapidamente.					✓
O termo de abertura de projeto é adequado para seu propósito.					✓
O processo atende a proposta da fase de iniciação.					✓

Fonte: Próprio autor.

6.2.4.2 Planejamento

4. A avaliação da fase de planejamento resultou nas respostas conforma a Tabela 4.

Tabela 4: Fase de planejamento

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo pode ser respeitado pelo laboratório.					✓

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo pode ser respeitado pelo laboratório.				✓	
O processo não é desnecessariamente complexo.					✓
Todas as atividades do processo são necessárias para a conformidade do todo.					✓
O processo pode ser entendido rapidamente.					✓
O plano de gerenciamento de projeto é adequado para seu propósito.					✓
O plano de gerenciamento de custos é adequado para seu propósito.				✓	
A integração entre o PMBOK e o Scrum é perceptível.					✓
A integração entre o PMBOK e o Scrum é adequada.					✓
O processo atende a proposta da fase de planejamento.					✓

Fonte: Próprio autor.

6.2.4.3 Execução

A fase de execução contempla a maior parte das questões deste questionário. As conclusões estão na Tabela 5.

Tabela 5: Fase de execução

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo pode ser respeitado pelo laboratório.					✓
O processo não é desnecessariamente complexo.					✓
Todas as atividades do processo são necessárias para a conformidade do todo.					✓
O processo pode ser entendido rapidamente.				✓	
A integração entre o PMBOK e o Scrum é perceptível.					✓
A integração entre o PMBOK e o Scrum é adequada.					✓
As etapas do Scrum estão bem definidas e podem ser executadas pelo laboratório.					✓
Todas as cerimônias propostas podem ser executadas pelo laboratório.					✓

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
A proposta de gamificação é perceptível.				✓	
A proposta de gamificação é adequada.				✓	
Os alunos serão engajados a participar do laboratório.				✓	
O uso de um método ágil é bem aceito pelo laboratório.					✓
O processo atende a proposta da fase de execução.					✓

Fonte: Próprio autor.

6.2.4.4 Encerramento

O encerramento do projeto, mesmo sendo a fase com menos atividades a executar, também foi avaliado aos moldes dos demais. A Tabela 6 aponta o desfecho da pesquisa.

Tabela 6: Fase de encerramento

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo pode ser respeitado pelo laboratório.					✓

Questionário					
	Discordo plenamente	Discordo parcialmente	Não concordo nem discordo	Concordo parcialmente	Concordo plenamente
	1	2	3	4	5
O processo não é desnecessariamente complexo.					✓
Todas as atividades do processo são necessárias para a conformidade do todo.					✓
O processo pode ser entendido rapidamente.					✓
O termo de encerramento de projeto é adequado para seu propósito.					✓
O processo atende a proposta da fase de encerramento.					✓

Fonte: Próprio autor.

O veredito final avaliou que é possível aplicar este processo no laboratório. As avaliações de todas as perguntas do questionário retornaram resultados positivos, portanto, a análise foi feita de forma correta e todas as fases podem ser executadas com êxito. Nas questões respondidas de modo “Concordo Parcialmente”, devem ser levantados os motivos da nota final não alcançar a pontuação máxima. Contudo, esta opção determina concordância, sendo assim, o processo permanece adequado.

Além disso, o processo propôs o uso de artefatos baseados em métodos tradicionais mas que não tornou o processo burocrático e enrijecido. Todos os documentos propostos são necessários para o controle do projeto, tanto na gestão como no desenvolvimento.

7 CONCLUSÃO

O presente trabalho conclui que o processo proposto com o suporte da ferramenta foi adequado ao laboratório. Para alcançar este veredito, foi necessário realizar pesquisas em diversas áreas e elaborar a colaboração entre elas, neste caso, as gerências de projetos tradicional e ágil, além da gamificação.

