



Jaqueline Weschenfelder

**APLICAÇÃO DE METODOLOGIA ÁGIL PARA GESTÃO DE
PROJETO DE PERSONALIZAÇÃO DE *SOFTWARE***

Horizontina - RS

2018

Jaqueline Weschenfelder

**APLICAÇÃO DE METODOLOGIA ÁGIL PARA GESTÃO DE
PROJETO DE PERSONALIZAÇÃO DE SOFTWARE**

Trabalho Final de Curso apresentado como requisito parcial para a obtenção do título de bacharel em engenharia de Produção na Faculdade Horizontina, sob a orientação do Prof. Me. Eloir Fernandes e coorientação do Prof. Esp. Fabrício Desbessel.

Horizontina - RS

2018

**FAHOR - FACULDADE HORIZONTALINA
CURSO DE ENGENHARIA DE PRODUÇÃO**

A Comissão Examinadora, abaixo assinada, aprova o trabalho final de curso

**“Aplicação de metodologia ágil para gestão de projeto de
personalização de *software*”**

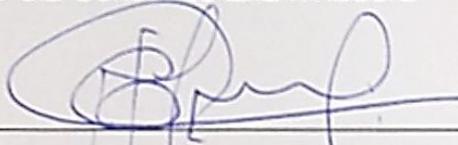
Elaborada por:

Jaqueline Weschenfelder

Como requisito parcial para a obtenção do grau de Bacharel em
Engenharia de Produção

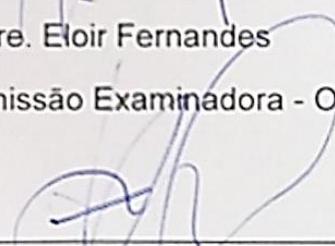
Aprovado em: 04/12/2018

Pela Comissão Examinadora



Mestre. Eloir Fernandes

Presidente da Comissão Examinadora - Orientador



Especialista. Fabrício Desbessel

FAHOR – Faculdade Horizontalina



Mestre. Marcelo Andre Losekann

FAHOR – Faculdade Horizontalina

Horizontalina - RS

2018

Dedicatória

Dedico este trabalho à minha mãe Lúcia e meu pai Ivo e ao meu companheiro Silvano.

AGRADECIMENTO

Agradeço primeiramente à minha família, em especial aos meus pais Lúcia e Ivo, pelo apoio em todas as minhas decisões, por comemorar minhas conquistas, pela força nos momentos difíceis e pela motivação diária. Ao meu companheiro Silvano, que esteve comigo durante todo o período da faculdade, pelo apoio, conselhos e motivação nesta jornada.

Aos meus amigos, que estiveram comigo em diversos momentos.

Aos professores que fizeram parte da minha graduação, em especial, ao professor e orientador Me. Eloir Fernandes, pela orientação, incentivo e ensinamentos. Ao professor e coorientador Esp. Fabrício Desbessel, pela oportunidade de aplicação deste projeto junto ao NTI e que não mediu esforços na orientação deste trabalho. Ao professor e coordenador do curso Me. Sirnei Cesar Kach pela disposição e ensinamentos ao longo da faculdade.

Agradeço também, ao Felipe e ao Luís pelo apoio e troca de experiências durante o projeto.

Muito obrigada a todos.

“Aprender é a única coisa que a mente nunca cansa, nunca tem medo e nunca se arrepende”.

Leonardo Da Vinci

RESUMO

Para que empresas e instituições se mantenham competitivas no mercado atual é necessária eficiência no desenvolvimento de seus produtos, resposta rápida às mudanças e satisfazer às necessidades dos clientes e, nesse sentido, o gerenciamento de projetos torna-se uma ferramenta essencial. As metodologias ágeis surgiram como uma alternativa para as empresas gerenciar seus projetos de forma rápida, eficiente e com qualidade. O objetivo deste trabalho é aplicar uma metodologia ágil de gestão de projetos no departamento responsável pela personalização de *softwares* em uma Instituição de ensino superior. A metodologia deste trabalho classifica-se como pesquisa-ação devido ao envolvimento da autora no desenvolvimento e aplicação do *Scrum* no núcleo de tecnologia da informação. Através desse estudo sugeriu-se uma metodologia para aplicação e controle do projeto, considerando o ambiente atual do núcleo de tecnologia da informação. Com base nos resultados, é possível identificar que a equipe teve maior eficiência, controle de dados e uma visão clara sobre o produto que precisava ser entregue, evidenciando a validade do trabalho.

Palavras-chave: Metodologias ágeis. *Scrum*. Eficiência.

LISTA DE FIGURAS

Figura 1 - Contexto de iniciação do projeto	16
Figura 2 - Ciclo de vida do projeto	19
Figura 3 - Iterações de grupo de processos dentro de um projeto ou fase.....	21
Figura 4 - Fases do modelo de gerenciamento ágil de projetos	26
Figura 5 - Ciclo de vida do gerenciamento ágil de projetos.	27
Figura 6 - Ciclo de vida do <i>Scrum</i>	34
Figura 7 - Modelo de cartas <i>planning poker</i>	35
Figura 8 - Gráfico de evolução regressiva de iteração	37
Figura 9 - Ciclo de vida da XP	39
Figura 10 - Representação visual do Método IVPM2	40
Figura 11 - Ciclo de vida Crystal.....	42
Figura 12 - Exemplo de quadro <i>Kanban</i>	44
Figura 13 - Modelo de avaliação com base na escala Likert	48
Figura 14 - Visão geral do GLPI	50
Figura 15 – Modelo de identificação dos itens de um cartão	55
Figura 16 - Modelo de quadro <i>Kanban</i>	58
Figura 17 - Identificação do tempo de uma estória no Trello	59
Figura 18 - Método de pontuação de estórias	60
Figura 19 - Gráfico <i>burndown</i>	61
Figura 20 – Quadro de acompanhamento <i>sprint 1</i>	63
Figura 21 - Gráfico <i>burndown</i> do <i>sprint 1</i>	65
Figura 22 – Quadro de acompanhamento <i>sprint 2</i>	67
Figura 23 - Pontuação de uma estória do <i>sprint 2</i>	68
Figura 24 - Andamento do <i>Sprint 2</i> no quadro Trello.....	69
Figura 25 - Quadro Trello após a conclusão do <i>Sprint 2</i>	69
Figura 26 - Gráfico <i>burndown</i> do <i>Sprint 2</i>	70
Figura 27 – Quadro de acompanhamento do <i>sprint 3</i>	72
Figura 28 - Gráfico <i>burndown</i> do <i>sprint 3</i>	72
Figura 29 – Quadro de acompanhamento do <i>sprint 4</i>	74
Figura 30 - Gráfico <i>burndown</i> do <i>sprint 4</i>	74
Figura 31 – Quadro de acompanhamento do <i>sprint 5</i>	76
Figura 32 - Gráfico <i>burndown</i> do <i>sprint 5</i>	77

LISTA DE QUADROS

Quadro 1 - Sequência contínua dos ciclos de vida do projeto.....	18
Quadro 2 - Divisão da família Crystal	42
Quadro 3 - Diferenças entre a metodologia ágil e tradicional	45
Quadro 4 - Comparação entre as práticas do <i>Scrum</i> e o PMBOK.....	46
Quadro 5 - Modelo de <i>sprint backlog</i>	54
Quadro 6 - Descrição dos papéis antes e depois da aplicação do <i>Scrum</i>	62

LISTA DE SIGLAS

ACG	Atividades Complementares de Graduação
CRM	Gestão de Relacionamento com o Cliente
DSDM	Método Orientado para Recursos
ERP	Sistema Integrado de Gestão Empresarial
FDD	Desenvolvimento Orientado para Recursos
GAP	Gerenciamento Ágil de Projetos
GLPI	Sistema Gratuito de Gerenciamento de ativos de Tecnologia da Informação
IES	Instituição de Ensino Superior
ID	Identificação
IVPM2	Método de Gerenciamento de Projeto Iterativo e visual
NTI	Núcleo de Tecnologia da Informação
PB	Requisitos do Produto
PMI	Instituto de Gerenciamento de Projetos
PO	Dono do Produto
SQL	Linguagem de Consulta Estruturada
TOTVS	Empresa Brasileira de <i>Software</i>
TI	Tecnologia da Informação
RUP	Processo Racional Unificado
XP	Programação Extrema

SUMÁRIO

1 INTRODUÇÃO	13
1.1 TEMA	13
1.2 DELIMITAÇÃO DO TEMA.....	14
1.3 PROBLEMA DE PESQUISA	14
1.4 HIPÓTESES	14
1.5 JUSTIFICATIVA	14
1.6 OBJETIVOS	15
1.6.1 Objetivo Geral	15
1.6.2 Objetivos Específicos	15
2 REVISÃO DA LITERATURA	16
2.1 PROJETOS	16
2.2 GERENCIAMENTO DE PROJETOS.....	17
2.2.1 Ciclo de vida do projeto	18
2.2.2 Grupo de processos de gerenciamento de projetos	20
2.2.3 Áreas de conhecimento.....	21
2.3 GERENCIAMENTO ÁGIL DE PROJETOS	23
2.3.1 Origem do manifesto ágil.....	23
2.3.2 Motivos e utilização	25
2.4 METODOLOGIAS ÁGEIS	27
2.4.1 <i>Scrum</i>	27
2.4.2 Ciclo de vida do <i>Scrum</i>	34
2.4.3 <i>Planning poker</i> (planejamento pôker).....	35
2.4.4 Gráfico <i>burndown</i> (progresso).....	35
2.5 XP.....	37
2.6 IVPM2.....	39
2.7 DSDM.....	41
2.8 <i>CRYSTAL METHODS</i> (MÉTODO CRYSTAL)	41
2.9 FDD.....	43
2.10 <i>KANBAN</i> (QUADRO DE TAREFAS)	43
2.11 GERENCIAMENTO ÁGIL DE PROJETOS VERSUS TRADICIONAL.....	45
2.12 COMPARAÇÃO ENTRE <i>SCRUM</i> E PMBOK.....	46
2.13 ESCALA LIKERT.....	47
3 METODOLOGIA	49

3.1 MÉTODOS E TÉCNICAS UTILIZADOS.....	49
3.2 MATERIAIS E EQUIPAMENTOS.....	51
4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	52
4.1 DEFINIÇÃO DE APLICAÇÃO DO <i>SCRUM</i>	52
4.1.1 Papéis	52
4.1.2 Artefatos	53
4.1.3 Eventos	54
4.1.4 <i>Kanban</i>	56
4.1.5 <i>Planning poker</i>	59
4.1.6 Gráfico <i>burndown</i>	60
4.2 ANÁLISE DOS RESULTADOS	61
4.2.1 <i>Sprint 1</i>	62
4.2.2 <i>Sprint 2</i>	66
4.2.3 <i>Sprint 3</i>	71
4.2.4 <i>Sprint 4</i>	73
4.2.5 <i>Sprint 5</i>	75
4.2.6 Avaliação de aplicação da metodologia	78
CONCLUSÃO	80
REFERÊNCIAS.....	83
APÊNDICE A	86
APÊNDICE B	87
APÊNDICE C	88
APÊNDICE D	89
APÊNDICE E	90
APÊNDICE F.....	91
APÊNDICE G.....	92

1 INTRODUÇÃO

As metodologias ágeis estão sendo cada vez mais utilizadas na gestão de projetos. Essa prática, vem ao encontro das empresas se manter competitivas, em um ambiente altamente dinâmico e tecnológico.

A primeira menção sobre o método de gerenciamento ágil foi feita em 2001, na área de desenvolvimento de *software*, onde alguns profissionais sintetizaram suas ideologias. O grande diferencial observado nesse modelo de gerenciamento de projetos, são os princípios e valores baseados na eficiência e eficácia (CRUZ, 2015).

O ambiente atual de negócios é caracterizado como dinâmico, e com ritmo acelerado de mudança. Neste contexto, “os projetos são uma das principais formas de criar valor e benefícios nas organizações, devido a necessidade de os gestores gerenciar orçamentos enxutos, prazos menores e tecnologia e constante atualização” (PMBOK, 2017, p.10).

O Núcleo de Tecnologia da Informação (NTI) da Faculdade Horizontina é o departamento de apoio que tem como principais atribuições propor políticas e diretrizes em assuntos relativos à área de tecnologia da informação. Entre os serviços do núcleo pode-se citar suporte a comunidade acadêmica, suporte e manutenção na infraestrutura de comunicação, suporte, manutenção e implementação de processos em *software* ERP (*Enterprise Resource Planning* / sistema integrado de gestão empresarial) educacional, com personalizações para adaptar-se às características da instituição.

Dessa forma, este trabalho visa analisar os métodos ágeis de gestão de projetos, aplicar o método *Scrum* no núcleo de tecnologia da informação, na área de personalização de *softwares*, e identificar quais as melhorias que este método irá proporcionar aos projetos do NTI.

1.1 TEMA

O tema do presente trabalho constitui-se na aplicação de gerenciamento ágil de projetos, com base nas práticas do *Scrum*.

1.2 DELIMITAÇÃO DO TEMA

Este trabalho delimita-se na utilização de uma metodologia ágil de gerenciamento de projetos aplicada ao projeto Gestão de Relacionamento com o Cliente (*customer relationship management* / CRM) desenvolvido pela área de tecnologia da informação, em uma Instituição de Ensino Superior.

1.3 PROBLEMA DE PESQUISA

O problema da pesquisa é a falta de eficiência e controle dos projetos de *softwares* desenvolvidos no NTI.

Com a utilização de ferramentas e de métodos ágeis de gerenciamento de projetos, a equipe poderá desenvolver seus projetos de forma sequenciada e organizada, atingindo os objetivos do projeto e realizando a entrega de produtos no prazo.

Com base no exposto, pretende-se responder a seguinte pergunta: a utilização de uma metodologia ágil na personalização de *softwares* auxilia a equipe a ter maior eficiência na gestão dos projetos?

1.4 HIPÓTESES

O *Scrum* auxiliará e organizará os projetos de personalização de *softwares*, para que no final de cada fase os objetivos sejam atingidos, permitindo um melhor controle sobre as atividades desenvolvidas pela equipe e, com isso ter uma melhoria significativa na entrega dos projetos, utilizando as documentações de lições aprendidas, para melhorar os projetos futuros.

1.5 JUSTIFICATIVA

Segundo o PMI *Survey* (2017), as metodologias ágeis estão sendo utilizadas em 71% dos projetos, além disso nos últimos 12 meses, 1 em cada 5 projetos, utilizou ferramenta ágil e, 1 entre 5, utilizou ferramentas híbridas. O mesmo instituto ainda relata que os projetos que utilizam metodologia ágil possuem performance de 65% de sucesso nos projetos *versus* 24% sem sucesso.

Antes da aplicação do *Scrum* não havia a utilização de uma metodologia específica para a execução dos projetos. Os projetos eram iniciados por meio de

chamados pela interface *web* do GLPI (sistema gratuito de gerenciamento de ativos de TI). A divisão das tarefas era realizada de acordo com o conhecimento de cada integrante, a definição de data de entrega não era regra, além disso, há situações que por motivo de prioridade, as mesmas não eram realizadas.

O *Scrum* é uma metodologia ágil de gerenciamento de projetos que propõe baixo nível de formalidade, maior qualidade no produto e entregas de curto prazo, gerando vantagem competitiva, redução de desperdício, equipes de projeto auto organizadas e respostas rápidas às mudanças dos clientes.

1.6 OBJETIVOS

1.6.1 Objetivo Geral

Aplicar uma metodologia ágil de gestão de projetos na personalização de *softwares* em uma Instituição de Ensino Superior.

1.6.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- a) Analisar a necessidade de adaptar metodologias ágeis;
- b) Aplicar o *Scrum* no projeto de atendimento ao cliente;
- c) Analisar os resultados obtidos com a utilização do *Scrum*;
- d) Avaliar a satisfação da equipe com a aplicação da metodologia.

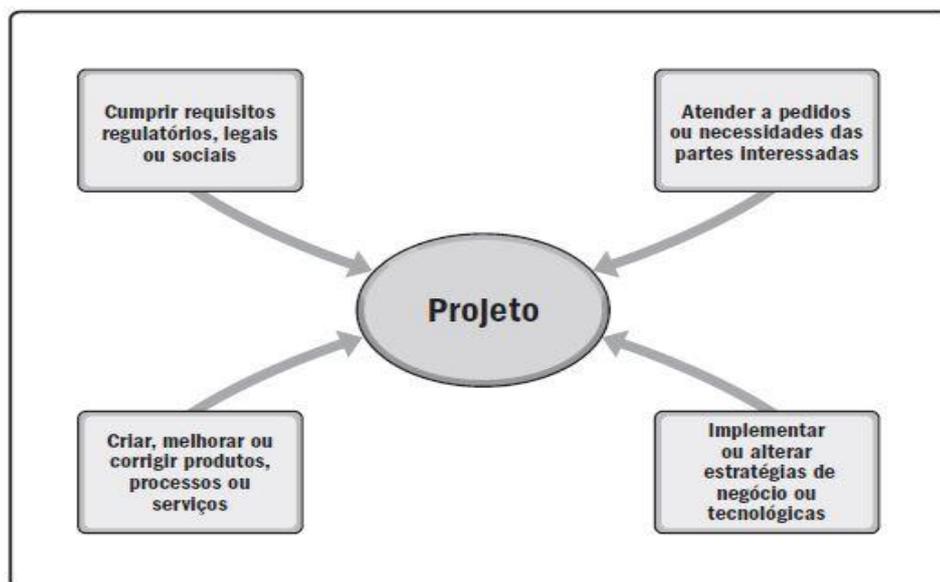
2 REVISÃO DA LITERATURA

2.1 PROJETOS

Projeto é definido como: “um esforço temporário, com início e término definidos; realizado para criar um produto, serviço ou resultado único e cumprir objetivos através da produção de entregas”. Esses projetos podem ocorrer em todos os níveis organizacionais, envolvendo um indivíduo ou um grupo (PMBOK, 2017, p.4).

Projetos são iniciados pela necessidade de resolução de fatores que afetam as organizações, por exemplo: “cumprir requisitos regulatórios, legais ou sociais; atender às necessidades das partes interessadas; implementar ou alterar estratégias de negócio ou tecnologias; e criar, melhorar ou corrigir produtos, processos ou serviços” (PMBOK, 2017, p.7). A figura 1 ilustra o contexto de iniciação do projeto.

Figura 1 - Contexto de iniciação do projeto



Fonte: PMBOK, 2017, p.8.

Cavalcanti (2016), conceitua projeto como um esforço temporário, que possui objetivo de criar um resultado, produto ou serviço único, mesmo que este tenha semelhança em sua estrutura com outros anteriores ou algum elemento próprio

como, por exemplo, as pessoas que vão executá-lo, um cliente diferente, as circunstâncias econômicas, organizacionais.

Para o autor, os projetos são: “criados para gerar valor para a organização executora, clientes e usuários dos produtos; consomem recursos; realizam uma ou mais entregas; apresentam riscos; podem ter um contrato associado” (CAVALCANTI, 2016, p.3).

Para Sabbag (2014), projetos são realizados por organizações modernas em diversas áreas como, por exemplo, engenharia, desenvolvimento e lançamento de produtos, lançamento de serviços, para a implantação de novos sistemas de trabalho, novos processos programas da qualidade, entre outros.

Sotille et al. (2014), listam os principais problemas enfrentados em projetos no Brasil e alguns outros países, que são: o não cumprimento do prazo estabelecido e escopo não definido adequadamente. Ainda, de acordo com o autor, estudos revelam que apenas 37% dos projetos realizados no mundo são bem-sucedidos, no cumprimento do cronograma, qualidade planejada e orçamento.

2.2 GERENCIAMENTO DE PROJETOS

Gerenciamento de projetos “é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto para atender aos seus objetivos” e possui grande importância nos desafios das organizações (CAVALCANTI, 2016, p. 4).

Para o autor, faz parte do gerenciamento de projetos: “identificar as necessidades; estabelecer objetivos claros e alcançáveis; balancear as demandas conflitantes de escopo, tempo, custo e qualidade; balancear os objetivos do projeto em face de diferentes prioridades e expectativas das partes interessadas” (CAVALCANTI, 2016, p.4)

De acordo com PMBOK (2017, p. 10),

O gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de cumprir seus requisitos. O gerenciamento de projetos é realizado através da aplicação e integração apropriadas dos processos de gerenciamento de projetos identificados para o projeto. O gerenciamento de projetos permite que as organizações executem projetos de forma eficaz e eficiente.

2.2.1 Ciclo de vida do projeto

Para Menezes (2018, p.58), ciclo de vida do projeto é “o conjunto de ações desenvolvidas desde o início do projeto, passando pelo seu desenvolvimento até a entrega final dos resultados”.

São as fases pelas quais um projeto passa, desde seu início até sua conclusão, fornecendo a estrutura básica para o gerenciamento do projeto. Estas “fases podem ser sequenciais, iterativas ou sobrepostas”. Já os ciclos de vida do projeto podem ser preditivos ou adaptativos, e normalmente há mais de uma fase associada com o desenvolvimento do produto, serviço ou resultado, estas são ciclo de vida de desenvolvimento e podem ser: “preditivos; iterativos, incrementais, adaptativos (ágeis) ou um modelo híbrido” (PMBOK, 2017, p.19). O quadro 1, apresenta a sequência contínua dos ciclos de vida projeto.