Nas pesquisas foi diagnosticado que há possibilidade de integração entre metodologia tradicional e ágil. O meio termo entre elas pode existir, de modo que nenhum controle fique sucumbido. Não há uma metodologia mais eficiente que a outra em todos os casos. O gerente de projeto deve entender o objetivo final, as variáveis que envolvem um projeto e com isso, tomar a melhor decisão. Dependendo da situação o método tradicional será mais eficaz. Todavia, em outro ambiente, a agilidade será a melhor opção. No caso do laboratório, foi possível unir os conceitos indicados por ambas (tradicional e ágil) para elaborar um processo único.

Com a integração entre os métodos definida, foi possível concluir este trabalho e atingir os objetivos. A análise do processo do laboratório foi essencial para que a proposta apresentada fosse condizente com a realidade, portanto, aplicável. Os estudos também ofereceram controles que não eram realizados antes e que enriqueceram o controle, tais como os planos de planejamento e encerramento do projeto. Além disso, As informações pertinentes ao trabalho dificilmente serão perdidas, visto que estão armazenadas na ferramenta. Este controle, advindo do método tradicional, supriu perfeitamente as práticas do Scrum propostas na fase de Execução do processo.

Durante a fase de execução e encerramento, o uso dos artifícios propostos pela gamificação também foram adequados com os objetivos de buscar novos colaboradores para o laboratório e engajar os atuais. Com o novo processo, os alunos podem consultar seu desempenho no laboratório e evoluir seu nível. A análise de perfis realizada, que identificou o interesse dos alunos por jogos, tornou o uso desta ideia ainda mais viável. Os alunos possuem sua motivação intrínseca de procurar o novo, com o adendo de ser recompensado pelo seu desempenho. Vale lembrar que, no momento que o processo esteja em plena execução, a rotatividade de alunos salientada durante o problema de pesquisa também será minimizado. Entende-se que os alunos terão motivação de manter-se aliados ao laboratório. O uso da gamificação durante a aplicação do processo é coerente.

Na execução das quatro fases do processo, é imprescindível que uma ferramenta seja capaz de administrar todas as atividades a serem realizadas pelos alunos e professores. A partir disto, a avaliação realizada neste trabalho nas ferramentas oferecidas pelo mercado deveria ser crítica. Nesta análise, identificou-se que os pontos cruciais do processo não foram atendidos. Contudo, a grande capacidade de adaptação do Redmine permitiu que este processo fosse suprido integralmente, de maneira que o desenvolvimento não foi necessário.

O Redmine, por ser software *open-source*, possui aprimoramento contínuo e as evoluções desenvolvidas pelos usuários ao redor do planeta supriram este processo. O plugin para uso da metodologia ágil ofereceu todas as funcionalidades necessárias, conforme foi citado durante a execução da solução. Além disso, seu suporte parcial para a língua portuguesa foi um bônus útil, visto que alguns alunos podem não ter o domínio do idioma inglês. A extensão para gamificação proporcionou a pontuação por tarefa e a geração do ranking, indispensáveis para garantir o engajamento. Na elaboração dos requisitos para validação da aplicação, o processo foi retomado e as principais funcionalidades foram levantadas a fim de provar a utilidade da ferramenta.

Cada fase do processo foi avaliada de modo qualitativo pelo professor responsável a partir de um questionário. Este método de avaliação foi usado por ser mais simples e retornar as respostas que eram necessárias na pesquisa. As perguntas foram elaboradas a fim de avaliar o processo como um todo, não apenas o suporte da ferramenta para o mesmo. Os resultados advindos do mesmo garantiram a aprovação da proposta pelo laboratório e mais um objetivo foi alcançado. O processo foi demonstrado a partir dos requisitos e não houve um contraponto do laboratório indicando que o processo não seria adequado.

Durante todo o projeto, as maiores dificuldades foram encontradas no período da pesquisa e execução do processo pela ferramenta. A elaboração do fluxo de atividades baseado em CRUZ (2013) e nas pesquisas realizadas proveram um processo alternativo para o laboratório que nem todo software suportaria. Assim, encontrar uma ferramenta e testar suas operações, conforme dito no capítulo 6.1, foi o ponto mais moroso deste projeto.