Quadro 1 - Sequência contínua dos ciclos de vida do projeto.

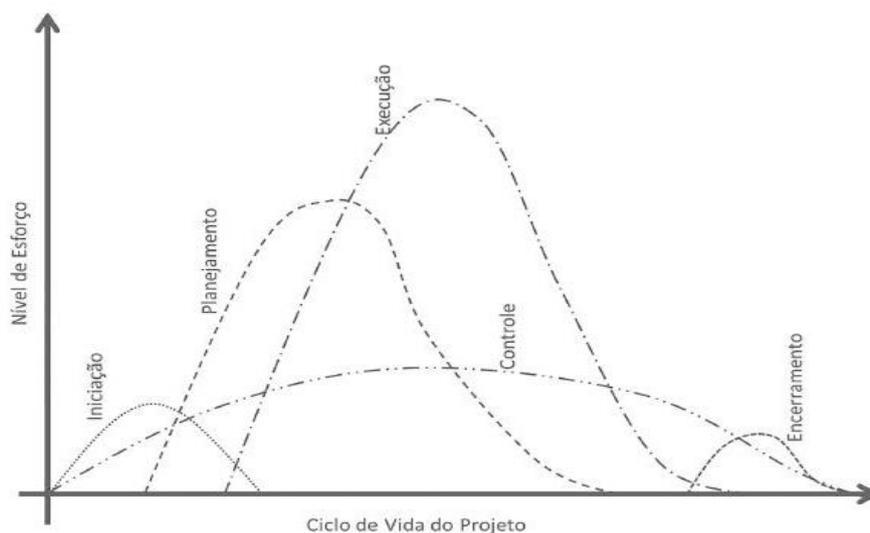
Preditivo	Iterativa	Incremental	Ágil
Requisitos são definidos previamente, antes do início do desenvolvimento	Requisitos podem ser elaborados em intervalos periódicos durante a entrega	Requisitos são elaborados com frequência durante a entrega	
Entrega planos para a entrega final. Em seguida, entregar apenas um único produto final, no fim do projeto	Entregas podem ser divididas em subconjuntos de todo o produto	Entregas acontecem com frequência de acordo com os subconjuntos avaliados pelo cliente de todo o produto	
Mudanças são restritas tanto quanto possível	Mudanças são incorporadas periodicamente	Mudanças são incorporadas em tempo real durante a entrega	
Partes interessadas chave são envolvidas em marcos específicos	Partes interessadas chave são envolvidas regularmente	Partes interessadas chave são envolvidas constantemente	
Riscos e custos são controlados pelo planejamento detalhado dos aspectos mais importantes	Riscos e custos são controlados pela elaboração progressiva dos planos com novas informações	Riscos e custos são controlados na medida em que surgem requisitos e restrições	

Fonte: PMBOK, 2017, p. 667.

Para Cavalcanti (2016), as fases do ciclo de vida da gestão do projeto, são formadas por processos e por sobreposição entre as fases durante o projeto. De

acordo com o autor, as fases são: iniciação, planejamento, execução, controle e encerramento. A figura 2, exemplifica o ciclo de vida do projeto.

Figura 2 - Ciclo de vida do projeto



Fonte: PMBOK, 2013, apud Cavalcanti, 2016, p.40.

Os projetos apresentam um ciclo de vida, com início e fim determinados, sendo que entre o início e o final do projeto ele passa pelas fases de: desenvolvimento, estruturação, implantação e, conclusão (MENEZES, 2018).

Ainda, de acordo com Menezes (2018), as fases do ciclo de vida de um projeto podem ser representadas por suas principais características:

- Fase I – fase inicial, ou fase para clarificar as ideias do projeto a ser executado. Fase de identificação das necessidades, objetivos/metapas, recursos necessários e viabilidade quanto a execução;
- Fase II – planejamento – fase de definição e detalhamento das tarefas a serem realizadas. Entre as atividades estão a definição dos papéis da equipe do projeto, estruturação e detalhamento das tarefas, definição dos tempos de cada atividade;
- Fase III – execução – nesta fase o trabalho é executado, sendo realizadas as atividades de: “comunicação entre os membros da equipe do projeto,

execução das etapas previstas e programadas, utilização dos recursos humanos e materiais” (MENEZES, 2018, p. 62);

- Fase IV – conclusão – é a fase de finalização do projeto. Nessa fase é realizada a verificação de possíveis pendências, coleta de lições aprendidas avaliação dos resultados e desempenho alcançados.

2.2.2 Grupo de processos de gerenciamento de projetos

É um conjunto de processos de gerenciamento de projetos para alcançar os objetivos específicos do projeto, sendo independentes das fases do projeto. Os cinco grupos de processos de gerenciamento de projetos, de acordo com o PMBOK (2017, p. 554), são:

a) Grupo de processos de iniciação

“São os processos realizados para definir um novo projeto ou uma nova fase de um projeto que já existe através da obtenção de autorização para iniciar o projeto ou fase”.

b) Grupo de processos de planejamento

“São aqueles realizados para definir um novo projeto ou uma nova fase de um projeto existente, através da obtenção de autorização para iniciar o projeto ou fase”.

c) Grupo de processos de execução

“Processos realizados para concluir o trabalho definido no plano de gerenciamento do projeto para satisfazer os requisitos do projeto”.

d) Grupo de processos monitoramento e controle

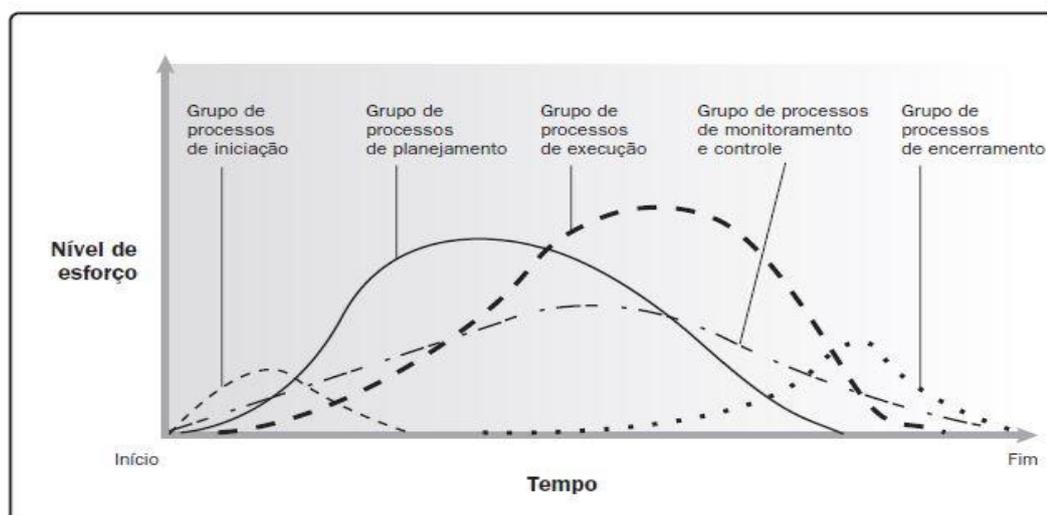
“São os processos realizados para concluir o trabalho definido no plano de gerenciamento do projeto para satisfazer os requisitos do projeto”.

e) Grupo de processos de encerramento

“Os processos realizados para concluir ou fechar formalmente um projeto, fase ou contrato”.

Os grupos de processos não são fases do projeto, os processos nos grupos de processos interagem dentro de cada fase. Quando os projetos são separados em diferentes fases, “os processos em cada grupo de processos são repetidos conforme necessário em cada fase, até que os critérios de conclusão para essa fase tenham sido cumpridos” (PMBOK, 2017, p. 555). A figura 3, representa as iterações de grupo de processos dentro de um projeto ou fase.

Figura 3 - Iterações de grupo de processos dentro de um projeto ou fase.



Fonte: PMBOK, 2017, p. 555.

2.2.3 Áreas de conhecimento

A metodologia do PMBOK divide as áreas de conhecimento em 10 áreas associadas com um tema específico em gerenciamento de projetos, que representam um conjunto de processos, conceitos e prazos. As 10 áreas de conhecimento do PMBOK (2017, p. 553) são:

2.2.3.1 Gerenciamento da integração do projeto

Inclui os processos e atividades para identificar, definir, combinar, unificar e coordenar os vários processos e atividades de gerenciamento dentro dos grupos de processos de gerenciamento do projeto.

2.2.3.2 Gerenciamento do escopo do projeto

“Inclui os processos necessários para assegurar que o projeto inclui todo o trabalho necessário, e apenas o necessário para terminar o projeto com sucesso”.

2.2.3.3 Gerenciamento do cronograma do projeto

“Inclui os processos necessários para gerenciar o término dentro do prazo do projeto”.

2.2.3.4 Gerenciamento dos custos do projeto:

Inclui os processos envolvidos em planejamento, estimativas, orçamentos, financiamentos, gerenciamento e controle dos custos, de modo que o projeto possa ser terminado dentro do orçamento aprovado.

2.2.3.5 Gerenciamento da qualidade do projeto

Inclui os processos envolvidos para incorporação da política de qualidade da organização em relação ao planejamento, gerenciamento e controle dos requisitos de qualidade do projeto e do produto para atender as expectativas das partes interessadas.

2.2.3.6 Gerenciamento dos recursos do projeto

“Inclui os processos para identificar, adquirir e gerenciar os recursos necessários para a conclusão bem-sucedida do projeto”.

2.2.3.7 Gerenciamento das comunicações do projeto:

Inclui os processos necessários para assegurar que as informações do projeto sejam planejadas, coletadas, criadas, distribuídas, armazenadas, recuperadas, gerenciadas, controladas, monitoradas e dispostas de maneira oportuna e apropriada.

2.2.3.8 Gerenciamento dos riscos do projeto:

Inclui processos de condução de planejamento, identificação e análise de gerenciamento de risco, planejamento e implementação de resposta e monitoramento de risco em um projeto.

2.2.3.9 Gerenciamento das aquisições do projeto

“Inclui os processos necessários para comprar ou adquirir produtos, serviços ou resultados externos à equipe do projeto”.

2.2.3.10 Gerenciamento das partes interessadas do projeto

Inclui os processos necessários para identificar todas as pessoas ou organizações impactadas pelo projeto, analisando as suas expectativas e o impacto das partes interessadas no projeto, e desenvolvendo estratégias de gerenciamento apropriadas para o engajamento eficaz das partes interessadas nas decisões e execução do projeto.

2.3 GERENCIAMENTO ÁGIL DE PROJETOS

2.3.1 Origem do manifesto ágil

Sbrocco e Macedo (2012, p.90), consideram que o surgimento das metodologias ágeis foi desencadeado devido às metodologias tradicionais ser “engessadas”, isso proporcionou “mudanças de ênfase significativas e estimulando pequenos grupos de desenvolvedores a adotar um processo disciplinado para projetos de qualquer natureza”.

De acordo com Tudesco (2014), os pioneiros no estudo do método ágil foram Nonaka e Takeuchi (1986), que propuseram um conjunto de características abrangentes, que precisavam estar sincronizadas, pois individualmente não traziam agilidade ao projeto. As características citadas pelos autores são: construir na instabilidade; auto-organização da equipe de projeto; sobreposição das fases; amplo aprendizado; controle e a transferência de aprendizado. Estas características, quando associadas, são responsáveis por gerar grandes mudanças, mesmo nas organizações e trazem um novo conceito para a organização, baseado nas necessidades dos clientes e inovação nos processos.

Na década de 1990, especialistas na área de Tecnologia da Informação, publicaram coletivamente “O Manifesto Ágil” (MENEZES, 2018, p. 287). O autor cita exemplos destes métodos: “RUP (*Rational Unified Process* / processo racional unificado) em 1994, *Scrum* em 1995, *Crystal Clear* (Cristal) e *Extreme Programming* (Programação extrema / XP) em 1996, *Adaptive Software Development* (desenvolvimento de *software* adaptativo) e FDD (*Feature Driven Development* / Desenvolvimento orientado para recursos) em 1997 e DSDM (*Dynamic Systems Development Method* / Método de desenvolvimento de sistemas dinâmicos) em 1995”.

O movimento de gestão ágil de projetos (GAP) surgiu a partir do questionamento de alguns autores sobre o futuro da manufatura enxuta. A resposta surgiu da necessidade das organizações se adaptarem o mais rápido possível a constantes mudanças e incertezas que ocorrem com a evolução da tecnologia (JUNGEND, 2014).

O autor ainda destaca que, foi na área de tecnologia de informação que o modelo de gestão de projetos foi aplicado inicialmente, pois o desafio era a compreensão dos requisitos, naquilo que realmente era necessário para os clientes e na distância entre a equipe do projeto e o cliente ao longo do projeto. (JUNGEND, 2014).

O modelo ágil requer constantes adaptações no desenvolvimento de novos produtos, sendo fundamental a percepção e a adequação das pessoas. Proporciona o compartilhamento de responsabilidades e tarefas do projeto, equipes engajadas focadas na solução de problemas e proativas (TUDESCO, 2014).

A agilidade, segundo Eder (2010), está relacionada de forma direta com a habilidade e não há um método. O autor considera que, mesmo que elementos como clientes, mudanças, negócio e ambiente estejam presentes, as características da GAP são a velocidade e a flexibilidade.

O autor ainda destaca que para um método ser considerado ágil, precisa ter características como, por exemplo, capacidade de absorver mudanças quaisquer, sem que estas causem danos ao longo do projeto (EDER, 2010).

Para Menezes (2018), o conceito de agilidade empregado é o da “agilidade da gestão” e está associado à capacidade de responder às mudanças que ocorrem durante o desenvolvimento do projeto, sejam elas técnicas, nos recursos humanos ou mesmo nos requisitos, estabelecidos ou não.

A gestão ágil facilita o caminho dos trabalhos para a produção de produtos e serviços com qualidade, capacidade de respostas rápidas a mudanças, além disso, o autor cita que também proporciona melhora no relacionamento entre os envolvidos no projeto e a colaboração com o cliente (CRUZ, 2016).

2.3.2 Motivos e utilização

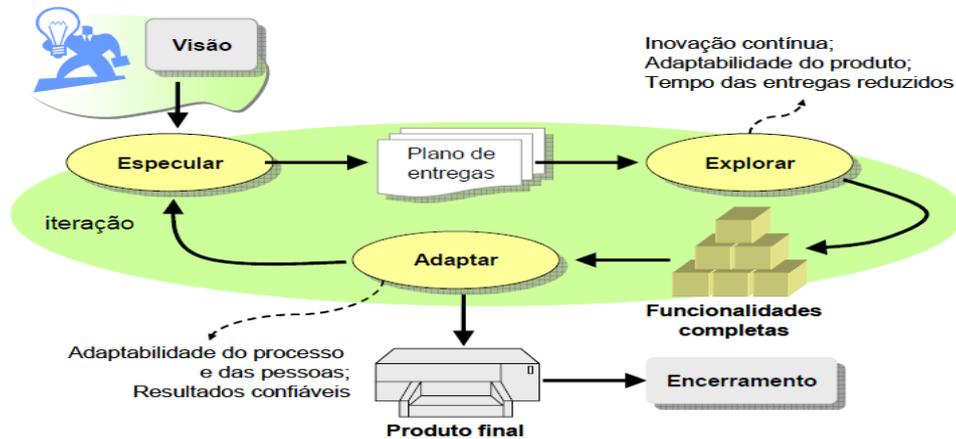
O ambiente no qual se desenvolvem projetos é considerado complexo pelas diversas disciplinas que devem somar seus conhecimentos, instável pelas prioridades e distribuição da atenção entre os projetos e as outras atividades da organização e, portanto, é considerado desafiador. Assim, os métodos ágeis possuem como característica serem não prescritivos ou menos prescritivos que aqueles presentes nos métodos tradicionais (MENEZES, 2018).

Para Conforto (2009), a GAP trouxe grandes inovações no que se refere aos princípios de gerenciamento de projetos. Highsmith (2004 apud Conforto, 2009), sugere um modelo composto por cinco fases, para o desenvolvimento de novos produtos. Conforme descrito a seguir:

- Visão: nessa fase são determinadas a visão e a descrição detalhada do produto do projeto, e as interações da equipe. Possui objetivo de definir o que será entregue ao cliente, quem são os envolvidos e como o time do projeto pretende trabalhar.
- Especulação: é fase de planejamento de maneira mais detalhada, em que serão realizadas a identificação os requisitos do produto, desenvolvimento do plano do projeto, identificação dos riscos, plano de entregas, alocação de recursos, além das estimativas de custos;
- Exploração: é a fase de execução do planejamento. É uma fase essencial em que devem ocorrer reuniões diárias, comprometimento da equipe do projeto com o resultado além da interação com o cliente;
- Adaptação: é uma fase analítica e corretiva da fase anterior e de absorção das possíveis mudanças;
- Encerramento: fase de transferência de lições aprendidas, sendo importantes os chamados “mini fechamentos” ao final de cada iteração.

Na figura 4, estão representadas as fases visão, especulação, exploração, adaptação e encerramento, descritas anteriormente.

Figura 4 - Fases do modelo de gerenciamento ágil de projetos



Fonte: Highsmith, 2004, p.81, apud Conforto, 2009.

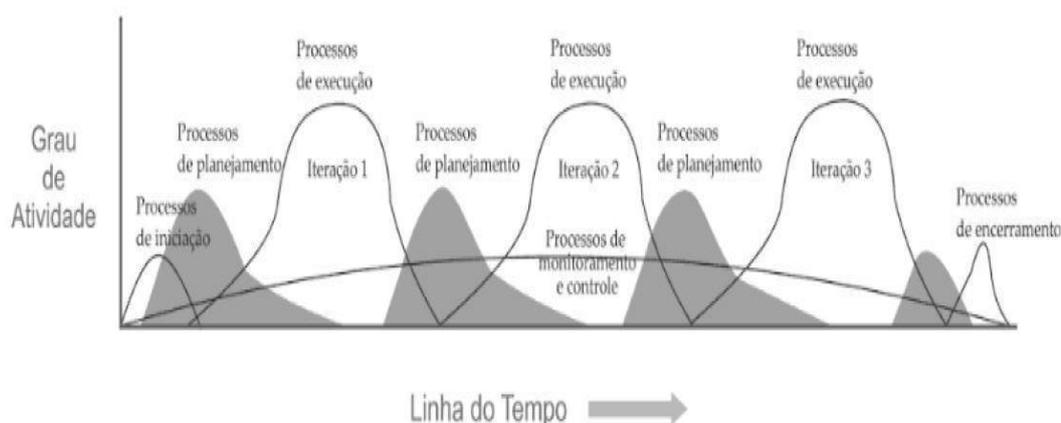
A abordagem ágil de gerenciamento de projetos é mais indicada para ambientes dinâmicos, com constantes transformações, onde as organizações são sistemas complexos adaptáveis e os requisitos são emergentes (Boehm e Turner, 2004 apud Conforto, 2009).

Para Eder (2012), metodologia ágil possui algumas características distintas da metodologia tradicional:

- O foco das entregas é no cliente;
- O escopo é definido de forma macro com orientação de resolução de problemas;
- A identificação da lista de atividades flexível e orientada pela necessidade do cliente;
- Atividades com duração semanal ou mensal;
- Mudanças constantes em benefício do projeto;
- Baixo nível de formalidade, reuniões diárias com participação do cliente.

A utilização do GAP ocorre principalmente no desenvolvimento de *software*, um dos exemplos é o *Scrum* indicado para projetos complexos e inovadores (CONFORTO, 2009). Recomenda-se sua utilização em projetos de muita complexidade e quando os requisitos, recursos e tecnologias são desconhecidos (MASSARI, 2014). O ciclo de vida do gerenciamento ágil de projetos está identificado na figura 5.

Figura 5 - Ciclo de vida do gerenciamento ágil de projetos.



Fonte: Massari, 2014.

O mercado atual possui características de competitividade e muitos ambientes possuem características como imprevisibilidade e incertezas, o que torna difícil a definição de escopo de longo prazo. “O planejamento adaptativo define um plano, mas reconhece que, uma vez iniciado o trabalho, as prioridades podem mudar e o plano precisa refletir esse novo conhecimento” (PMBOK, 2017, p.177).

2.4 METODOLOGIAS ÁGEIS

2.4.1 Scrum

O objetivo do *Scrum* é fornecer um processo apropriado para o projeto e desenvolvimento orientado a objeto. A metodologia sugere uma maneira para desempenhar as atividades com flexibilidade, produtividade e adaptabilidade, por meio de uma abordagem empírica (SCHWABER e BEEDLE, 2002, apud CONFORTO, 2009). Os autores ainda destacam que o foco está na forma de trabalho dos membros da equipe para produzir *software* de forma flexível em um ambiente de mudanças.

O *Scrum* é muito utilizado na área de desenvolvimento de *software*, bem como no planejamento, gerenciamento e desenvolvimento de qualquer produto, principalmente por ser iterativo e incremental (CRUZ, 2016).