Após a definição do uso do Redmine, a instalação dos plugins no ambiente foi a segunda situação com dificuldades. A migração (termo usado pelo Redmine) das extensões para a aplicação em si passou por várias tentativas e erros até a conclusão de como a vinculação entre os componentes deveria funcionar. Contudo, após este aprendizado, as próximas instalações necessárias não apresentaram contratemplos.

Para trabalhos futuros, o uso deste processo em um projeto piloto reforçaria a eficiência do mesmo. A aplicação do fluxo, o uso da ferramenta e a elaboração de todos os artefatos de forma correta resultariam em uma avaliação mais precisa do processo e inclusive algumas propostas de melhoria. Além disso, a identificação dos alunos com as técnicas de gamificação proposta podem ser melhor ponderados simulando um ambiente real.

REFERÊNCIAS

ALVES, Fiora. **Gamification**: como criar experiências de aprendizagem engajadoras: um guia completo do conceito à prática. 1. ed. São Paulo: DVS Editora, 2014.

AMARAL, Daniel; CONFORTO, Edivaldo Carlos; BENASSI, João Luís Guilherme; ARAUJO, Camila de. **Gerenciamento de Projetos**: aplicação em produtos inovadores. São Paulo: Saraiva, 2011.

ARRUDA, Eucídio Pimenta. **Fundamentos para o desenvolvimento de jogos digitais**. Porto Alegre: Bookman, 2014.

ASTELS, David; GRANVILLE, Miller; NOVAK, Miroslav. **Practical Guide to eXtreme programming**. Upper Saddle River: Prentice Hall, 2002.

AUDY, Jorge Horário Nicolás. **Adaptação à Mudança nas Características do Trabalho**: Níveis de Demanda e Controle Durante a Adoção do Método Ágil SCRUM por Equipes de Desenvolvimento de Software. 2015. 148 p. Dissertação (Mestrado em Administração e Negócios) – Pontifícia Universidade Católica do Rio Grande de Sul, Programa de Mestrado em Administração e Negócio, 2015. Disponível em: <<http://tede2.pucrs.br/tede2/bitstream/tede/6178/2/470810%20-%20Texto%20Completo.pdf>>. Acesso em: 21 jun. 2017.

AUER, Ken; MILLER, Roy. **Extreme programming applied: playing to win**. Indianapolis: Pearson Education Inc, 2002.

BARTLE, Richard. **Hearts, clubs, diamonds, spades**: Players who suit muds. Colchester: MUSE Ltd, 1996.

BECK, Kent. **Programação extrema aplicada**: acolha as mudanças. Traduzido por: Adriana P.S. Machado e Natália N.P. Lopes. Porto Alegre: Bookman, 2004.

BOEING, Alexandre; SOUZA, Dieimes Nunes de. **Gerenciamento de Projetos de Software com Scrum e PMBOK®**. 2013. 92 p. Dissertação (Graduação em Análise e Desenvolvimento de Sistemas) – Faculdades Integradas do Vale do Ivaí, ICEI, 2013. Disponível em: <http://www.univale.com.br/unisite/documentos/publicacoes/gerenciamento_de_projetos_de_software_com_scrum_e_pmbok.pdf>. Acesso em: 20 jun. 2017.

BROOKE, John et al. **Usability Evaluation In Industry**. Londres: Taylor & Francis Inc., 1996

BURKE, Brian. **Gamificar**: como a gamificação motiva as pessoas a fazerem coisas extraordinárias. Traduzido por: Sieben Gruppe. São Paulo: DVS Editora, 2015.

CARVALHO, Bernardo Vasconcelos de. **Aplicação do método ágil Scrum no desenvolvimento de produtos de software em uma pequena empresa de base**

tecnológica. 2009. 100 p. Dissertação (Mestrado em Ciências em Engenharia de Produção) – Universidade Federal de Itajubá, UNIFEI, 2009.

CAVALCANTI, Francisco Rodrigo P. **Fundamentos de gestão de projetos: gestão de riscos**. São Paulo: Atlas, 2016.

CHIAVENATO, Idalberto. **Comportamento Organizacional: a dinâmica do sucesso nas organizações**. Barueri-SP: Manole, 2014.

COHN, Mike. **Desenvolvimento de Software com Scrum**. Traduzido por: Bookman. Pearson Education Inc. Porto Alegre: Bookman. 2011.

CONFORTO, Edivandro Carlos. **Gerenciamento ágil de projetos: proposta e avaliação de método para gestão de escopo e tempo**. 2009. 304 p. Dissertação (Mestrado em Engenharia de Produção) – Universidade de São Paulo, Escola de Engenharia de São Carlos, 2009.

CRISTAL, Maurício; WILDT, Daniel; PRIKLADNICKI, Rafael. Usage of SCRUM Practices within a Global Company. In: 2008 **IEEE International Conference on Global Software Engineering**. 2008, Bangalore, Índia. Anais... Bangalore, Índia: IEEE, 2008. p. 222-226. Disponível em: <<http://www.few.vu.nl/~hans/publications/y2012/ICGSE2012-RUPtoScrum/Agile%20and%20GSD/Cristal2008.pdf>> Acesso em: 15 jun. 2017

CRUZ, Fábio. **Scrum e Guia PMBOK® unidos no gerenciamento de projetos**. Rio de Janeiro: Brasport, 2013.

CSIKSZENTMIHALY, Mihaly. **Flow: The Psychology of Optimal Experience**. Nova Iorque: Harpercollins, 1990.

EUROPEAN CONFERENCE ON INFORMATION SYSTEMS, 21., 2013, Utrecht. **AIS Electronic Library (AISeL)**. Utrecht: Association For Information Systems, 2013. Disponível em: <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1348&context=ecis2013_cr>. Acesso em: 09 out. 2017.

EMERGING TRENDS. **Introduction to Gamification**. Princess Risborough: ILX, 2014.

FERREIRA, Luciana A. et al. Gamificação Aplicada ao Ensino de Gerência de Projetos de Software. In: V CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016, Uberlândia. **Anais do XXII Workshop de Informática na Escola**. Uberlândia: V Congresso Brasileiro de Informática na Educação, 2016. p. 151 - 160. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/6612/4523>>. Acesso em: 15 jun. 2017.

FOGGETTI, Cristiano. **Gestão Ágil de Projetos**. São Paulo: Education do Brasil, 2014.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Editora Atlas S.a, 2008.

JUHO HAMARI (Finlândia) (Org.). Why do people use gamification services? **International Journal Of Information Management**. Aalto, ago. 2015. p. 419-431. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0268401215000420>>. Acesso em: 01 jun. 2017.

HELDMAN, Kim. **Gerência de Projetos**. 5. ed. Rio de Janeiro: Elsevier, 2009.

HOUAISS, Antônio; VILLAR, Mauro de Salles; FRANCO, Francisco Manoel de Mello. **Minidicionário Houaiss da Língua Portuguesa**. Rio de Janeiro: Objetiva, 2009.

HUIZINGA, Johan. **Homo Ludens: O Jogo como Elemento da Cultura**. Traduzido por: João Paulo Monteiro. 4. Ed. São Paulo: Editora Perspectiva S.A., 1999.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 9126-1:2001**: Software engineering -- Product quality -- Part 1: Quality model. Genebra: International Electrotechnical Commission (iec), 2001.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 9241-11**: Ergonomic requirements for office work with visual display terminals (VDTs). Genebra, 1998.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 14598-1:1999**: Information technology -- Software product evaluation -- Part 1: General overview. Genebra: International Electrotechnical Commission (iec), 1999

KANABAR, Vijay; WARBURTON, Roger J. **Gestão de Projetos**. Traduzido por: Cecília Balbinotti. São Paulo: Saraiva, 2012.

KARDEC, Marcela Silva. **Estudo de compatibilidade entre PMBOK® e SCRUM**. Revista Tecnologias em Projeção, Brasília, v. 3, n. 1, p. 1-7, jul. 2012.

KERZNER, Harold. **Gestão de projetos: as melhores práticas**. 2. ed. Porto Alegre: Bookman, 2007.