O *Scrum* pode ser utilizado em empresas de vários tamanhos e tipo, pois sua aplicação não é limitada a projetos de *softwares*. O autor ainda considera que o *Scrum* é responsável por diversos benefícios, quando comparado a outras formas de se conduzir projetos. Ainda para Sabbagh (2014), devido a metodologia ser apoiada no empirismo, não há uma definição daquilo que realmente é importante para o negócio no início do projeto, pois o ambiente é caracterizado com grau de incerteza e complexidade. Além disso, a metodologia utiliza uma abordagem caracterizada por entregas iterativas e incrementais, fazendo com que ocorra a redução dos riscos do projeto já na sua identificação, tendo em vista que ocorrem ciclos sucessivos de *feedbacks*.

Com a aplicação do *Scrum* os problemas complexos e adaptativos, são resolvidos de maneira produtiva e criativa, entregando produtos com o mais alto valor para o cliente. (SCHWABER e SUTHERLAND, 2017).

Para os autores, para a implementação de controle de processo empírico são necessários três pilares:

- **Transparência:** assegura que os aspectos do processo que afetam o resultado sejam conhecidos e visíveis aos que controlam o resultado;
- **Inspeção:** os processos devem ser totalmente revisados com uma frequência suficiente a fim de detectar as variações, sendo que o processo pode ser modificado pelo próprio ato de inspecionar;
- **Adaptação:** é o ajuste que deve ser realizado o mais rápido possível se durante a inspeção for determinada uma variação fora dos limites aceitáveis em um ou mais aspectos do processo ou material produzido, para minimizar os desvios futuros. A inspeção e adaptação podem ser conduzidas de quatro formas: reunião de planejamento do *sprint* (ciclo/iteração); reunião diária; reunião de revisão do *sprint*; retrospectiva do *sprint*.

Os valores do *Scrum* são comprometimento, coragem, foco, transparência e respeito. As incorporações desses valores, juntamente com os pilares, constroem confiança entre todos do time. O sucesso depende do comprometimento de cada indivíduo em alcançar os objetivos do time (SCHWABER e SUTHERLAND, 2017).

O autor destaca que a ideia principal do *Scrum* é controlar processos que se dão a partir da experiência e aplicação. Esses fatores geram conhecimento adquirido por percepções e pela relação de causa e efeito, além de ser orientado a manter foco na entrega de valor de um negócio no menor tempo possível (CRUZ, 2016).

Para que a metodologia consiga atingir todos os seus objetivos, alguns processos básicos foram definidos:

2.4.1.1 Papéis

Para Schwaber e Sutherland (2017), todos os papéis possuem o mesmo grau de importância para alcançar o resultado do projeto, e identificam a equipe do *Scrum* por:

2.4.1.1.1 *Product Owner* (dono do produto)

O *Product Owner* (PO), é responsável pela maximização do valor do produto, resultado do trabalho do time de desenvolvimento, por gerenciar o *product backlog* (requisitos do produto), e garantir clareza deste, para todos.

2.4.1.1.2 *Scrum Team* (time de desenvolvimento)

O *Scrum Team*, são os profissionais que realizam o trabalho de entregar um incremento potencialmente liberável do produto “ponto” no final de cada *sprint*. Os times de desenvolvimento são estruturados e autorizados pela organização para organizar e gerenciar seu próprio trabalho gerando eficiência e a eficácia do time como um todo. O tamanho de um time de desenvolvimento é pequeno o suficiente para se manter ágil e grande o suficiente para completar o trabalho significativo dentro do *sprint* (SCHWABER e SUTHERLAND, 2017, p.7).

2.4.1.1.3 *Scrum Master* (gerente do projeto)

É o responsável por garantir que as práticas, regras e valores da metodologia sejam seguidas pela equipe. Ele atua como facilitador do *daily meeting* (reunião diária). Também é responsável por resolver os impedimentos encontrados pela equipe do projeto, como também auxiliar o PO e o time de desenvolvimento a desempenhar suas tarefas e atribuições com maior eficiência.

2.4.1.2 Artefatos

Os artefatos são muito importantes para o projeto, pois guiam os integrantes durante todo o projeto. A equipe *Scrum* usa como apoio artefatos específicos e

aplica regras que unem os eventos, os papéis e os artefatos. Além disso, aumentam a transparência das informações (CRUZ, 2015). Os artefatos são:

2.4.1.2.1 *Product backlog* (pacote de entrega do produto)

O *product backlog* (PB) é “uma lista de itens que possui todas as características, funções, atendimento de requisitos, do produto a ser entregue ao cliente. Os itens são em forma de estórias” (CRUZ, 2015).

Esta lista possui tudo o que provavelmente será desenvolvido no projeto. Os itens são geralmente descritos como estórias de usuários (*user stories*), que descrevem os requisitos a partir da visão do usuário final (SABBAGH, 2014).

2.4.1.2.2 *Sprint backlog* (pacote de entrega do ciclo)

Os projetos são divididos em *sprints* de curta duração, que permitem modificar e adaptar os ciclos para corrigir os desvios no produto (incrementos) de forma mais rápida possível, podem ter duração de duas a quatro semanas (CRUZ, 2016).

São itens do PB que serão desenvolvidos no *sprint* para atingir seu objetivo. A partir do *sprint backlog* o time de desenvolvimento identifica a quantidade de trabalho necessário para entregar o incremento “pronto” (SCHWABER e SUTHERLAND, 2017).

Para os autores, no desenvolvimento de um *sprint*, poderá surgir a necessidade de incluir novas estórias, devido ao aprendizado do time e a identificação da necessidade de incluí-las para que o objetivo seja atingido. Quando surge essa necessidade o time de desenvolvimento deve adicioná-lo ao *sprint* e com isso a estimativa do trabalho restante é atualizada. Além disso, o *sprint backlog* é altamente visível em tempo real de trabalho (SCHWABER e SUTHERLAND, 2017).

Para Schwaber e Beedle (2002, apud Conforto 2009), o objetivo do *sprint* é definido nesta reunião, onde ocorre o planejamento do *sprint*. PO apresenta o PB priorizado que serão incluídos no *sprint* e o time de desenvolvimento define quais estórias do PB serão incluídas no *sprint backlog*.

Os elementos da lista do *sprint backlog* são as estórias e serão divididas em tarefas pelo time de desenvolvimento (SCHWABER e BEEDLE, 2002, apud CONFORTO, 2009).

2.4.1.3 Eventos

Os eventos são realizados de forma constante com o propósito de minimizar a necessidade de reuniões não definidas. Iniciado o *sprint*, os eventos restantes podem terminar sempre que o objetivo do evento for alcançado, garantindo dessa forma, que uma quantidade adequada de tempo seja gasta, reduzindo chances de desperdícios no processo. São projetados para permitir transparência e inspeção criteriosa, onde é possível inspecionar e adaptar algum item (SCHWABER e SUTHERLAND, 2017). Os eventos são: *sprint planning* (planejamento do ciclo), *daily meeting* (reunião diária), *sprint review* (revisão do ciclo) e *sprint retrospective* (retrospectiva do ciclo).

2.4.1.3.1 *Sprint planning* (planejamento do ciclo)

As abordagens ágeis caracterizam-se por utilizar ciclos curtos para realizar os trabalhos, analisar os resultados e realizar melhorias. “Esses ciclos fornecem *feedback* rápido sobre as abordagens e adequação das entregas, e geralmente se manifestam como elaboração de cronograma iterativa e sob demanda, baseadas em extração” (PMBOK, 2017, p.178).

O início de um *sprint* ocorre com a reunião de *sprint planning*, momento em que o trabalho que será realizado é planejado. O time de desenvolvimento e o PO negociam o que será desenvolvido, a partir dos itens do *product backlog*. Nessa reunião é estabelecido um objetivo a ser alcançado a partir das estórias (itens), chamada de meta do *sprint* (SABBAGH, 2014).

O *sprint* é caracterizado pelos autores como o coração do *Scrum*. “Sua duração varia entre duas a quatro semanas, ou menos, durante o qual o incremento de produto potencialmente liberável é criado. Inicia-se um novo *sprint* logo após a conclusão do *sprint* anterior” (SCHWABER e SUTHERLAND, 2017, p.9).

Quando uma abordagem ágil é utilizada, é necessário realizar reuniões de planejamento de *sprint* ou iteração para discutir itens da lista de *backlog* de produto priorizados (histórias de usuário) e decidir com quais desses itens a equipe se comprometerá a trabalhar na próxima iteração. A equipe divide

histórias de usuário em tarefas de baixo nível, com estimativas em horas e, em seguida, valida as estimativas que são realizáveis com base na capacidade da equipe ao longo da duração (iteração). Esta reunião é normalmente realizada no primeiro dia da iteração e conta com a participação do proprietário do produto, da equipe de *Scrum* e do gerente do projeto. O resultado da reunião inclui um *backlog* de iteração, bem como premissas, preocupações, riscos, dependências, decisões e ações (PMBOK, 2017, p. 203).

O *sprint planning* é realizado em conjunto com o time *Scrum*. Para um *sprint* de 30 dias, sugere-se uma reunião de 8 horas, para os *sprint* menores esse tempo será menor. Nesta etapa, é papel do *Scrum Master* é garantir que o evento ocorra e que os participantes entendam seu propósito. O PO informa o objetivo a ser atendido pelo *sprint* e os itens do PB que serão necessários para o atendimento dos objetivos. Há colaboração de todo o time *Scrum* (SCHWABER e SUTHERLAND, 2017).

De acordo com Schwaber e Sutherland (2017), o planejamento do *sprint* responde às seguintes questões:

- 1) O que pode ser entregue como resultado ao incremento do próximo *sprint*?
- 2) Como o trabalho necessário para entregar o incremento será realizado?

Eles são formados por: planejamento, reuniões diárias, trabalho de desenvolvimento, revisão e retrospectiva do *sprint*. Outro ponto a destacar, é que durante seu andamento, não serão realizadas mudanças potenciais de perigo ao objetivo do *sprint*, as metas de qualidade não diminuem e poderá haver uma renegociação do escopo entre o PO e o time de desenvolvimento (CRUZ, 2015).

Como dito anteriormente, um *sprint* possui duração máxima de um mês, devido a possibilidade de ocorrer mudanças sobre o que será construído, e conseqüentemente, aumento de complexidade e de risco. Eles permitem a antecipação e garantem a inspeção e adaptação do progresso em busca do objetivo. No final de cada *sprint* deve-se ter uma parte do produto concluída para que possa ser apresentada ao cliente. Essas entregas parciais vão sendo implementadas até a conclusão do produto final (SCHWABER e SUTHERLAND, 2017).

Caso o objetivo do *sprint* não satisfazer as necessidades do projeto, seu cancelamento poderá ocorrer por parte do PO, podendo haver influência do time de desenvolvimento, *Scrum Master* ou das partes interessadas. O produto completo e

“pronto” deverá ser revisado. Os autores enfatizam que o cancelamento de um *sprint* consome recursos, são traumáticos para o time *Scrum* e são muito incomuns (SCHWABER e SUTHERLAND, 2017).

2.4.1.3.2 *Daily meeting* (reunião diária)

É uma reunião diária, realizada em todos os dias do *sprint*, é um evento de 15 minutos em que o time de desenvolvimento e o *Scrum Master* reúnem-se em mesmo local e horário e o time de desenvolvimento analisa o que foi realizado nas últimas 24 horas e faz o planejamento para trabalho para as próximas 24 horas, além de informar os obstáculos que estão atrapalhando o time a alcançar a meta do *sprint* (SCHWABER e SUTHERLAND, 2017).

Abaixo, estão as perguntas, listadas por Schwaber e Sutherland (2017):

- 3) O que eu fiz ontem que ajudou o Time de desenvolvimento a atingir a meta do *sprint*?
- 4) O que eu farei hoje para ajudar o Time de desenvolvimento atingir a meta do *sprint*?
- 5) Eu vejo algum obstáculo que impeça a mim ou o Time de desenvolvimento no atingimento da meta do *sprint*?

O *Scrum Master* é responsável por garantir que as reuniões diárias serão realizadas, sendo que o time de desenvolvimento é responsável por conduzir esta reunião. Após esta reunião, a equipe do *Scrum* ou o time de desenvolvimento poderá realizar reuniões detalhadas, ou adaptar, replanejar o restante do trabalho do *sprint*. Caso outras pessoas estejam presentes no local, estas não poderão interferir. “As reuniões diárias melhoram o nível de conhecimento do time de desenvolvimento e é uma reunião chave para inspeção e adaptação” (SCHWABER e SUTHERLAND, 2017, p.12).

2.4.1.3.3 *Sprint review* (revisão do ciclo)

Ao final de um *sprint*, é realizada a inspeção e ajustes no produto, onde o time de desenvolvimento deverá apresentar e entregar um incremento de produto, que representa valor visível para os clientes do projeto. (CRUZ, 2015).

O resultado da revisão do *sprint* define os possíveis itens para o próximo *sprint*, além disso, também pode ser ajustado completamente para atender novas oportunidades (SCHWABER e SUTHERLAND, 2017).

2.4.1.3.4 *Sprint retrospective* (retrospectiva do ciclo)

É nesse evento que a equipe avalia como foi o *sprint*. É uma oportunidade de a equipe criar melhorias para o próximo *sprint* (SCHWABER e SUTHERLAND, 2017).

Para Cruz (2015), a retrospectiva ocorre ao final de cada *sprint*, momento em que o time do projeto analisa como ocorreu o último *sprint*. Essa reunião é realizada após a apresentação de entrega do produto. O *Scrum* propõe que esta retrospectiva ocorra no final de cada *sprint* para que as lições aprendidas já possam ser colocadas em prática, criando assim, um ambiente de melhoria contínua para os *sprints* seguintes.

2.4.2 Ciclo de vida do *Scrum*

Para os autores Sbrocco e Macedo (2012), a partir da visão inicial do projeto é elaborada uma lista enxuta dos principais itens, contendo funcionalidades e requisitos para desenvolvimento até o final do projeto. Durante o projeto poderão ser acrescentados outros itens a essa lista. A figura 6, representa o ciclo de vida do *Scrum*.

Figura 6 - Ciclo de vida do *Scrum*



Fonte: Cohn, 2008, apud Sbrocco e Macedo, 2012, p. 162.

2.4.3 *Planning poker* (planejamento pôker)

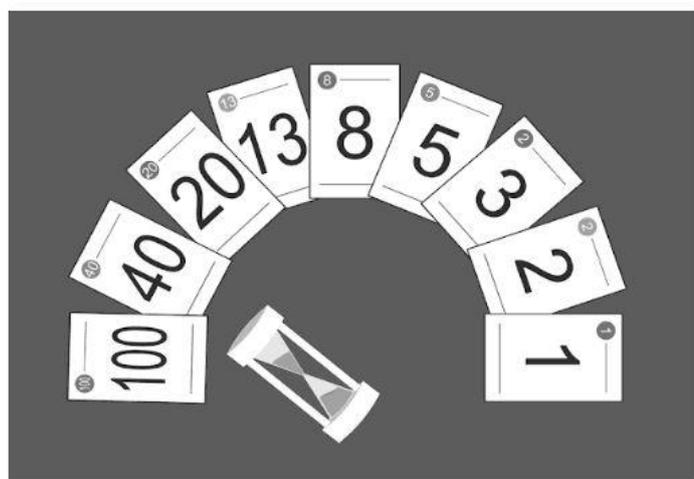
O esforço, tempo, necessário para realizar uma estória é estimado pelo time de desenvolvimento, utilizando um jogo de cartas, o *planning poker* (MASSARI, 2014).

É uma “unidade relativa de tempo criada pelo time de desenvolvimento, sendo a unidade mais utilizada por equipes ágeis”. Representam as estimativas para o planejamento das estórias (SABBAGH, 2014, p. 120).

Para Massari (2014), em cada estória, os integrantes do time de desenvolvimento escolhem uma carta de estimativa, de forma individual, e todos mostram a carta escolhida ao mesmo tempo, caso ocorram valores diferentes é realizada uma nova rodada até que a equipe chegue em um consenso.

As vantagens dessa técnica, citadas pelo autor são: estimativas com base no consenso, evitando superestimativas; discussão entre equipe sobre a estória (MASSARI, 2014). A figura 7 apresenta um modelo de cartas - *planning poker*.

Figura 7 - Modelo de cartas *planning poker*



Fonte: Massari, 2014, p. 69.

2.4.4 Gráfico *burndown* (progresso)

É um gráfico de acompanhamento em que a equipe do projeto identifica o andamento das tarefas, de acordo com o planejado, através da representação da

quantidade restante de trabalho do *sprint backlog* ao longo da sua duração. O produto “pronto” é aquele que atende aos critérios do planejamento do produto, satisfazendo as necessidades do cliente (CRUZ, 2015).

É uma forma simples que a equipe possui para identificar o andamento das tarefas não prontas. O gráfico é criado após a reunião de planejamento do *sprint* e a elaboração do *sprint backlog*, sendo atualizado diariamente com os valores já executados e o que ainda falta executar (ISTOB).

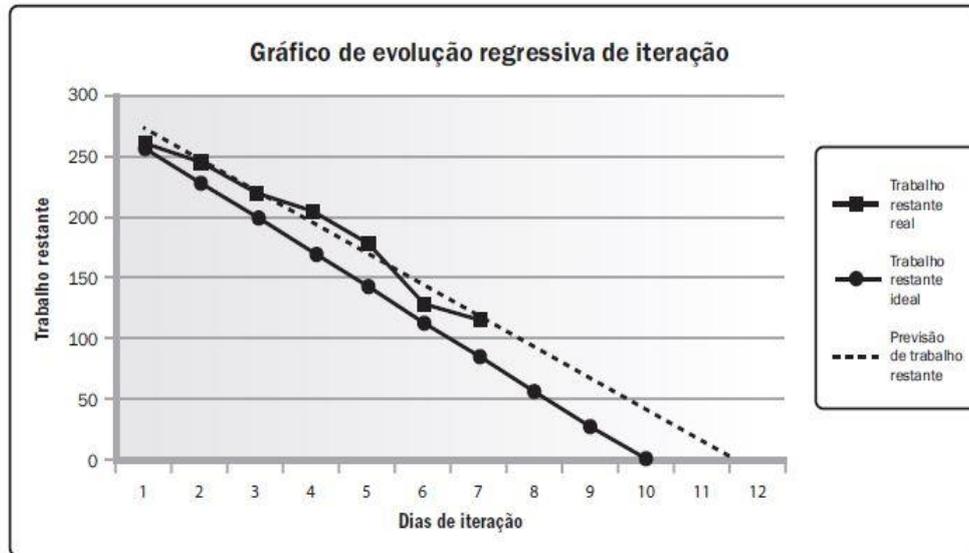
O eixo X representa os dias do *sprint*, e o eixo Y representa o esforço restante para concluí-lo, podendo ser em horas ou pontos de estória. A linha pontilhada representa o fluxo ideal de trabalho e a linha contínua demonstra o andamento real do trabalho. Quando a linha contínua está abaixo da linha pontilhada, significa que o trabalho da equipe está adiantado em relação planejado, a linha contínua acima da linha pontilhada, indica que a equipe está atrasada em relação ao planejado (ISTOB).

Este “gráfico rastreia o trabalho que ainda precisa ser concluído na lista de pendências de iteração”. A partir do gráfico é possível realizar uma análise da variação entre o realizado e o planejado, PMBOK (2017, p. 226):

Uma linha de tendência de previsão pode ser usada para prever a provável variação na conclusão da iteração e adota medidas apropriadas durante o curso da iteração. Uma linha diagonal que representa a evolução regressiva ideal e o trabalho restante real diário e então desenhada. Uma linha de tendência e então calculada para prever a conclusão com base no trabalho restante.

Na figura 8 está representado o gráfico de evolução regressiva de iteração, também chamado de *burndown*.

Figura 8 - Gráfico de evolução regressiva de iteração



Fonte: PMBOK, 2017, p. 226.

2.5 XP

A *Extreme Programming* (Programação Extrema / XP) é uma metodologia ágil indicada para pequenas e médias equipes que desenvolvem *softwares*, em que os requisitos são vagos ou mudam com frequência. São valores da metodologia: comunicação, simplicidade, *feedback* e coragem (BECK, 2004).

Para Sommerville (2011), esta metodologia possui abordagem desenvolvida para impulsionar práticas reconhecidamente boas, como o desenvolvimento iterativo, a “níveis extremos”, permitindo que várias versões de um sistema possam ser desenvolvidas, integradas e testadas em um único dia por programadores diferentes.

A XP é mais utilizada na aplicação de pequenos projetos e normalmente no desenvolvimento de ambientes *web*. Porém, não aconselhasse sua utilização no desenvolvimento de *softwares* complexos, de grande porte ou críticos (HIRAMA, 2012).

Os valores em XP são conceitos não tangíveis, por isso entendeu-se que esta característica representa grande diferença na qualidade final do produto e na motivação de seus times (HIRAMA, 2012). Os valores são:

- Comunicação: direta, eficaz e esclarecedora;

- *Feedback*: testes automatizados de unidade e um servidor de interação contínua para que os testes mais longos sejam realizados com frequência;
- *Coragem*: somente no que efetivamente é necessário no momento, sem previsão de futuro;
- *Simplicidade*: implementar apenas o básico, não antecipar funcionalidades;
- *Respeito*: a preocupação dos membros de uma equipe deverá ser somente com uma comunicação melhor quando se importam uns com os outros;

Para Teles (2004), as práticas desta metodologia podem ser aplicadas a processos que agregam valor à equipe de desenvolvimento. As práticas citadas pelo autor são:

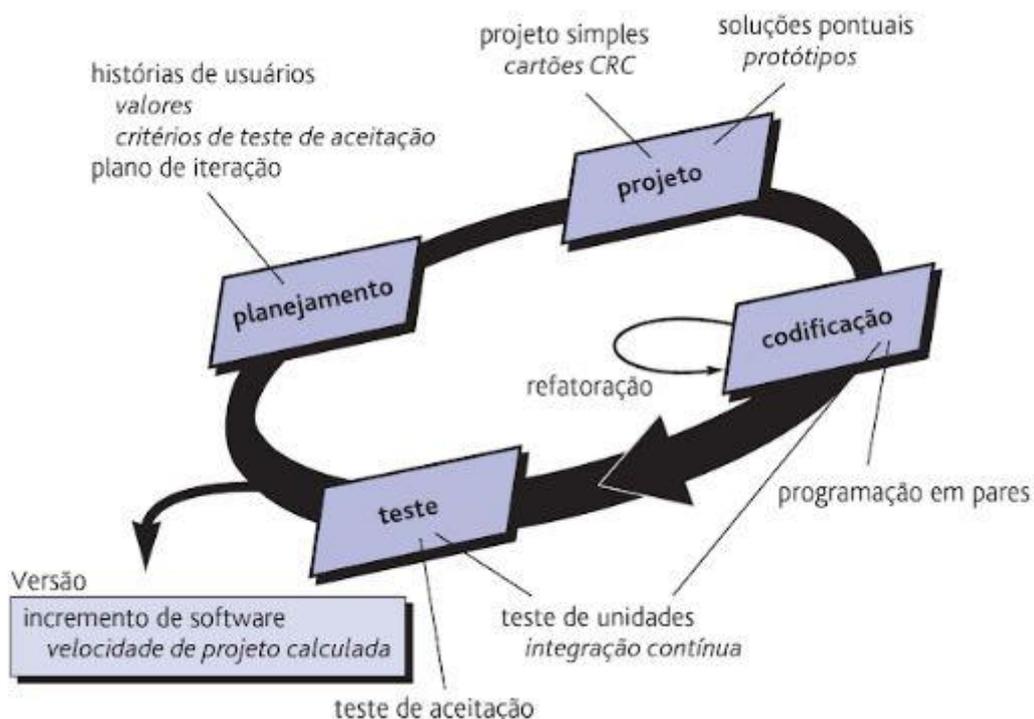
- a) Programação em pares: em dupla, em um único computador;
- b) Cliente presente: o cliente integra a equipe de desenvolvimento;
- c) Reuniões breves: em pé, para não perder o foco, tornando-se reuniões rápidas;
- d) Testes frequentes: construído pelo cliente e conjunto com a equipe;
- e) Refatoração do código;
- f) Integração contínua e semanas de 40 horas.

Os requisitos são definidos como cenários - histórias do usuário - implementadas diretamente como uma série de tarefas. No desenvolvimento de testes para cada tarefa antes de escreverem o código, as histórias são realizadas em pares pelos programadores. Além disso, todos testes devem ser executados com sucesso (SOMMERVILLE, 2011).

Pressman e Maxim (2016), destacam que a XP envolve um conjunto de regras e práticas em um contexto de atividades metodológicas: planejamento; projeto; codificação e testes.

A figura 9 destaca alguns conceitos e tarefas que estão associadas a cada uma das atividades da XP.

Figura 9 - Ciclo de vida da XP



Fonte: Pressman e Maxim, 2016, p.72.

2.6 IVPM2

O *Iterative and visual project management method* (Método de gerenciamento de projeto iterativo e visual / IVPM2) é o método apresentado por Conforto (2009), tem foco na gestão do escopo e tempo no GAP.

O método possui algumas práticas de planejamento e desenvolvimento de produtos e gerenciamento de projetos, destinado ao desenvolvimento de novos produtos. Recomenda-se a sua aplicação para pequenas empresas tecnológicas (CONFORTO, 2009).

As características desta metodologia são: técnicas simplificadas de gerenciamento, planejamento e controle; métodos visuais de informações e documentação simples; processo de desenvolvimento por fases, evolução por meio de iterações ciclos pequenos; participação direta do cliente; integração com sistema de gestão do projeto, poucos gastos com *software*; autodisciplina e auto-gestão da equipe (CONFORTO, 2009).

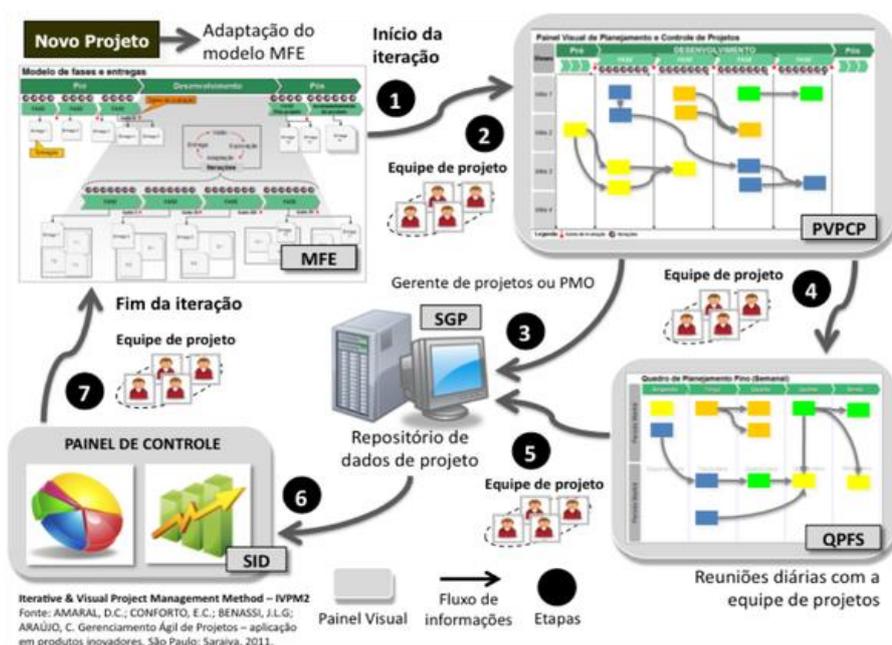
Conforto (2009), explica que o modelo é estruturado com a utilização de cinco métodos práticos. São eles: modelo de fases e entregas, painel visual de planejamento e controle do projeto, quadro de planejamento semanal, sistema de gestão de projetos, sistema de indicadores de desempenho.

Já as etapas descritas por Conforto (2009) são sete, e todas as iterações passam por estas etapas. São elas:

- 1) Definição das entregas e do plano de entregas a partir dos documentos iniciais, partida ao projeto;
- 2) Inserir e distribuir as entregas no painel visual de planejamento e controle, classificando de acordo com o prazo;
- 3) Inserir dados no *software* de gestão indicando datas e responsáveis;
- 4) Definição das tarefas através de reuniões curtas e objetivas;
- 5) Execução das atividades e tarefas do projeto;
- 6) Geração de relatório de desempenho e progresso do projeto;
- 7) Plano de ações e mudanças para próxima iteração.

As etapas do método IVPM2 descritas anteriormente, estão representadas na figura 10.

Figura 10 - Representação visual do Método IVPM2



Fonte: Conforto, 2009.

2.7 DSDM

O *Dinamic Systems development methodology* (Método de desenvolvimento dinâmico de sistema / DSDM), é uma abordagem ágil de desenvolvimento de *software*. Esta fornece uma estrutura para a construção de projetos de curto prazo, e a utilização de processo iterativo. Nessa metodologia utiliza-se a regra 80/20, que sugere que 80% de uma aplicação podem ser entregues em 20% do tempo que levaria para entregar uma aplicação completa (PRESSMAN, 2011).

Para Sbrocco e Macedo (2012, p.112), o objetivo do DSDM “visa estabelecer os recursos e o tempo fixo para o desenvolvimento de um projeto, ajustando suas funcionalidades, para o atendimento de prazos estipulados”. Para os autores, “o ciclo de vida esta metodologia possui 5 fases, sendo: estudo de viabilidade, estudo de negócio, modelo de iteração funcional, projeto e construção de iteração e implementação”.

Nesta metodologia os papéis são 15, entre *stakeholders* (partes interessadas) e desenvolvedores. Os autores consideram que as funções podem ser acumuladas, dependendo da complexidade do projeto. Os papéis são: coordenador técnico/arquiteto; desenvolvedor sênior; analista; programador; designers; testadores; usuário embaixador; usuário visionário; usuário conselheiro; patrocinador executivo; redator técnico; desenvolvedor; DBA (data base administrador); configurador; e, suporte técnico (SBROCCO e MACEDO, 2012).

2.8 CRYSTAL METHODS (MÉTODO CRYSTAL)

A metodologia foi criada em 1998, com o objetivo de suprir as necessidades da época. Para Sbrocco e Macedo (2012, p.134), “trata-se de uma família de metodologias com um código genérico para o atender diferentes tipos de projetos e tamanhos”.

Já para Pressman (2011), o principal objetivo é entregar *softwares* úteis funcionando, e o objetivo secundário de preparar-se para o jogo seguinte.

Para Sbrocco e Macedo (2012, p. 134):

A metodologia foi dividida em cores: quanto mais escuro, mais crítico o sistema seria. Cada cor tem um objetivo diferente e elas são separadas de acordo com a criticidade, ou seja, projetos menores envolvem poucos desenvolvedores e, no caso de problemas, o prejuízo tende a ser menor. Já para projetos maiores, ou de segurança crítica normalmente envolvem mais profissionais, portanto o prejuízo deve ser muito maior, podendo colocar a vida das pessoas em risco.

No quadro 2, estão representadas a divisão do método Crystal as cores, número de desenvolvedores e as consequências em caso de falha.

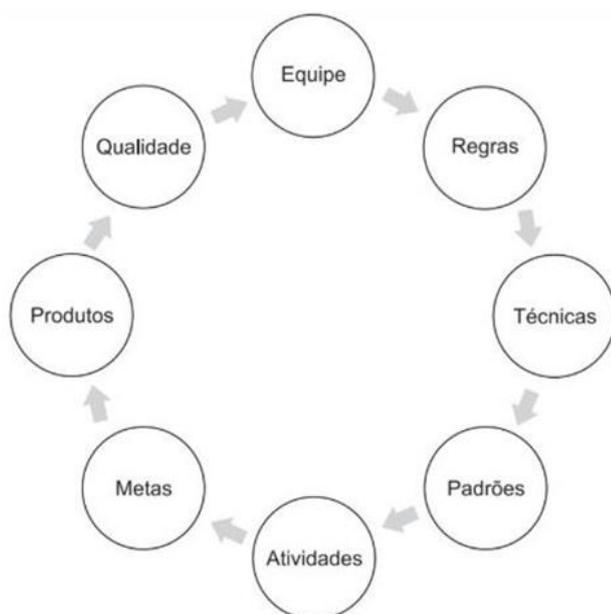
Quadro 2 - Divisão da família Crystal

Cores	Número de desenvolvedores	Em caso de falha...
Clear	1-6	Perdem dinheiro, mas recuperam facilmente.
Yellow	7-20	Perdem dinheiro discretamente.
Orange	21-40	Perdem dinheiro substancialmente.
Red	41-100	Há perda substancial de dinheiro e, possivelmente, vidas humanas.

Fonte: Sbrocco e Macedo, 2012, p.135.

O ciclo de vida de Cristal é baseado em iteração e possui semelhança com um relógio, como pode ser observado na figura 11.

Figura 11 - Ciclo de vida Crystal



Fonte: Sbrocco e Macedo, 2012, p.137.

2.9 FDD

Feature driven development (desenvolvimento orientado a recursos / FDD), é considerada robusta e muito utilizada. Foi criada em Singapura, entre os anos de 1997 e 1998. A primeira utilização da metodologia foi na construção de um sistema bancário internacional, o qual havia sido considerado inviável de ser desenvolvido em um prazo determinado (SBROCCO e MACEDO, 2012).

FDD pode ser traduzido como um desenvolvimento guiado por características no sentido de “uma função valorizada pelo cliente que poderá ser implementada em até duas semanas”, (PRESSMAN, 2011). Para o autor, os benefícios são:

- Facilidade dos clientes em descrever seus requisitos em pequenos blocos de entrega, além de visualizar e revisar a implementação desses requisitos;
- Entregas a cada duas semanas;
- Facilidade na inspeção dos requisitos por serem pequenos;
- Facilidade na criação e monitoramento do cronograma.

Pressman e Maxim (2016), destacam que os projetos que utilizam esta metodologia possuem seis marcos de referência, são eles: travessia do projeto; projeto; inspeção de projeto; código; inspeção de código e promoção para construção. A diferença desta metodologia, se comparada a outros métodos ágeis, é que a FDD concentra esforços nas diretrizes e técnicas de gestão de projeto, pois, entende que à medida que os projetos crescem em tamanho e complexidade, torna-se importante que os desenvolvedores, gerentes e clientes possam visualizar o estado do projeto, quais os avanços realizados e quais os problemas identificados.

2.10 KANBAN (QUADRO DE TAREFAS)

A utilização do sistema *Kanban* permite um controle detalhado de produção com informações sobre quando, quanto e como produzir. Na área de *software*, ferramentas *Kanban* começaram a ser utilizadas para auxiliar a organização visual das tarefas, tais como *XP* e *Scrum*. Os cartões sinalizam os itens de trabalho em cada iteração (ARRUDA, 2012).

De acordo com PMBOK (2017, p. 177) “algumas das práticas emergentes para os métodos de elaboração do cronograma do projeto incluem, mas não estão limitadas a:

Elaboração iterativa de cronograma com lista de pendências (*backlog*). Esta é uma forma de planejamento em ondas sucessivas, baseado em ciclos de vida adaptativos, como a abordagem ágil para desenvolvimento de produtos. Os requisitos são documentados em histórias de usuário, que por sua vez são priorizadas e refinadas pouco antes da construção, e as características do produto são desenvolvidas usando períodos de trabalho definidos. Esta abordagem é muitas vezes usada para fornecer um valor incremental para o cliente ou quando várias equipes podem desenvolver simultaneamente um grande número de características com poucas dependências interligadas. Esse método de cronograma é apropriado para muitos projetos, conforme indicado pelo uso generalizado e crescente de ciclos de vida adaptativos para desenvolvimento de produtos. O benefício desta abordagem é que ela facilita as mudanças durante todo o ciclo de vida de desenvolvimento (PMBOK, 2017, p.177).

Cronograma sob demanda. Essa abordagem, normalmente usada em um sistema *Kanban*, é baseada na teoria das restrições e nos conceitos de sistema puxado de produção enxuta para limitar o trabalho em andamento de uma equipe a fim de equilibrar a demanda com o rendimento de entrega da equipe. A elaboração de cronograma sob demanda não depende de um cronograma desenvolvido previamente para o desenvolvimento do produto nem de incrementos do produto, mas extrai o trabalho de uma lista de pendências (*backlog*) ou fila de trabalhos intermediários que devem ser executados imediatamente à medida que os recursos se tornem disponíveis. A elaboração de cronograma sob demanda é usada frequentemente para projetos que desenvolvam o produto em incrementos em ambientes de operação ou de manutenção, e nos quais as tarefas possam ser realizadas e agrupadas por tamanho e escopo (PMBOK, 2017, p.177).

Para os autores Sbrocco e Macedo (2012), é um dos recursos para a comunicação da equipe, permitindo que esta identifique as estórias de forma clara. A utilização deste quadro ocorre durante todo o *sprint*, principalmente nas reuniões diárias. A figura 12 identifica um quadro em que cartões preenchidos sinalizam os itens de trabalho que a ser concluídos na iteração.

Figura 12 - Exemplo de quadro *Kanban*



Fonte: Adaptado de Kiniberg, 2007.

Sbrocco e Macedo (2012), sugerem que o quadro *Kanban* seja utilizado com *post-it* e com as seguintes informações: *product backlog*; *sprint backlog*; atividades “para fazer”; atividades “fazendo”; atividades “feitas”; atividades “para verificar”.

2.11 GERENCIAMENTO ÁGIL DE PROJETOS VERSUS TRADICIONAL

Eder (2012), apresentou algumas diferenças entre as metodologias ágeis e tradicionais. Enquanto a teoria tradicional é encontrada nos principais guias de práticas, como PMBOK, IPMA (associação internacional de gestão de projetos) e PRINCE2 (*Project In a Controlled Environment / Projetos em Ambiente Controlado*). O autor cita Shenar e Dvir (2007), que realizaram uma comparação entre as teorias. O quadro 3 apresenta as principais diferenças entre o gerenciamento ágil e o gerenciamento tradicional de projeto.

Quadro 3 - Diferenças entre a metodologia ágil e tradicional

	Gerenciamento Ágil	Gerenciamento Tradicional
Objetivo do projeto	Foco em resultados para o negócio. O sucesso é medido em várias dimensões.	Foco em Tempo x Custo x Requisitos de qualidade.
Plano do projeto	Organizado para atingir resultados para o negócio.	As atividades são realizadas para atender a tríplice restrição seguindo um planejamento inicial.
Planejamento	É realizado a cada iteração.	Realizado no início do projeto e seguido até o fim.
Abordagem gerencial	Flexível Variável e Adaptativa.	O gerenciamento é realizado sobre o plano inicial.
Execução	Imprevisível, Não linear e Imensurável, Complexa.	Previsível, Linear, Mensurável e Simples.
Influência organizacional	Afeta o projeto durante a sua execução.	Mínimo e Imparcial.
Controle do projeto	Identificar mudanças e ajustar o plano o quanto antes.	Identificar desvios e retornar ao plano inicial.
Aplicação da metodologia	Processos adaptativos de acordo com o tipo de projeto.	Metodologia igualitária para todos os tipos de projetos.
Gestão	Adaptativa de acordo com o projeto.	Única para todos os tipos de projetos.

Fonte: Shenar e Dvir, 2007, apud EDER, 2012.

2.12 COMPARAÇÃO ENTRE SCRUM E PMBOK

Há possibilidade de compatibilidade entre algumas áreas de conhecimento do PMBOK com a metodologia ágil. Isto é possível em áreas do conhecimento como: gerenciamento de escopo, gerenciamento de cronograma, gerenciamento de integração e gerenciamento de comunicação (LEAL, 2007 apud BRANDÃO, 2010).

Para Vargas (2016), a partir das últimas edições do PMBOK, é possível identificar um movimento lento e de adequação para as metodologias ágeis. No quadro 4, é realizada uma comparação entre a metodologia *Scrum* e o PMBOK.

Quadro 4 - Comparação entre as práticas do *Scrum* e o PMBOK

Scrum	PMBOK
Metodologia ágil.	Padrão de gerenciamento de projetos.
Foco na visão do produto.	Foco no planejamento.
Plano de projeto evolutivo de acordo com a realização dos ciclos, os documentos possuem as seguintes características: simplicidade; linguagem comum para o entendimento de todas as partes interessadas; não são prioridade e possuem informações básicas. O foco do trabalho está na implementação do produto com qualidade, a documentação é secundária.	A documentação é criada pelo gerente de projetos, deve conter o máximo de informações.
Ambiente de incertezas com foco na visão do produto.	O trabalho é realizado a partir do dimensionamento de recursos (certeza).
O escopo é detalhado durante o projeto adaptativo.	Escopo bem definido e detalhado, de onde serão realizadas estimativas de tempo e de custos.
Trata as mudanças como parte do projeto, e devem ser implementadas assim que possível.	Mudanças não são bem-vindas, é preciso realizar uma avaliação dos impactos gerados ao projeto.
Não existe processo específico, os custos são detalhados a cada ciclo, estimados através do <i>product backlog</i> .	Os custos são estimados a partir do escopo do projeto e seus requisitos, utilizando ferramentas específicas.
Orientado a pessoas que são tratadas como indivíduos. As pessoas da equipe definem as entregas a serem executadas até a próxima iteração.	Orientado a processos bem definidos, e devem ser executados obrigatoriamente. O gerente de projetos distribui as atividades e é responsável pelo cumprimento do cronograma.