KUMAR, Janaki; HERGER, Mario. **Gamification at Work: Designing engaging business software**. Aarhus: Interaction Design Foundation, 2013.

LARSON, Erik W. **Gerenciamento de projetos: o processo gerencial**. 6. Ed. Porto Alegre: AMGH, 2016.

LAZZARO, Nicole. **Why We Play Games: Four Keys to More Emotion Without Story**. Player Experience Research and Design for Mass Market Interactive Entertainment. Oakland: XEODesign Inc, 2004.

LORIGGIO, Alexandre Fekettia; FARIAS, Victor Mitsunaga; MUSTARO, Pollyana Notargiacomo. Aplicações de gamificação e técnicas de motivação à aprendizagem

da metodologia ágil scrum. In: VIII INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 8., 2013, Luanda, Angola. **VIII International Conference on Engineering and Computer Education**. Luanda, Angola: Viii International Conference On Engineering And Computer Education, 2013. p. 328 - 332. Disponível em: <<http://copec.eu/congresses/icece2013/proc/works/73.pdf>>. Acesso em: 29 mar. 2017.

MALAYSIAN CONFERENCE IN SOFTWARE ENGINEERING, 5., 2011, Kuala Lumpur. **Using JIRA and Redmine in Requirement Development for Agile Methodology**. Kuala Lumpur: Mimos Berhad, 2011. 6 p. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/6140707/?section=abstract>>. Acesso em: 09 out. 2017.

MARINS, Luiz; MUSSAK, Eugenio. **Motivação: Do querer ao fazer**. Campinas: Papirus 7 Mares, 2 013.

MARTINS, José Carlos Cordeiro. **Técnicas para gerenciamento de projetos de software**. Rio de Janeiro: Brasport, 2007.

MCCLEAN, Alan. **An Exploration of the Use of Gamification in Agile Software Development**. 2015. 180 f. Dissertação (Mestrado) - Curso de Computação, Gerenciamento do Conhecimento, Dublin Institute Of Technology, Dublin, Irlanda, 2015. Disponível em: <<http://arrow.dit.ie/cgi/viewcontent.cgi?article=1077&context=scschcomdis>>. Acesso em: 15 jun. 2017.

NASCIMENTO, André Luan Chiquetto. **MINERAÇÃO DE DADOS APLICADA AO CONTROLE DE PRAZOS E PRIORIDADES EM PROJETOS DE SOFTWARE**. 2017. 66 f. TCC (Graduação) - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/7423/1/PG_COADS_2017_1_02.pdf>. Acesso em: 09 out. 2017.

POPPENDIECK, Mary; POPPENDIECK, Tom. **Implementando o desenvolvimento Lean de software: do conceito ao dinheiro**. Traduzido por: Luiz Cláudio Parzianello, Jean Patikowski Cheiran. Porto Alegre: Bookman, 2011.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. Traduzido por: Ariovaldo Griesi, Mario Moro Fecchio. 7. ed. Porto Alegre: AMGH, 2011.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.

PROJECT MANAGEMENT INSTITUTE. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)**. Pennsylvania-USA: PMI Publications, 2013.

RIBEIRO, André Luiz Dias; ARAKAKI, Reginaldo. Gerenciamento de Projetos Tradicional x Gerenciamento de Projetos Ágil: Uma análise comparativa. In: 3º Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação, 3., 2006, São Paulo. **3º Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação**. São Paulo: Tecsi, 2006. p. 1594 - 1604. Disponível em: <<http://www.contecsi.fea.usp.br/envio/index.php/contecsi/3contecsi/paper/viewFile/2014/1128>>. Acesso em: 27 jun. 2017.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. **Metodologias ágeis: engenharia de software sob medida**. 1. ed. São Paulo: Érica, 2012.

SILVA, Walmir Rufino da; RODRIGUES, Cláudia Medianeira Cruz. **Motivação nas Organizações**. São Paulo: Atlas, 2007.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TRENTIM, Mário Henrique. **Gerenciamento de projetos: guia para as certificações CAPM® e PMP®**. São Paulo: Atlas, 2014.