Scrum	PMBOK
Simplicidade, o que for útil para a geração do resultado.	Burocracia: soluções completas, as equipes têm grande carga de trabalho, compromete a velocidade de desenvolvimento.
Equipes pequenas, multifuncionais, auto organizadas. Ambiente colaborativo a partir da confiança dos membros.	Equipes grandes, liderado pelo gerente de projetos. A alocação da equipe é realizada conforme a necessidade, não necessariamente do mesmo local geograficamente.
Ciclo de vida baseado em iterações (<i>sprint</i>), o detalhamento do escopo acontece a cada ciclo durante o projeto. São realizadas reuniões diárias de planejamento e revisão da iteração. O ciclo tem duração fixa de 2 a 4 semanas.	O ciclo de vida pode ser preditivo, iterativo/incremental e adaptativo. O escopo é definido no início do projeto, o planejamento pode ser detalhado em ondas.
O atendimento da qualidade está no cumprimento dos requisitos estabelecidos pelo cliente e do <i>feedback</i> do mesmo. A revisão da qualidade é realizada em cada ciclo onde é realizada a revisão e as melhorias necessárias.	Padrões de qualidade normatizados e mensurados. O gerenciamento da qualidade é aplicado durante todo o projeto para garantir que os requisitos sejam atendidos.
A comunicação é interpessoal, colaborativa, realizada nas reuniões de planejamento e revisão da <i>sprint</i> , e nas reuniões diárias. São reuniões objetivas, rápidas e com pauta preestabelecida.	A comunicação é planejada e realizada a partir da identificação dos <i>stakeholders</i> e por meios adequados. Também é necessária a formalização e geração da documentação necessária.
Participação do cliente (<i>product owner</i>) durante toda a execução, como parte do projeto.	Pouca participação do cliente no planejamento do escopo ou de gerenciamento de riscos.
O processo de retomada das lições aprendidas é realizado ao final de cada <i>sprint</i> durante a reunião de retrospectiva.	Ao final do projeto é realizada reunião para arquivamento de documentos e retomada das lições aprendidas durante o projeto.
Não há.	Gerenciamento de aquisições.

Fonte: Vargas, 2016.

2.13 ESCALA LIKERT

É um método de avaliação muito utilizado, em que os entrevistados indicam discordância ou concordância sobre determinado item, dessa forma, a escala Likert possui cinco categorias de resposta de “discordo totalmente” até “concordo totalmente” (MALHOTRA, 2012, p. 222).

Ainda, para o autor, “os dados são geralmente tratados como intervalares, dessa forma, a escala Likert possui características de descrição, ordem e distância. Para realizar a análise cada afirmação recebe uma pontuação numérica, variando de -2 a +2 ou de 1 a 5”. (MALHOTRA, 2012, p. 222). O modelo de escala Likert está representado na figura 13.

Figura 13 - Modelo de avaliação com base na escala Likert

O treinamento desenvolvido atingiu seus objetivos ?				
Instrução: marque com um X o que você considera mais próxima da realidade.				
1	2	3	4	5
Discordo totalmente	Discordo	Não concordo nem discordo	Concordo	Concordo plenamente

Fonte: Guimarães e Campos, 2009.

“Com a escala Likert, os entrevistados indicam suas atitudes verificando quão fortemente concordam ou discordam com declarações delas” (ZIKMUND e BABIN, 2012, p. 299).

3 METODOLOGIA

3.1 MÉTODOS E TÉCNICAS UTILIZADOS

Para definir a metodologia de pesquisa foram utilizados os conceitos de Vergara (2016), que define os critérios:

a) quanto aos fins:

- descritiva: mostra características de determinada situação, não tendo compromisso de explicar o motivo pelo qual ocorre;
- explicativa: justifica os motivos, esclarece os fatores que contribuem para que determinado fenômeno ocorra;
- aplicada: ocorre pela necessidade de resolver problemas concretos, tendo finalidade prática.

b) quanto aos meios:

- bibliográfica: o autor busca informações em materiais publicados em livros, revistas, artigos;
- pesquisa-ação: caracteriza-se por ser um tipo de pesquisa participante e pesquisa aplicada, onde o autor possui envolvimento de maneira participativa;
- estudo de caso: é caracterizado por profundidade e detalhamento, utiliza métodos diferenciados de coleta de dados.

Inicialmente buscou-se dados em literaturas, artigos e demais publicações para embasar e enriquecer o presente estudo. A pesquisa bibliográfica é o estudo sistematizado desenvolvido com base em material publicado de maneira acessível para o público em geral (VERGARA, 2016).

O objetivo deste trabalho é aplicar o *Scrum* no núcleo de tecnologia da informação, na área de personalização de *softwares*, em uma Instituição de Ensino Superior, documentando as atividades de aplicação do *Scrum* bem como suas etapas. Dessa forma, Vergara (2016), caracteriza o trabalho como um estudo de caso, por possuir caráter de profundidade e detalhamento e utiliza métodos diferenciados de coleta de dados.

A técnica adotada para desenvolver esta pesquisa em questão pode ser classificada com uma pesquisa-ação, pois caracteriza-se por ser um tipo de

pesquisa participante e aplicada, com envolvimento do autor de maneira participativa (VERGARA, 2016).

Este estudo apresenta critérios de uma pesquisa descritiva, pois expõe características de determinada situação, não tendo compromisso de explicar o motivo pelo qual ocorre. Explicativa, pois justifica os motivos, esclarece os fatores que contribuem para que determinado fenômeno ocorra. Aplicada pois, ocorre da necessidade de resolver problemas existentes, tendo finalidade prática (VERGARA, 2016).

O *Scrum* será aplicado em um projeto de implementação de *software* para atendimento a clientes de uma Instituição de Ensino Superior, o qual, recebe as tarefas diariamente, podendo ser novas implementações, incidentes ou somente solicitações, através do sistema de chamados, denominado GLPI, que é um sistema gratuito de Gerenciamento de Ativos de TI, sistema de rastreamento de problemas e central de atendimento. As atividades são distribuídas entre os integrantes, conforme níveis de conhecimento técnico e prioridades. O andamento das tarefas não é controlado formalmente, e em algumas situações o projeto é deixado de lado para iniciar outro que possui atribuição de maior prioridade.

Figura 14 - Visão geral do GLPI

The screenshot displays the GLPI web interface. At the top, there is a navigation menu with options like 'Ativos', 'Assistência', 'Gerência', 'Ferramentas', 'Plug-ins', 'Administração', and 'Configurar'. Below this, a breadcrumb trail shows 'Home > Assistência > Chamados'. The main content area is titled 'Chamado - PROJETO: Servidor - Criação das regras de bloqueio para a central de cópias - ID 29808'. It contains a form with various fields for ticket management, including 'Data de abertura', 'Data de vencimento', 'Por', 'Tipo', 'Status', 'Urgência', 'Impacto', 'Prioridade', 'Categoria', 'Origem da requisição', 'Aprovação', and 'Localização'. A sidebar on the left provides navigation options for the ticket, such as 'Chamado', 'Acompanhamentos', 'Aprovações', 'Tarefas', 'Solução', 'Estatísticas', 'Custos', 'Tarefas do projeto', 'Itens', 'Documentos', 'Problemas', 'Mudanças', 'Histórico (3)', and 'Todos'. The 'Descrição' field at the bottom contains the text: 'Criar regras para que os computadores da central de cópias tenham acesso apenas à rede local. Na internet, eles devem poder apenas receber e-mail (não enviar). Ver regras que estavam em vigência no firewall antigo.'

Fonte: NTI, 2018.

Na reunião de abertura o PO vai apresentar o projeto à equipe. Em seguida serão definidos os papéis da equipe e quais os objetivos do *sprint*.

Nesta mesma reunião, ocorrerá a distribuição das estórias, definição das ferramentas que serão utilizadas e o controle do projeto, através do gráfico *burndown*.

Para o controle de cartões das as estórias e tarefas será utilizado um quadro da ferramenta *Trello*, simulando um quadro *Kanban*, sem a necessidade de utilizar o quadro físico e, isso possibilitará aos membros do time, acesso e atualização das informações, estando em qualquer local. Com o avanço do projeto, a equipe vai ter uma visão geral sobre as atividades que precisam ser iniciadas, o integrante do time de desenvolvimento responsável por determinada estória, as tarefas que fazem parte da estória, os pontos da estória, e ainda, realizar uma comparação entre a pontuação planejada e a executada.

Os *sprints* serão encerrados na reunião de encerramento e retrospectiva, com a apresentação do produto desenvolvido e a identificação das lições aprendidas do *sprint*, visando tomar as ações para os *sprints* futuros.

Por fim, após finalizados alguns *sprints*, o time de desenvolvimento responderá a um questionário sobre a aplicação do *Scrum* no projeto.

3.2 MATERIAIS E EQUIPAMENTOS

Para a realização deste trabalho utilizou-se:

- Planilhas eletrônicas excel;
- Ferramenta Trello;
- Formulário Google;
- *Skype*;
- Computador.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

4.1 DEFINIÇÃO DE APLICAÇÃO DO SCRUM

A primeira parte do trabalho foi definir como seria aplicado o *Scrum*. Com todos os conhecimentos acerca dessa metodologia e, com um conhecimento inicial de como a equipe trabalhava, definiu-se os papéis, artefatos e eventos do *Scrum* que seriam utilizados nesse trabalho. Na sequência deste capítulo apresenta-se todos os itens da metodologia utilizada, explicando os motivos de sua utilização, bem como a forma de utilização no trabalho.

4.1.1 Papéis

Conforme citado no capítulo 2, item 2.4.1.1, o *Scrum* possui os seguintes papéis: *Product Owner*, *Scrum Team* e *Scrum Master*. Os papéis possuem grande importância para atender as principais características de uma equipe que trabalha com *Scrum* que são: criatividade; produtividade e flexibilidade. Por isso o primeiro passo do trabalho foi esclarecer para a equipe as responsabilidades de cada papel e definir quais membros do time assumirão os mesmos.

4.1.1.1 *Product Owner*

É o dono do produto, responsável por fazer a comunicação entre o cliente e a equipe do projeto. Fornece ao time de desenvolvimento os requisitos do produto esclarecendo suas dúvidas. Além disso, também realiza o gerenciamento do *Product Backlog*. Na aplicação deste projeto, o PO é o coordenador do NTI.

4.1.1.2 *Scrum Team*

O time de desenvolvimento é responsável por desenvolver e entregar a ideia do produto solicitada pelo PO, caracteriza-se por ser multidisciplinar, e auto organizado. Realiza a distribuição das tarefas entre os membros e planeja o tempo necessário para realizá-las utilizando o *planning poker*.

Neste projeto, o time de desenvolvimento foi de dois colaboradores que integram o NTI. Além do projeto, é importante salientar que o time continuou realizando atividades rotineiras.

4.1.1.3 *Scrum Master*

É o facilitador da equipe, é responsável por garantir que o *Scrum* seja aplicado de maneira correta, participa de todos os eventos definidos pela metodologia.

No projeto, a *Scrum Master* não faz parte dos colaboradores da Instituição onde o projeto foi aplicado. O papel de *Scrum Master* ficou a cargo da autora desta pesquisa.

4.1.2 Artefatos

A partir da aplicação do *Scrum*, a equipe de projeto iniciou a utilização de artefatos. O artefato utilizado neste trabalho foi o *sprint backlog*. É através dos artefatos que a equipe consegue identificar a necessidade de adaptação e inspeção do projeto.

4.1.2.1 *Product Backlog*

O *product backlog* (PB) é uma lista de histórias (itens) que precisam ser desenvolvidos ao longo do projeto para entregar o produto pronto. Ao longo dos *sprint*, o PB acaba se tornando o *sprint backlog*.

4.1.2.2 *Sprint Backlog*

É a lista de histórias do PB para ser desenvolvidas e entregues ao final de cada *sprint* e que se incorporam ao produto final. O quadro 5 apresenta o modelo de *sprint backlog*.

Neste modelo de lista de histórias, estão identificadas as datas de início de *sprint*, a data e hora e local da demonstração do *sprint*, os objetivos, as histórias e os pontos estimados, ao final será realizada a soma desses pontos e essa será considerada a velocidade estimada do *sprint*.

Quadro 5 - Modelo de *sprint backlog*

LISTA DE ESTÓRIA		
Data de Início: --/--	Data e Hora da Demonstração: --/--/-- xx:xx	Local da Demonstração: xx
SPRINT	ESTÓRIA	PONTOS ESTIMADOS
Objetivos - Sprint: x		
	VELOCIDADE ESTIMADA	

Fonte: Adaptado de Kiniberg, 2007.

4.1.3 Eventos

Com a aplicação do *Scrum*, a equipe do projeto integrou à sua rotina os eventos. Conforme mencionado anteriormente, no item 2.4.1.3, os eventos são: planejamento do *sprint*, reunião diária, revisão do *sprint* e retrospectiva do *sprint*. Esses eventos têm objetivo de minimizar a necessidade de reuniões que não estão programadas.

4.1.3.1 Planejamento do *sprint*

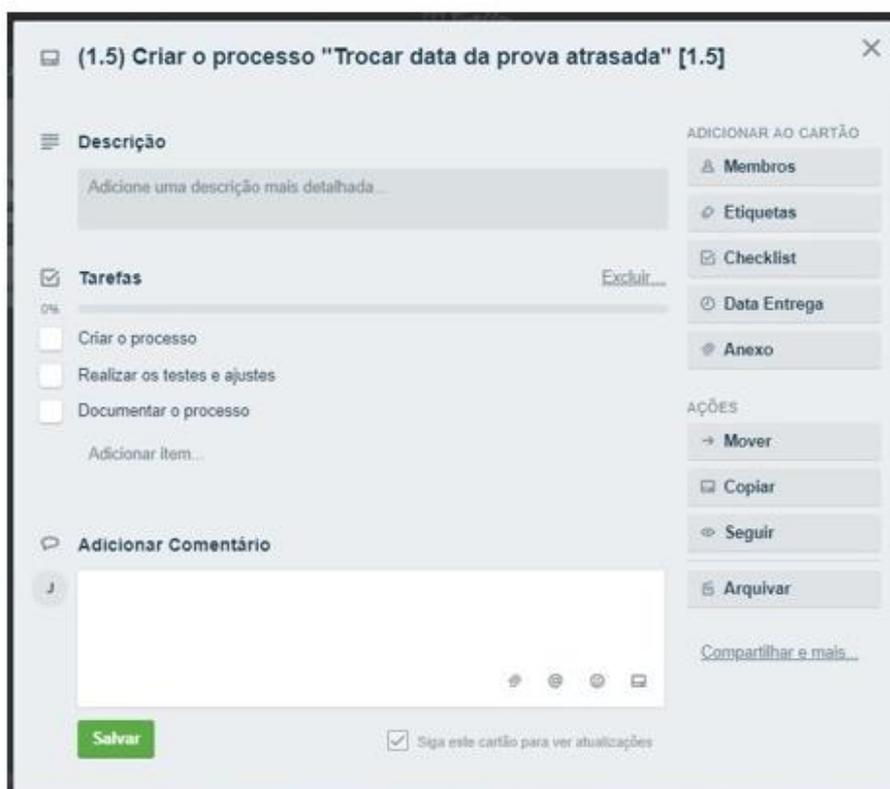
É nesta etapa que o *sprint* é planejado, com a participação de todos os integrantes do projeto. Inicialmente, o PO informa as funcionalidades que precisam ser executadas durante o *sprint* para que o objetivo deste seja atingido e quando incluídas no *sprint* tornam-se o *sprint backlog*.

Neste projeto, o PO define o tempo diário que cada integrante terá disponível para dedicar-se ao projeto. Conforme mencionado anteriormente, além do projeto, o time também executa atividades rotineiras do NTI e que não integram o projeto.

Após identificação das estórias, deve-se montar o quadro *Kanban* na ferramenta Trello, conforme mencionado anteriormente. Após concluída esta parte da reunião, o time de desenvolvimento e o *Scrum Master* se reúnem para realizar o planejamento do tempo necessário para realizar as estórias utilizando o *planning poker*.

As tarefas que incorporam as estórias devem ser adicionadas ao cartão através do item “*check-list*”. No modelo da figura 15, identifica-se que a estória “Criar o processo “Trocar data da prova atrasada” possui 3 tarefas, que são: “Criar o processo”; “Realizar os testes e ajustes”; “Documentar o processo”. A estória somente estará concluída para todas tarefas estarão concluídas.

Figura 15 – Modelo de identificação dos itens de um cartão



Fonte: A autora, 2018.

4.1.3.2 Reunião diária

É uma reunião diária, de 15 minutos onde o time de desenvolvimento e o *Scrum Master* se reúnem preferencialmente sempre no mesmo local e horário. Esta reunião ocorrerá quando o desenvolvimento das estórias do *sprint* é iniciada. Um

dos objetivos desta reunião é proporcionar maior visibilidade e comunicação e avaliar o andamento daquilo que foi planejado. Nesta reunião, o time de desenvolvimento responde às seguintes perguntas referentes às suas atividades no projeto:

- 1) O que eu fiz ontem?
- 2) O que vou fazer hoje?
- 3) Há algo que está impedindo o andamento da minha atividade?

No caso de existir algum impedimento o *Scrum Master* será responsável por removê-lo e, assim, a equipe poderá continuar suas tarefas conforme o planejado.

Essa reunião deve ser realizada com todos os integrantes em pé, conforme sugere-se na metodologia XP, isso favorece que a reunião ocorra de forma rápida e com assuntos focados no essencial.

Finalizada a reunião, realiza-se a atualização dos pontos das estórias concluídas no gráfico *burndown*.

4.1.3.3 Revisão do *sprint*

Ao final de cada *sprint*, o time de desenvolvimento realiza a apresentação do produto para o PO os itens do *sprint backlog* concluídos. Essa reunião é conduzida pelo *Scrum Master*.

4.1.3.4 Retrospectiva do *sprint*

Nesta reunião, avalia-se os aspectos positivos e negativos do *sprint*, identificados no projeto como “lições aprendidas”. As lições são importantes para que sejam criadas ações necessárias para que aspectos negativos não ocorram novamente ou sejam minimizados, como também potencializar aquilo que ocorreu de forma positiva nos próximos *sprints*. Essas lições aprendidas serão atualizadas em planilha *excel* no Drive, na mesma pasta do gráfico *burndown*.

4.1.4 *Kanban*

Conforme apresentado no item 2.10, a utilização do sistema *Kanban* permite um controle detalhado para a equipe manter-se atualizada do andamento do *sprint*. No projeto, definiu-se por utilizar a ferramenta Trello pois trata-se de um sistema

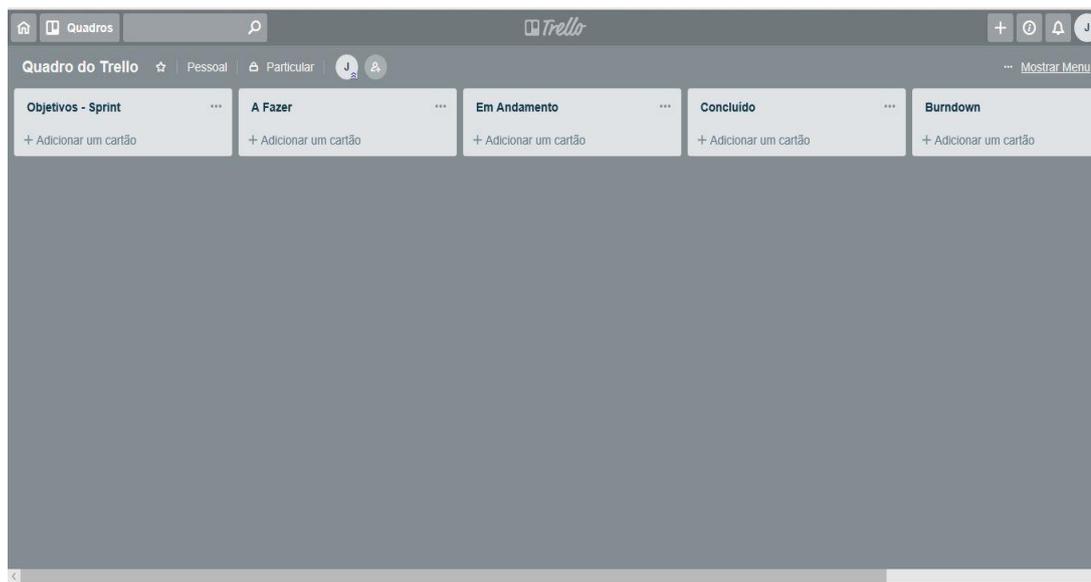
informatizado e possui uma interface própria para o gerenciamento detalhado dos *sprints*. Foi utilizada a versão gratuita da ferramenta que está disponível *on-line*. Os integrantes da equipe tiveram que realizar cadastro *on-line* para ter acesso a mesma.

A ferramenta possui várias funcionalidades para personalização dos cartões. No projeto, será utilizado um quadro para cada *sprint*, onde serão incluídos cartões, sendo que cada um desses representa uma estória definida no *sprint backlog*. Ainda, em cada cartão será identificado o(s) responsável(is) por cada estória. As tarefas que compõe as estórias serão incluídas em um *check-list* denominado “tarefas”. Quando uma tarefa é concluída, a mesma será marcada e fica com *status* “concluída”, identificando-se em porcentagem (%) a quantidade de tarefas concluídas no cartão.

Com a utilização do item “descrição” pode-se incluir informações pertinentes à estória, como por exemplo informações adicionais e o tempo utilizado para realizar a estória. A partir desta informação atualiza-se o gráfico *burndown*. Uma das ações da ferramenta, é a possibilidade de “seguir” o cartão, onde as atualizações realizadas neste são também atualizadas por e-mail do integrante que desejaram seguir o referido cartão. Quando uma estória muda de etapa: de “A Fazer” para “Em andamento”, ou “Em andamento” para “Concluído” o membro responsável move este cartão para a respectiva fase.