VALERIANO, Dalton. **Moderno Gerenciamento de projetos**. 2. ed. São Paulo: Pearson Education do Brasil, 2015.

VALERIANO, Dalton. **Moderno Gerenciamento de Projetos**. São Paulo: Prentice Hall, 2005.

VARGAS, Letícia Marques. **Gerenciamento ágil de projetos em desenvolvimento de software: um estudo comparativo sobre a aplicabilidade do Scrum em conjunto com PMBOK® e/ou PRINCE2**. Revista de Gestão e Projetos, Pelotas, v. 7, n. 3. p. 1-14, set. 2016.

VARGAS, Ricardo. Gerenciamento de Projetos: **Estabelecendo Diferenciais Competitivos**. Rio de Janeiro: Brasport, 2016.

VIANNA, Ysmar; VIANNA, Maurício; MEDINA, Bruno; TANAKA, Samara. **Gamification, Inc: como reinventar empresas a partir de jogos**. 1. ed. Rio de Janeiro: MJV Press, 2013.

ZICHERMANN, Gabe; CUNNINGHAM, Christopher. **Gamification by Design**. Sebastopol: O'Reilly Media Inc, 2011.

APÊNDICE A – TERMO DE ABERTURA DE PROJETO

Sumário

1	Objetivos deste documento	1
2	Situação atual	1
3	Critérios de sucesso do projeto	1
4	Estrutura Analítica do Projeto – Fases e principais entregas	1
5	Marcos	1
6	Partes interessadas do projeto	2
7	Restrições	2
8	Premissas	2

1 Objetivos deste documento

2 Situação atual

3 Critérios de sucesso do projeto

4 Estrutura Analítica do Projeto – Fases e principais entregas

5 Marcos

Fase ou Grupo de Processos	Marcos	Previsão

6 Registro de partes interessadas

Empresa	Nome	Função

7 Restrições

(Limitação que afeta o desempenho do projeto).

8 Premissas

(Requisitos necessários para fins de planejamento).

APÊNDICE B – PLANO DE GERENCIAMENTO DE PROJETO

Sumário

1	Objetivos deste documento	1
2	Registro de requisitos/histórias	1
3	EAP	1
4	Registro de partes interessadas	1
5	Registro de membros da equipe e funções.....	2
6	Registro de riscos	2
7	Registro de formas de comunicação	2
8	Registro de requisitos de qualidade	2
9	Registro de aquisições.....	2
10	Arquitetura e diagramas UML.....	2

1 Objetivos deste documento

2 Registro de requisitos/histórias

3 EAP

4 Registro de partes interessadas

Empresa	Nome	Função

5 Registro de membros da equipe e funções

Nome	Função

6 Registro de riscos**7 Registro de formas de comunicação****8 Registro de requisitos de qualidade****9 Registro de aquisições****10 Arquitetura e diagramas UML**

APÊNDICE C – PLANO DE GERENCIAMENTO DE CUSTOS

Sumário

1	Objetivos deste documento	1
2	Lista de aquisições realizadas	1
3	Estimativa de custos	1

1 Objetivos deste documento

2 Lista de aquisições realizadas

3 Estimativa de custos

(Estimar valor hora do desenvolvimento e levantar histórias da EAP para estipular o tempo de desenvolvimento).

APÊNDICE D – TERMO DE ENCERRAMENTO DO PROJETO

Sumário

1	Objetivos deste documento	1
2	Planejado x Realizado	1
2.1	Os objetivos foram atingidos?	1
2.2	Atendeu o escopo?	1
3	Processos de gerenciamento de projetos	1
4	Recomendações a serem adotadas para os próximos projetos	1

1 Objetivos deste documento

2 Planejado x Realizado

2.1 Os objetivos foram atingidos?

2.2 Atendeu o escopo?

3 Processos de gerenciamento de projetos

(Em projetos futuros, quais são os aspectos que devem ser evoluídos?)

4 Recomendações a serem adotadas para os próximos projetos