Vale ressaltar, que neste projeto, as atividades do *sprint backlog* que ainda não foram iniciadas compõe a fase “A Fazer”, quando a estória é iniciada será movida para “Em andamento” permanecendo nesta etapa até que todas suas funcionalidades são testadas. Somente quando a estória está totalmente concluída, o cartão será movido para “Concluído”. Essas atualizações serão realizadas pelo membro da equipe no momento que iniciar ou concluir uma tarefa, não sendo necessária a atualização somente no momento da reunião diária.

A figura 16 apresenta o modelo de quadro *Kanban*, a partir da ferramenta Trello, proposto para o acompanhamento dos *sprints* do projeto.

Figura 16 - Modelo de quadro *Kanban*

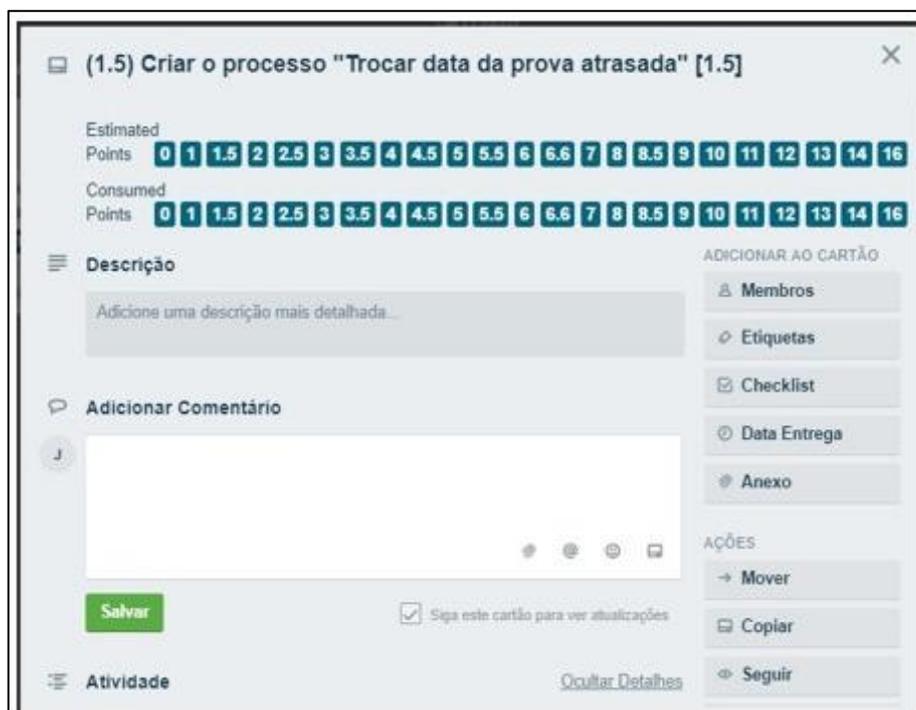
Fonte: A autora, 2018.

A pontuação prevista para cada estória conforme mencionado anteriormente, será incluída em cada cartão, essa configuração é possível pelo fato do Trello “possuir extensão” para essa funcionalidade. A pontuação prevista pelo time é selecionada em “*estimated points*” (pontos estimados). A extensão possui pontuação padrão ou personalizada, no projeto a pontuação foi personalizada para ter maior precisão, de acordo com cada estória.

À medida em que esta é desenvolvida, o membro do time atualiza o tempo no item “descrição” e quando a tarefa é concluída soma-se o tempo e inclui-se em “*consumed points*” (pontos consumidos), pela *Scrum Master*. Desta forma, ao abrir o quadro de um *sprint* realiza-se uma rápida identificação do tempo planejado e o tempo utilizado para realizar determinada estória.

É possível identificar na figura 17, que a estória “Criar processo “Trocar data da prova atrasada”, possui pontuação estimada de 1.5 e quando a estória estiver totalmente completa será inserido o tempo que realmente foi necessário para realizar a estória, nesta, o tempo foi o mesmo do estimado.

Figura 17 - Identificação do tempo de uma estória no Trello



Fonte: A autora, 2018.

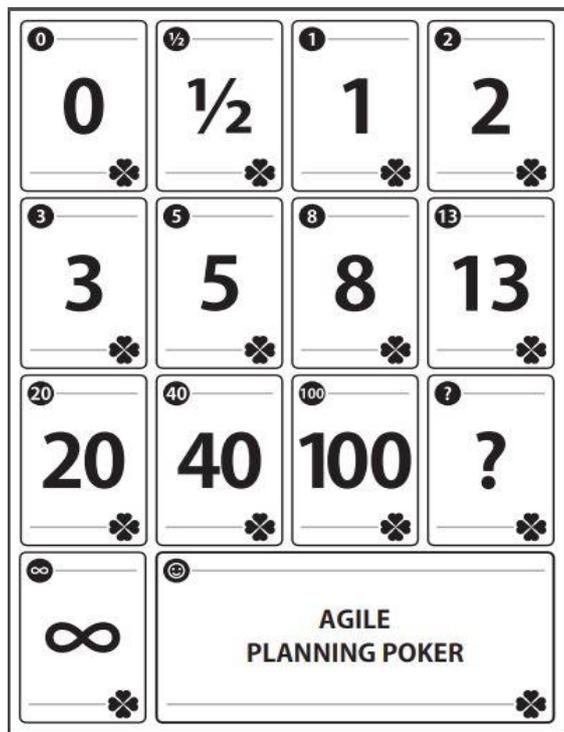
4.1.5 *Planning poker*

A partir das listas de estórias definidas no *sprint planning*, o time de desenvolvimento irá realizar a estimativa de tempo para concluí-las. Será utilizado o método *Fibonacci* de pontuação com cartas com os seguintes valores: 0, 0.5, 1, 2, 3, 5, 8, 13, 21, 100, e “infinito”. Para facilitar controle e organização da pontuação, 1 ponto será o equivalente à 1 hora de trabalho.

Cada integrante do time de desenvolvimento, após a identificação da estória em “A Fazer”, define o tempo necessário para realizá-la. Com isso, integrantes com diferentes habilidades e velocidades sugerem uma pontuação. Caso ocorra diferença na pontuação, cada integrante expõe sua opinião sobre os passos para realizar a estória e sugere a pontuação até chegar em consenso.

A figura 18 mostra o *planning poker*, método utilizado para a pontuação das estórias dos *sprint* 2, 3, 4 e 5 do projeto.

Figura 18 - Método de pontuação de estórias



Fonte: Adaptado de Kiniberg, 2007.

4.1.6 Gráfico *burndown*

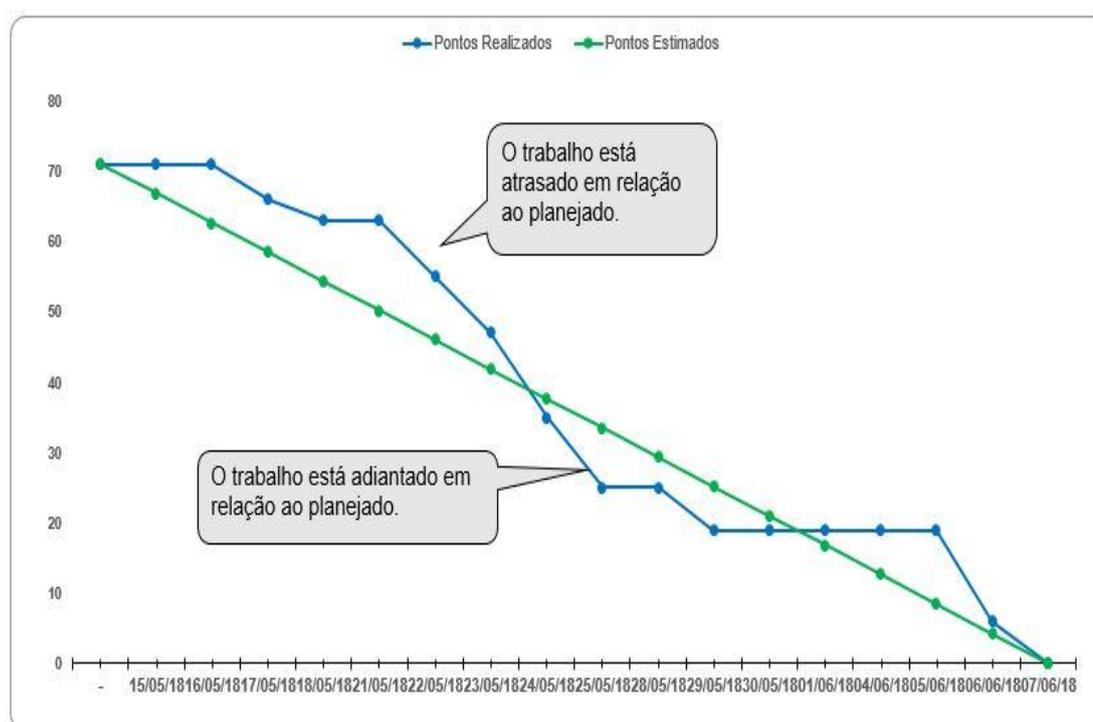
O gráfico *burndown* é atualizado diariamente para comparar a quantidade de trabalho planejada e o progresso do time de desenvolvimento em relação ao planejado. A partir da pontuação definida no *planning poker*, as estórias são somadas e correspondem ao valor total do gráfico, localizado no eixo Y. No eixo X, estão definidos os dias necessários para a conclusão do *sprint*. Conforme mencionado anteriormente, os dias necessários para a conclusão do *sprint* são resultado da divisão dos pontos totais pelo tempo diário disponível time de desenvolvimento. À medida que as estórias são concluídas, a pontuação total é subtraída até chegar ao ponto zero. Nos dias em que nenhuma estória for concluída a pontuação a ser atualizada no gráfico é zero.

Para compor o gráfico definiu-se criar uma tabela com as seguintes informações: na coluna “data” estão incluídos os dias de execução do *sprint*; na coluna “pontos planejados/dia” a pontuação diária prevista durante a execução do *sprint*, sendo que essa pontuação é resultado dos “pontos estimados” dividido pelas horas diárias que o time de desenvolvimento dispõe para desenvolver o projeto; já

os “pontos realizados/dia” representa a pontuação das estórias finalizadas no dia; os “pontos realizados” referem-se ao valor total da pontuação prevista para o *sprint* menos a pontuação realizada diariamente da coluna “pontos realizados/dia”; e, por fim, na coluna “pontos estimados” estão representados os valores da pontuação estimada no *planning poker* menos a soma dos pontos “planejados/dia”. A tabela modelo criada para esse trabalho encontra-se no apêndice A.

A figura 19, apresenta o modelo do gráfico *burndown* que será utilizado para analisar o andamento do projeto, durante as reuniões diárias.

Figura 19 - Gráfico *burndown*



Fonte: A autora, 2018.

4.2 ANÁLISE DOS RESULTADOS

A aplicação do *Scrum* iniciou-se com a definição do projeto e dos membros da equipe do projeto. Esta definição foi realizada pelo PO.

No quadro 6, estão descritos os papéis dos membros da equipe antes da aplicação do *Scrum* e após sua aplicação.

Quadro 6 - Descrição dos papéis antes e depois da aplicação do *Scrum*

Quem	Papel antes do <i>Scrum</i>	Papel com o <i>Scrum</i>
Fabrcio	Coordenador do núcleo de Tecnologia da Informação	<i>Product Owner</i>
Felipe	Programador	Time de desenvolvimento
Luís	Analista	Time de desenvolvimento
Jaqueline	Estudante	<i>Scrum Master</i>

Fonte: A autora, 2018.

O projeto em questão, o qual foi aplicado o *Scrum*, é o CRM (*Customer Relationship Management* / Gestão de Relacionamento com o Cliente). O objetivo do projeto era de possibilitar a utilização de solicitações acadêmicas através do módulo de CRM existente no ERP Educacional utilizado pela instituição. Como objetivo específico do projeto estabeleceu-se a necessidade de criar várias solicitações on-line que os estudantes poderão abrir via Portal Acadêmico e que seguirá um fluxo de processos que poderá envolver coordenadores de curso, professores, direção e secretaria acadêmica, facilitando a vida dos estudantes, uma vez que não precisarão mais ir fisicamente na central de atendimento e aguardar em filas, bem como possibilitar o acompanhamento das solicitações, a qualquer momento.

4.2.1 *Sprint* 1

O planejamento do *sprint* 1 foi iniciado após a reunião de abertura do projeto. Contou com a presença da equipe do projeto. O PO apresentou o objetivo do *sprint* e as histórias que seriam desenvolvidas e entregues pelo time de desenvolvimento ao final do *sprint*. O objetivo deste foi: criar solicitações de provas atrasadas pelo Portal Acadêmico.

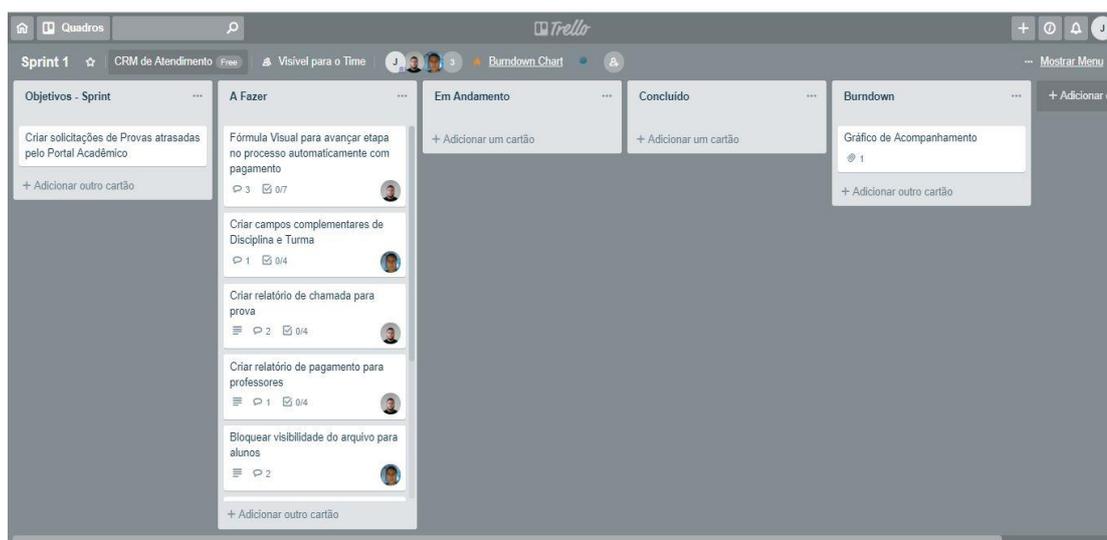
Durante a reunião, a equipe e o PO, analisaram tudo que era necessário ser realizado para entregar o solicitado, definiram 9 histórias, às quais, estão descritas no APÊNDICE B.

Neste primeiro *sprint*, o PO definiu que cada integrante do time de desenvolvimento teria 5 horas diárias para desenvolver as histórias do *sprint*.

Após a definição das estórias, o *Scrum Master* e o time de desenvolvimento reuniram-se para dividir estas em tarefas, bem como definir o responsável por realizá-las. Vale ressaltar que a *Scrum Master*, neste momento, não possui participação opinativa na pontuação, somente atualiza a lista de estórias e as tarefas na planilha *excel* e na ferramenta Trello.

Conforme mencionado anteriormente, para realizar o controle visual das estórias do projeto adotou-se a ferramenta Trello, pois se trata de um sistema informatizado e possui uma interface própria para o gerenciamento detalhado dos *sprints*. Este quadro pode visto na figura 20, no qual consta o objetivo do *sprint*, as estórias e pontuação, as tarefas e o acesso ao gráfico *burndown*.

Figura 20 – Quadro de acompanhamento *sprint* 1



Fonte: A autora, 2018.

Ainda, nesta reunião, em conjunto com o time de desenvolvimento definiu-se que as reuniões diárias seriam realizadas às 13h30min, no laboratório de informática 01. Além disso, definiu-se que se algum integrante não poderá participar da reunião no horário estabelecido, o mesmo torna-se responsável por informar aos demais esta situação para definir-se um novo horário para aquele dia.

Os integrantes do time de desenvolvimento informaram à *Scrum Master* que esporadicamente realizam suas atividades na outra unidade da instituição localizada em outro endereço, nesta situação, para não causar impacto de deslocamento dos demais integrantes, ou a possibilidade da reunião diária não ser realizada, definiu-se

que a mesma seria realizada por *Skype*. No APÊNDICE B é possível visualizar a lista de estórias, a ID (número de identificação de cada tarefa), as tarefas e o responsável pelo desenvolvimento das estórias que integram o *sprint* 1.

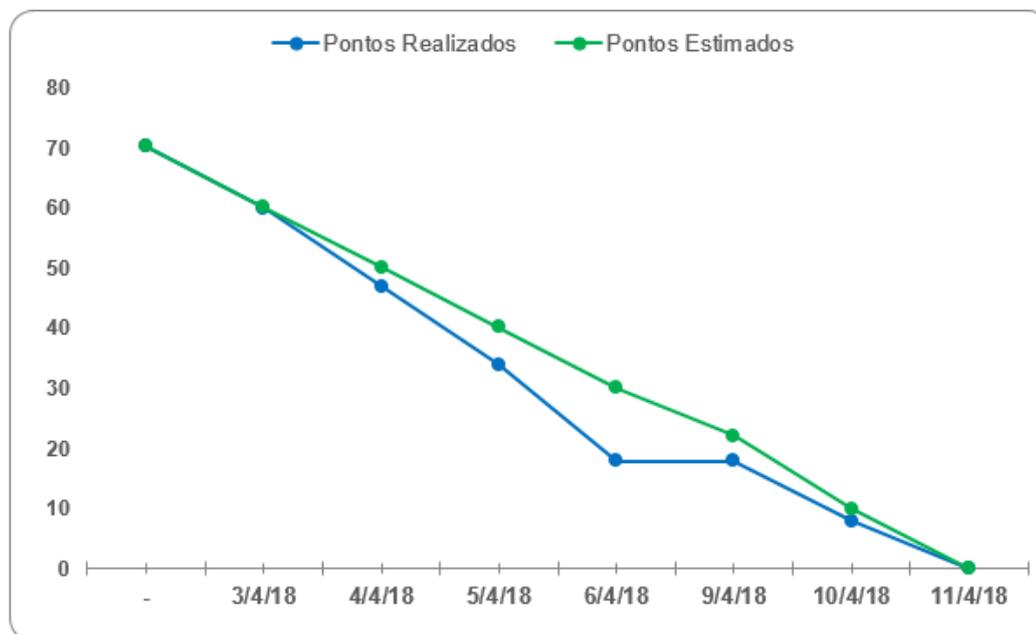
As reuniões diárias iniciaram um dia após o planejamento do *sprint*, dia 03/04/2018, em local e horário definidos anteriormente. Nestas reuniões, utilizou-se projetor multimídia possibilitando a visualização das informações do andamento do projeto no Trello, sendo que a *Scrum Master* é a facilitadora da reunião. Ainda, nesta reunião, o time de desenvolvimento respondeu ao seguinte questionário:

- 1) Pergunta 1: “o que eu fiz ontem para cumprir a meta do *sprint*”;
- 2) Pergunta 2: “o que vou fazer hoje para cumprir a meta do *sprint*”,
- 3) Caso identificado algum impedimento seria relatado para tomar as decisões o mais rápido possível para não impactar no resultado do *sprint*.

É importante informar que esse modelo de reunião foi utilizado nas demais reuniões diárias do projeto.

Conforme mencionado anteriormente, o *planning poker* do *sprint* 1 não foi realizado e por isso, o gráfico *burndown* foi atualizado diariamente de acordo com as estórias concluídas. Sendo que a análise do andamento das atividades do time foi realizada a partir das horas diárias disponíveis, um total de 10 horas/dia, e o número de dias para conclusão do *sprint*, 5 dias. Desta forma o time de desenvolvimento teve 10 horas diárias nos 4 primeiros dias e 8 horas no último dia, pois a apresentação do *sprint* foi marcada para o dia 09/04/2018, no turno da tarde. A figura 21 representa o gráfico do andamento do *sprint* 1.

Conforme pode ser identificado no gráfico, da figura 21, a linha na cor verde representa a estimativa de tempo do *sprint*. Já a linha na cor azul representa o progresso diário da equipe, ou seja, o trabalho efetivo realizado, desta forma é possível comparar o progresso do *sprint*. A linha na cor verde foi calculada com base nos dias disponíveis multiplicado pelas horas disponíveis do time de desenvolvimento para desenvolver as estórias.

Figura 21 - Gráfico *burndown* do *sprint* 1

Fonte: A autora, 2018.

Durante o andamento do *sprint*, no dia 06/04/2018, foi realizada a atualização da versão do sistema, sendo que a empresa responsável esqueceu de executar uma parte do processo de atualização no ambiente de testes, devido a isso o time de desenvolvimento ficou sem acesso a este ambiente até o dia 09/04/2018, causando atraso no andamento do *sprint* e a necessidade de realizar as estórias em um ambiente paralelo, aguardando que a atualização fosse realizada com sucesso. Sendo assim, encerrou-se as estórias do *sprint* 1 no dia 11/04/2018.

A revisão do *sprint* ocorreu no mesmo dia, à tarde, com a presença de todos os integrantes do projeto. Nesta reunião o time de desenvolvimento demonstrou as estórias implementadas, simulando todos os processos envolvidos, garantindo que as estórias foram concluídas e estão em funcionamento. Em seguida o PO deu o aceite, informando que o objetivo do *sprint* 1 foi concluído.

Finalizada a demonstração e as considerações do PO, realizou-se a retrospectiva do *sprint*, para analisar os pontos positivos e negativos deste. As lições aprendidas foram:

- A utilização da ferramenta Trello facilitou a visualização do andamento do projeto. Sugeriu-se a utilização desta ferramenta por ser *on-line*, onde todos os integrantes do projeto tiveram acesso para atualizar o andamento das estórias e tarefas. A utilização do Trello trouxe maior organização e visualização do andamento das atividades. Para os próximos *sprints* a ferramenta continuará sendo utilizada;
- As reuniões diárias auxiliaram na comunicação dos membros do Time de desenvolvimento. Estas reuniões compõe os eventos do *Scrum*, sendo possível o compartilhamento de experiências ao desenvolver determinada tarefa. Conforme mencionado anteriormente, as reuniões diárias integram os eventos do *Scrum* e serão mantidas nos próximos *sprint*;
- O time de desenvolvimento deslocou-se para outra sala para desenvolver as estórias do *sprint*. Na reunião de planejamento do *sprint* o PO sugeriu para o time de desenvolvimento para que se deslocassem para outra sala para trabalhar no projeto. O time destacou que a ausência de interrupção proporcionou maior concentração e agilidade no desenvolvimento das estórias e tarefas. Para os próximos *sprints* este procedimento será mantido;
- Os integrantes do time de desenvolvimento realizaram as estórias no mesmo horário e sala. Essa decisão foi do próprio time, o que facilitou a comunicação, troca de experiências e sugestões entre o time. Para os próximos *sprints* isso será mantido;
- Não havia instrução completa no Manual da TOTVS, fazendo-se necessário buscar soluções para as tarefas e com isso utilizou-se mais tempo para solucionar as tarefas. Entende-se que para os próximos *sprints* isso só voltará a acontecer se a tarefa foi inédita;
- Atualização, não comunicada, do sistema no final de semana que antecedeu a entrega do projeto, a qual, foi uma decisão da mantenedora, ocasionou que o atraso de dois dias na finalização do *sprint*. Para esta situação definiu-se melhorar a comunicação com a mantenedora, para que informe o NTI antecipadamente sobre a atualização do sistema.

4.2.2 *Sprint 2*

O *sprint 2* foi iniciado na reunião de encerramento da *sprint 1*, com a presença do time do projeto. O objetivo deste foi “Acompanhar as solicitações das provas

atrasadas pelo Sistema que entrou em Produção e identificar pontos que podem ser melhorados e, implementar o processo executado pela Secretaria”.

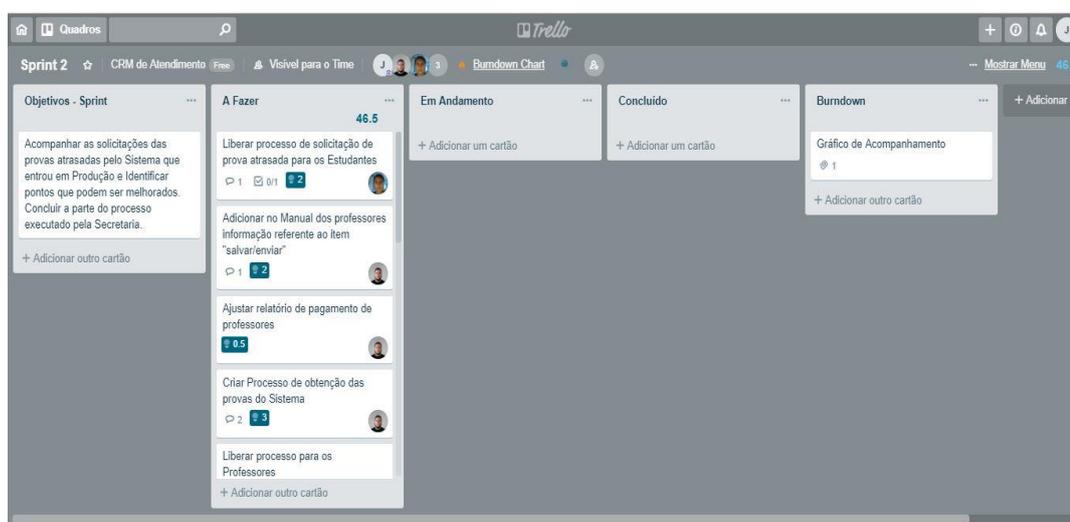
Ainda, nessa reunião, o PO definiu que o tempo diário para desenvolver o *sprint* seria 3 horas e também que logo que as estórias estivessem concluídas seria marcada a reunião de revisão e de retrospectiva do *sprint*.

Com as estórias definidas, o *Scrum Master* e o time de desenvolvimento reuniram-se para realizar o detalhamento destas que resultaram nas tarefas.

Feito isso, realizou-se a pontuação das estórias utilizando-se a técnica *planning poker*. Conforme mencionado anteriormente, este foi o primeiro *sprint* em que foi utilizada essa técnica, pois no *sprint 1* o time de desenvolvimento não tinha experiência prática com o tipo de personalização de *software* que iriam realizar. Desta forma, a *Scrum Master* explicou o funcionamento da pontuação das cartas.

A pontuação atribuída foi incluída no cartão correspondente, na ferramenta Trello, esta realiza a soma automática da pontuação e a deixa visível no canto superior direito do quadro. A pontuação total das estórias desse *sprint* foi 46,5 pontos. Na figura 22 pode ser identificado: o objetivo do *sprint 2*, composto por 13 estórias; a pontuação de cada estória, as tarefas e o responsável pelo desenvolvimento, os pontos estimados para cada estória e o acesso ao gráfico *burndown*.

Figura 22 – Quadro de acompanhamento *sprint 2*



Fonte: A autora, 2018.

Ao final da reunião o *Scrum Master* definiu com o time de desenvolvimento que o horário e o local de realização das reuniões diárias seriam o mesmo do *sprint* 1 (no laboratório de informática 01) às 13h30min.

Conforme pode ser identificado na figura 23, para a estória “Liberar processo de solicitação de prova atrasada para os Estudantes”, foi atribuída 2 pontos, ou seja, serão necessárias duas 2 horas para realizar esta estória. A tarefa que compõem a estória é “Enviar e-mail para os estudantes informando o novo processo de solicitação de provas atrasadas”.

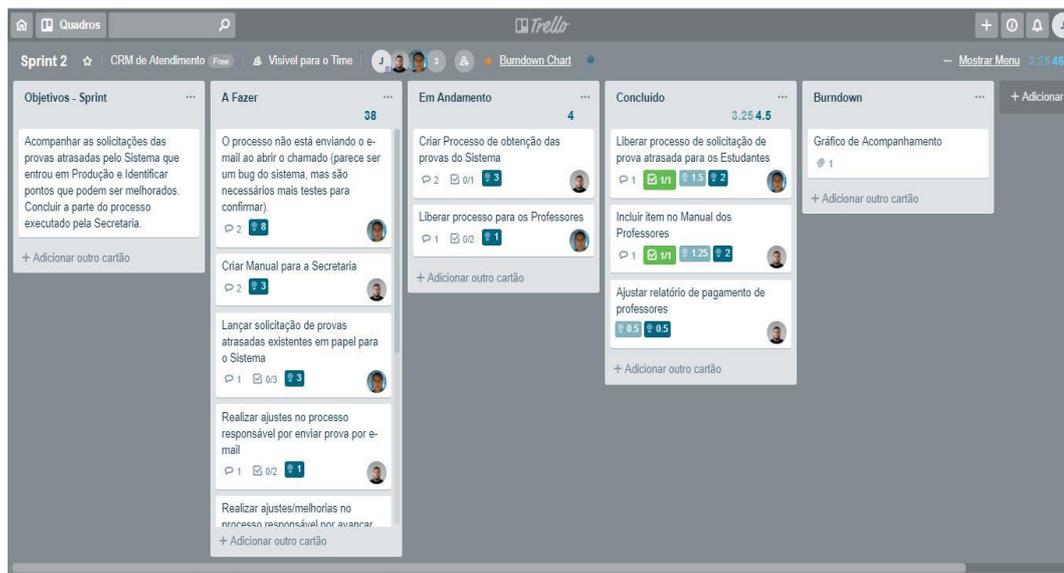
Figura 23 - Pontuação de uma estória do *sprint* 2



Fonte: A autora, 2018.

A figura 24 representa o andamento das estórias do Sprint 2, que foi atualizado no quadro Trello. Neste é possível identificar que há duas estórias em andamento e três estórias concluídas, bem como tempo necessário para desenvolvê-las.

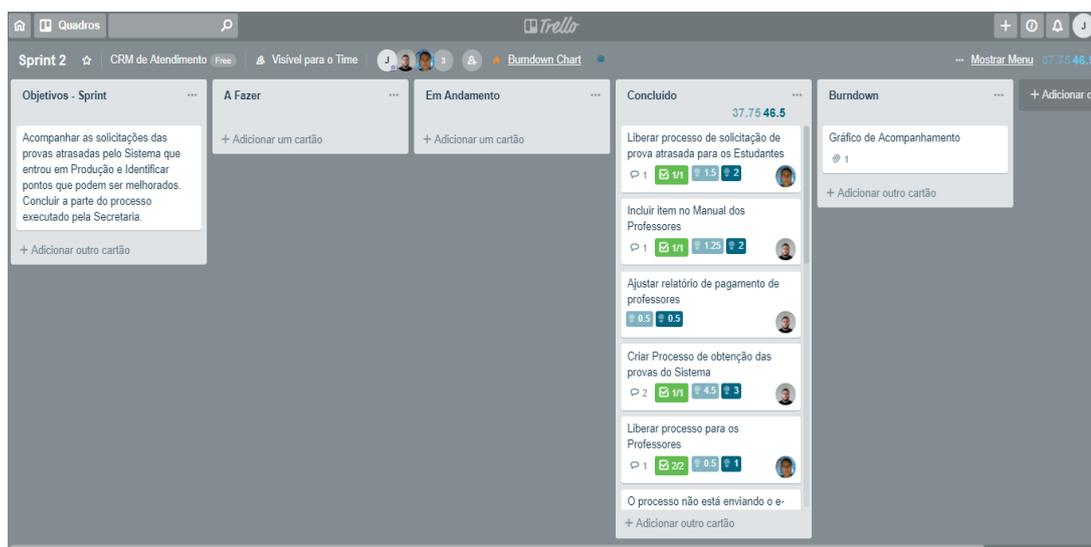
Figura 24 - Andamento do *Sprint 2* no quadro Trello



Fonte: A autora, 2018.

Na figura 25, é possível identificar a conclusão do *Sprint 2*. Conforme mencionado anteriormente, o *Sprint* é concluído quando todas as estórias foram desenvolvidas e testadas.

Figura 25 - Quadro Trello após a conclusão do *Sprint 2*

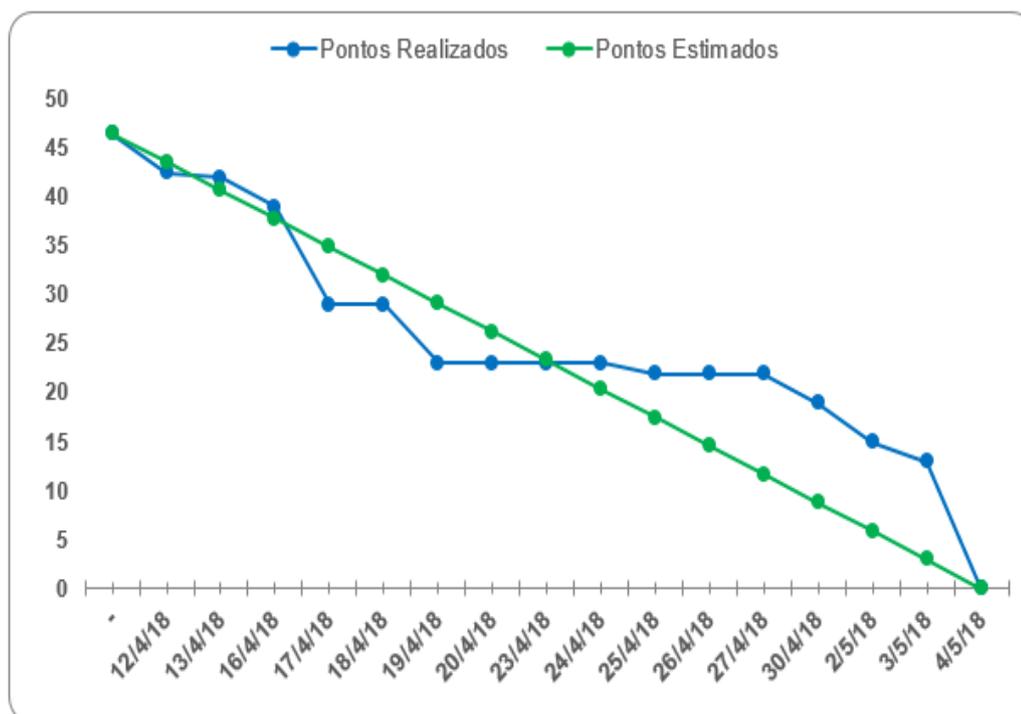


Fonte: A autora, 2018.

No APÊNDICE C estão identificadas a lista de estórias, a ID, as tarefas, o responsável por desenvolvê-las e a pontuação estimada de cada estória do *sprint 2* e a pontuação total das estórias do *sprint* – “velocidade estimada”.

Conforme mencionado anteriormente, a pontuação do *sprint* 2 foi 46,5 pontos. A evolução da execução das estórias foi registrada no gráfico *burndown* diariamente, o que proporcionou a visualização de seu andamento. O gráfico *burndown* do *sprint* 2 está representado na figura 26.

Figura 26 - Gráfico *burndown* do *Sprint* 2



Fonte: A autora, 2018.

O gráfico representado na figura 26 nos mostra que do dia 17/04 até o dia 20/04 o time de desenvolvimento estava adiantado em relação ao projeto, no dia 23/04 a pontuação do planejado *versus* realizado era a mesma, já no dia 24/04 os pontos estimados para a data eram 20,40 e o realizado pela equipe foi de 23,00 indicando que a equipe estava 2,6 pontos atrasada em relação ao planejado. Porém, a estória "Acompanhar a solicitação das provas" estimada em 13 pontos foi realizada diariamente pelo time de desenvolvimento a partir do dia 12/04, sendo que esta pontuação somente foi atualizada no gráfico no último dia do *sprint*, momento em que foi totalmente concluída. Nesta estória acompanhou-se a solicitação das provas realizada pelos acadêmicos, a disponibilização de provas por parte dos professores, o acesso e impressão das provas por parte da secretaria. Essa pontuação maior

instiga a ideia de atraso do projeto, porém, nas reuniões diárias o time de desenvolvimento mostrou que as estórias seguiam o planejado.

A revisão do *sprint* ocorreu no dia 10/05/2018, onde as estórias concluídas foram apresentadas pelo time de desenvolvimento, através da simulação de todos os processos envolvidos, garantindo que as estórias foram concluídas e estão em funcionamento. Após essa apresentação o PO informou que o objetivo do *sprint 2* havia sido concluído.

Ao término do *sprint 2* foi realizada a reunião de retrospectiva com o intuito de fazer as anotações das lições aprendidas, essas lições estão descritas abaixo:

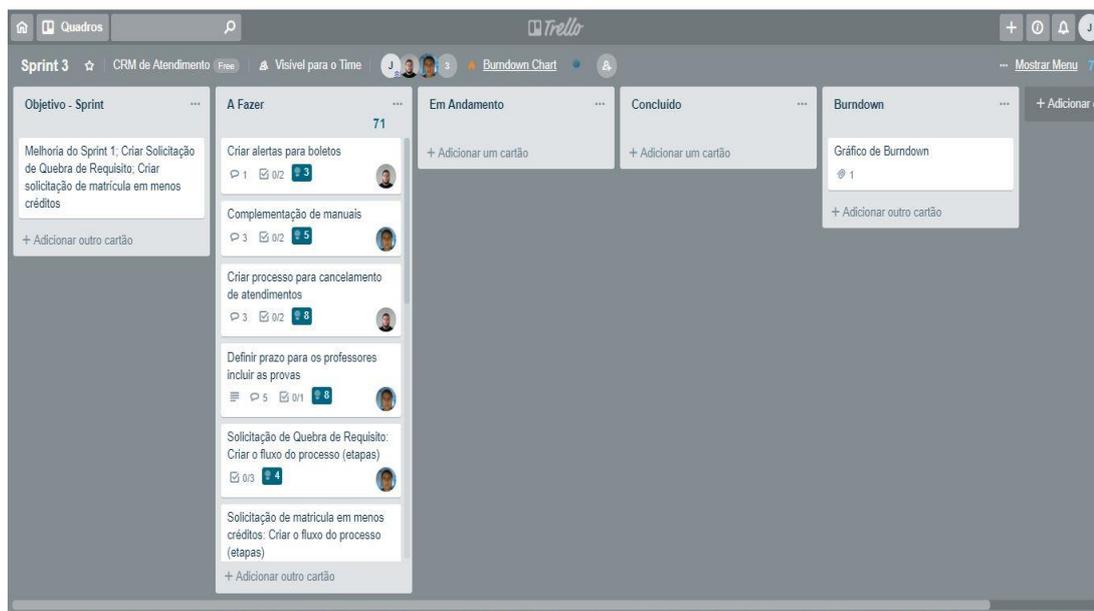
- Ocorreram falhas de segurança no processo de “salvar/enviar” prova, pois as instruções contidas no manual não foram seguidas. Ao identificar este problema incluiu-se uma *trigger* para avançar etapa. Para os próximos *sprints* deverão ser realizados processos que não dependam do usuário;
- O time de desenvolvimento avaliou de forma positiva a utilização do *planning poker* pois foi possível estipular uma pontuação para as estórias e realizar o acompanhamento durante o *sprint*.

4.2.3 Sprint 3

No dia 10/05/2018, foi realizada a reunião de abertura do *sprint 3*. No evento estavam presentes o PO que realizou a abertura da reunião e na mesma aproveitou para apresentar o objetivo do *sprint*: “Melhoria do *sprint 1*, criar solicitação de quebra de requisito; criar solicitação de matrícula em menos créditos”. Nesse *sprint*, o tempo diário para a time desenvolver as estórias foi de 4 horas. Definiu-se também que a reunião de encerramento seria marcada após a conclusão do mesmo.

Após a definição do objetivo do *sprint* e das estórias, o time de desenvolvimento e o *Scrum Master* reuniram-se para realizar o *planning poker*. O *sprint 3* possui 13 estórias e sua pontuação total somou 71 pontos.

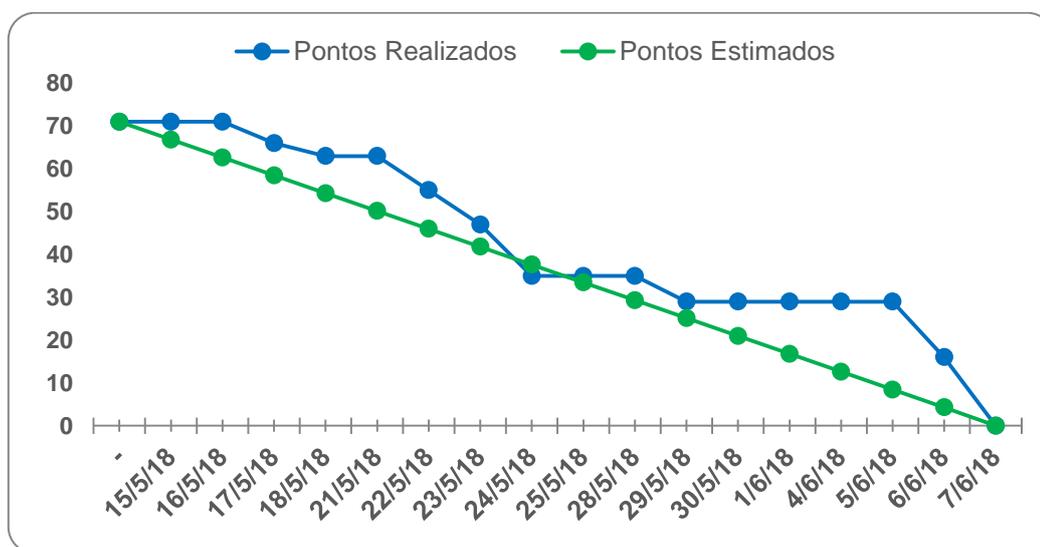
Ao final da reunião a *Scrum Master* definiu com o time de desenvolvimento que o horário e o local para realizar as reuniões diárias será mantido o mesmo dos *sprint* anteriores: às 13h30min no laboratório de informática 01. O quadro de acompanhamento do *sprint 3* está representado na figura 27.

Figura 27 – Quadro de acompanhamento do *sprint* 3

Fonte: A autora, 2018.

No APÊNDICE D estão identificadas as estórias, a ID e as tarefas, o responsável pelas estórias e a pontuação do *sprint* 3, e por fim, a soma dos pontos, que representa a velocidade do *sprint*.

O *sprint* teve 71 pontos, que foi inserido em uma tabela para gerar e atualizar o gráfico *burndown* diariamente. Este gráfico está representado na figura 28.

Figura 28 - Gráfico *burndown* do *sprint* 3

Fonte: A autora, 2018.

Conforme pode ser identificado no gráfico, até o dia 23/05/2018 o *sprint* estava atrasado, isso justifica-se, pois, o desenvolvimento deste teve início efetivo com dois dias de atraso devido a um novo problema no sistema da instituição. Somente no dia 24/05/2018 a pontuação realizada estava à frente da planejada.

Na reunião de revisão do *sprint* que ocorreu no dia 07/06/2018. Da mesma forma que na revisão dos *sprints* anteriores, o time de desenvolvimento demonstrou as estórias concluídas, simulando todos os processos envolvidos, garantindo que as estórias foram concluídas e estão em funcionamento. Após essa apresentação o PO informou que o objetivo do *sprint* 3 foi alcançado.

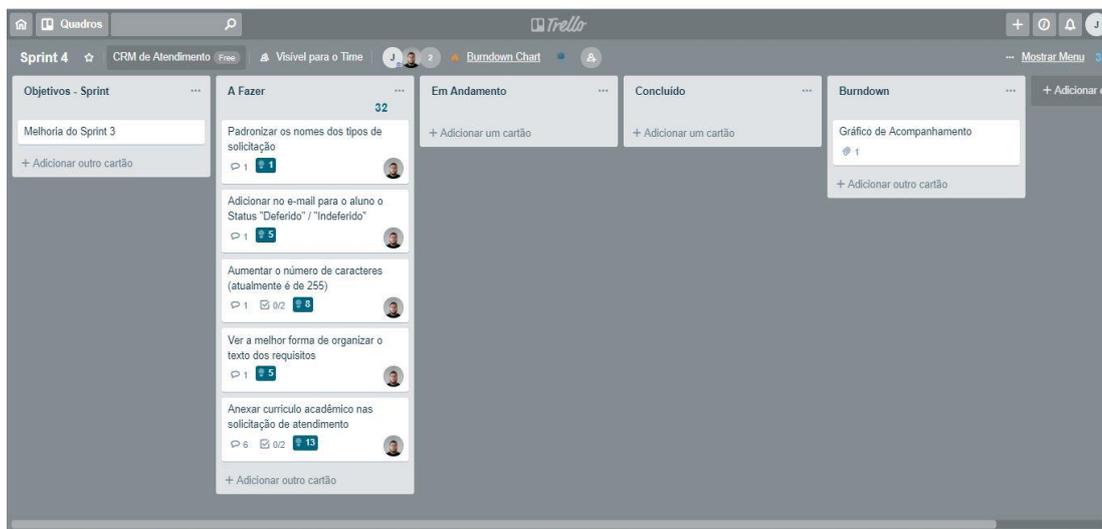
Ao término do *sprint* 3 foi realizada a reunião de revisão do *sprint* com o intuito de fazer as anotações das lições aprendidas. As lições aprendidas estão descritas abaixo:

- Houve um evento da instituição que requereu maior disponibilidade dos integrantes do time de desenvolvimento e estes tiveram que priorizar o evento que aconteceu antes do encerramento do *sprint*, desta forma alguns dias do *sprint* ficaram sem ter tarefas concluídas, causando atraso no andamento do projeto. Isso não impactou no atendimento ao objetivo do *sprint*. Para os próximos *sprint* sugere-se analisar a possibilidade de eventos que requeiram o envolvimento de integrantes do projeto do núcleo de tecnologia da informação.

4.2.4 *Sprint* 4

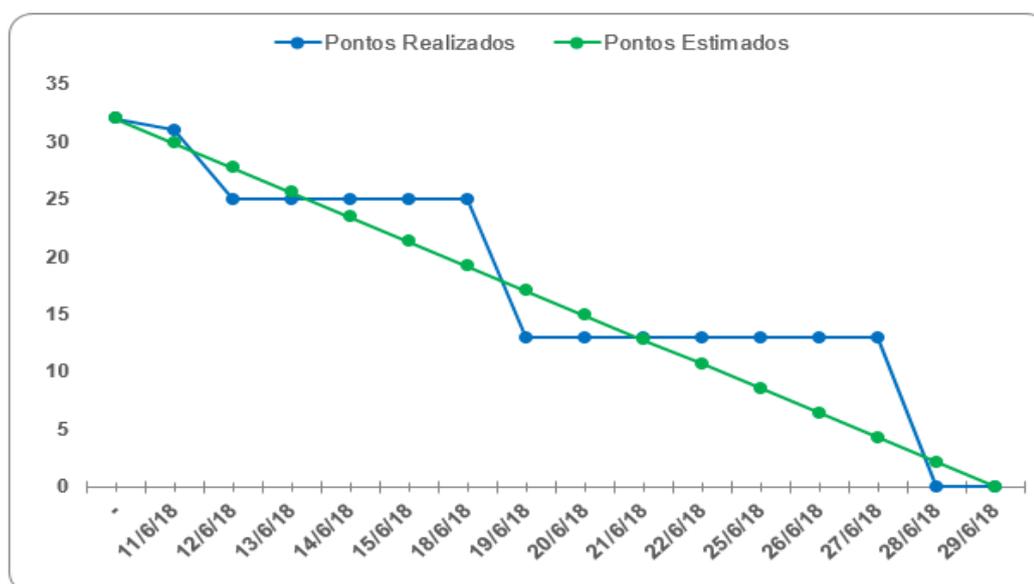
Concluída a reunião de encerramento no *sprint* 3, iniciou-se a reunião de planejamento do *sprint* 4. O objetivo deste foi: “Melhoria do *sprint* 3”. Na execução desse o time de desenvolvimento do projeto esteve reduzido pois um dos integrantes esteve em férias. O responsável pelo desenvolvimento deste teve 2 horas/dia para trabalhar no projeto.

O planejamento da pontuação dessas estórias ocorreu no dia seguinte, onde o integrante do time de desenvolvimento estimou o tempo das estórias e informou à *Scrum Master*. Este *sprint* teve 5 estórias com pontuação total de 32 pontos. A figura 29 representa na ferramenta Trello com as estórias e pontuação.

Figura 29 – Quadro de acompanhamento do *sprint* 4

Fonte: A autora, 2018.

Manteve-se o mesmo horário e local para realizar as reuniões diárias às 13h30min no laboratório 01. No APÊNDICE E identifica-se a estórias, a ID, bem como as tarefas que compõem o *sprint*, o responsável por desenvolvê-las e a pontuação das estórias. Ao final do quadro está a velocidade do *sprint*, ou seja, a pontuação total deste. O andamento do projeto foi acompanhado nas reuniões diárias. A figura 30 representa o gráfico *burndown* do *sprint* 4.

Figura 30 - Gráfico *burndown* do *sprint* 4

Fonte: A autora, 2018.

Conforme pode ser identificado no gráfico, durante o *sprint* 4 houve oscilações no andamento do projeto, sendo que o projeto foi concluído com um dia de antecedência, isso foi possível pois foram planejados 32 pontos e concluiu-se o *sprint* com 20,5 pontos.

No dia 02/07/2018 realizou-se a revisão do *sprint*. O integrante do time de desenvolvimento apresentou ao demais integrantes do projeto as estórias concluídas, simulando todos os processos envolvidos, garantindo que as estórias foram concluídas e estão em funcionamento. Após essa apresentação o PO informou que o objetivo do *sprint* 4 havia sido concluído.

Após a revisão do *sprint* foi realizada a retrospectiva do *sprint*, para identificar os pontos positivos e o que pode ser melhorado. As lições aprendidas do *sprint* 4 foram:

- Nesse *sprint* não foi possível desenvolver as estórias em uma sala separada, pois o único integrante do time de desenvolvimento ficou responsável pelo núcleo de tecnologia da informação. Apesar disso, as atividades ao final do *sprint* não foram impactadas pois a pontuação estimada no início do *sprint* era 32 pontos e concluiu-se as estórias com 20,5 pontos;
- A secretaria solicitou ajuste para incluir o código das disciplinas, pois essa informação facilitaria o processo de matrícula;
- Não foi possível gerar relatório através do processo fórmula visual, sendo necessário gerar um relatório para anexar o currículo acadêmico dos estudantes que realizaram solicitação de matrícula com mais de 12 créditos e quebra de requisito. O objetivo de anexar o currículo acadêmico nas solicitações de atendimento foi atendido. Foi utilizada uma *web service* para gerar relatório e um SQL para anexá-lo a solicitação.

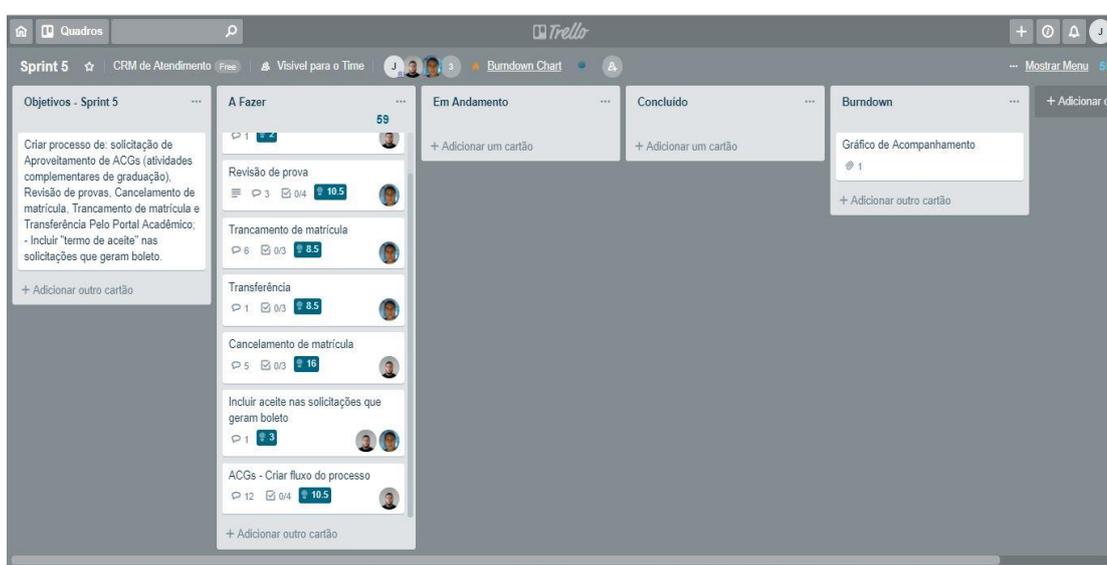
4.2.5 *Sprint* 5

A reunião de planejamento do *sprint* foi realizada no dia 01/08/2018. Nesse evento, o PO informou o objetivo: “Criar processo de: solicitação de Aproveitamento de ACGs (atividades complementares de graduação); Revisão de provas; Cancelamento de matrícula; Trancamento de matrícula e Transferência Pelo Portal Acadêmico; e, Incluir o termo de aceite nas solicitações que geram boleto”.

O time teve 4 horas diárias para desenvolver o *sprint*. O PO encerrou a reunião informando que após concluídas todas as estórias a reunião de revisão do *sprint* e a retrospectiva do *sprint* deveria ser agendada.

O *Scrum Master* e o time de desenvolvimento reuniram-se para estimar a pontuação das estórias e definir as tarefas que compõem cada estória. O *sprint* 5 possui 7 estórias e uma pontuação total de 59 pontos. A figura 31 apresenta a ferramenta Trello com as informações do *sprint*.

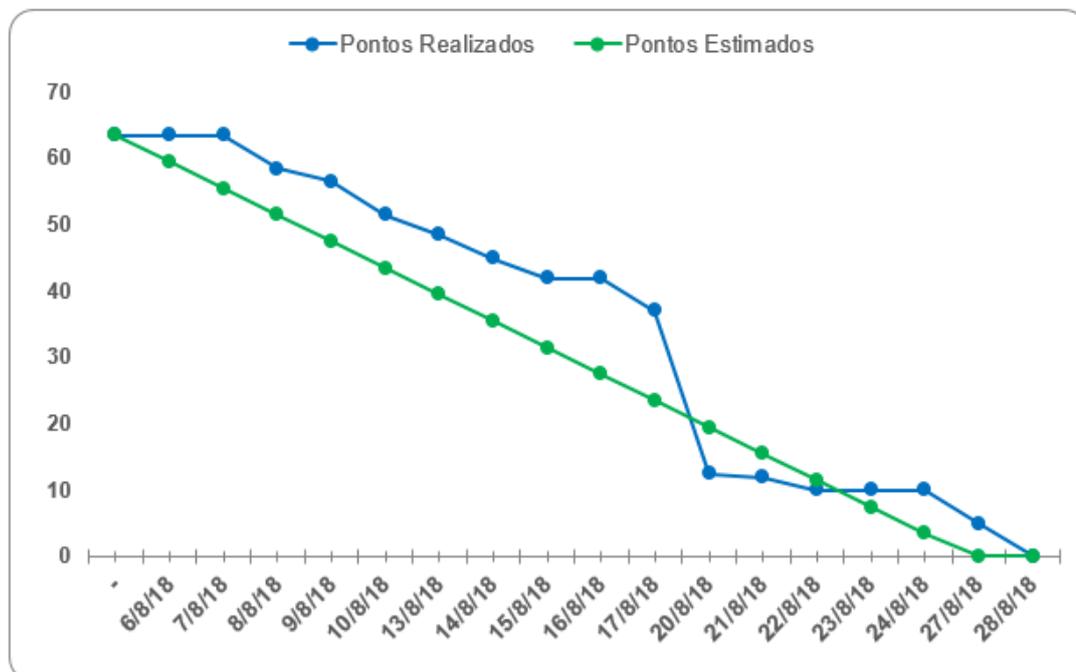
Figura 31 – Quadro de acompanhamento do *sprint* 5



Fonte: A autora, 2018.

No APÊNDICE F está representada a estória, a ID, as tarefas necessárias para concluir cada estória, o responsável por realizar as tarefas e o tempo necessário para conclusão das estórias. Ao final do quadro está a velocidade do *sprint*, ou seja, a pontuação total deste.

Realizou-se o acompanhamento do *sprint* 5 através do gráfico *burndown*, conforme pode ser identificado na figura 32.

Figura 32 - Gráfico *burndown* do *sprint* 5

Fonte: A autora, 2018.

Conforme pode ser identificado no gráfico, o projeto estava atrasado até o dia 17/08/2018. No dia 20/08/2018, foi possível concluir 24,5 pontos e com isso, foi possível identificar através do gráfico que o projeto estava adiantado em relação ao planejado. Já, nos dois últimos dias previstos inicialmente no *sprint* (23/08 e 24/08), um integrante do time de desenvolvimento estava de atestado, com isso não foi possível concluir todas as estórias do *sprint* na data definida. Dessa forma, trabalhou-se mais dois dias para concluí-las.

No dia 29/08/2018, foi realizada a revisão do *sprint*, momento em que o time de desenvolvimento apresentou as estórias concluídas, simulando todos os processos envolvidos, garantindo o funcionamento das mesmas. Após essa apresentação o PO informou que o objetivo deste *sprint* havia sido concluído.

Após a revisão do *sprint*, foi realizada a retrospectiva do *sprint* para identificar os pontos positivos e a ser melhorados. As lições aprendidas foram:

- Com a solicitação de ACGs no portal acadêmico, os estudantes terão acesso para anexar arquivos e com isso não precisam se dirigir até a secretaria para entregar os documentos de ACGs;

- Os manuais necessários para que os acadêmicos, professores, coordenadores e secretaria consigam realizar as etapas do processo demandam tempo para ser realizado pois é necessário detalhar todos os processos. Sugere-se estudar uma forma mais eficiente para criar os manuais, reduzindo o tempo investido.

4.2.6 Avaliação de aplicação da metodologia

Após a conclusão dos *sprints* realizou-se uma avaliação com os integrantes do time de desenvolvimento com o objetivo de analisar a satisfação deste com a aplicação do *Scrum*.

Para a formulação do questionário de avaliação, utilizou-se a escala Likert em que os avaliados responderam à 6 perguntas. Cada pergunta possui escala de 1 a 5, que indicam grau de discordância e concordância, respectivamente. O questionário encontra-se no Apêndice G.

Todos os integrantes do time de desenvolvimento consideraram que o *Scrum* trouxe benefícios para execução do projeto, pela definição exata do que devia ser realizado, e pela oportunidade de desenvolver as atividades em um local isolado. Além disso, foi destacada a importância em dividir o projeto em *sprints* e realizar sua entrega em data definida, e também, a definição de tempo específico para trabalhar no projeto.

Os integrantes do time avaliaram que a utilização das ferramentas do *Scrum* foi importante durante o projeto, pois possibilitou a revisão de informações e acompanhamento de cada etapa do projeto. A comunicação entre os integrantes da equipe através da reunião diária e o acompanhamento do *sprint* através do gráfico de *burndown*, também foi considerada importante.

No que se refere a melhorias nas entregas dos produtos com a utilização do *Scrum* se comparado a projetos anteriores, o time entendeu que, para que o prazo do *sprint* fosse atendido, houve a necessidade do time de desenvolvimento priorizar este projeto, deixando as atividades não relacionadas ao projeto para serem realizadas posteriormente. Além disso, o desenvolvimento do projeto em *sprints* proporcionou maior clareza daquilo que seria entregue e quando a entrega seria feita. A apresentação do *sprint* também proveu uma oportunidade para fazer uma

"entrega oficial" já que, nos outros projetos, a entrega normalmente era feita documentando-se o uso e enviando e-mail com os *links* dos manuais para quem iria utilizar o processo.

Quanto a utilização do *Scrum* em outros projetos do departamento, os pesquisados entendem que se todas as atividades do departamento fossem planejadas, e em um *sprint*, iria melhorar a forma de se organizar e trabalhar. O que impediria o funcionamento 100%, seriam os problemas com alto grau de urgência que surgem diariamente. O outro integrante avaliou que pode ser utilizado em outras situações, desde que elas possam ser desmembradas em outras tarefas com tempo relativamente "previsível". Se não for possível quebrar em subtarefas, ou se essas tarefas forem muito rápidas de serem executadas ou mesmo executadas todas por uma mesma pessoa, acredita-se que o *overhead* (sobrecarga) de fazer todo o processo do *Scrum* para ela não compense (para estes, uma lista simples com um "feito"/"não feito" talvez seja mais eficaz por não requerer alocar tanto tempo para a metodologia em si).

Todos os integrantes entenderam que o *Scrum* trouxe benefícios para a equipe, pois houve maior grau de integração e maior conhecimento absorvido, além disso, foi mencionado que a importância de acompanhar o projeto através do gráfico de *burndown*.

Sobre a importância da utilização do Trello nos *sprints*, um dos integrantes, considerou importante pois possibilitou a visualização do andamento do *sprint* em qualquer momento, porém, foi considerado que não é a melhor ferramenta para ser utilizada com o *Scrum*, mesmo atendendo as necessidades da metodologia. Outro integrante considerou que a ferramenta é uma boa forma de visualizar as tarefas. Mas, para uso constante seria necessário identificar alguma forma de integrá-lo ao sistema de chamados já existentes, para que não seja necessário atualizar as informações em dois locais, pois isso pode causar erros e demanda um tempo maior.

CONCLUSÃO

O objetivo do presente trabalho foi a aplicação de uma metodologia ágil de gestão de projetos, no núcleo de tecnologia da informação de uma instituição de Ensino Superior. A partir da análise do cenário atual, identificou-se a necessidade de gerenciamento nos projetos. Aproveitou-se o início de um projeto de CRM para validar essa pesquisa, fazendo a aplicação da metodologia escolhida.

A revisão bibliográfica permitiu a identificação das principais diferenças entre as metodologias ágeis como, por exemplo a definição de papéis, eventos, o tipo e características de projeto que deseja desenvolver, envolvendo a plataforma (*desktop, web*), a complexidade, prazo específico de entregas, melhoria contínua no próprio projeto, testes em pares.

A partir da revisão bibliográfica, realizou-se a proposta de aplicação do *Scrum* no projeto, definindo, inicialmente, quais ferramentas seriam aplicadas. Para o controle e acompanhamento das estórias definiu-se a utilização da ferramenta Trello, por ser uma ferramenta de gestão visual, gratuita e *on-line*. Essa ferramenta também possibilita o acesso e atualização de dados por parte de todos os integrantes do projeto. Com sua utilização possibilitou-se controlar as estórias, tarefas e pontuações, e também alterar os cartões de “a fazer” para “em andamento” ou “concluído”, de forma prática e visual. Além disso, possibilitou identificar, antes mesmo de analisar o gráfico *burndown*, qual era a pontuação restante para concluir o projeto.

Os papéis, eventos e artefatos a serem utilizados foram definidos. Para o acompanhamento das atividades planejadas e realizadas utilizou-se o gráfico *burndown*, que permitiu o acompanhamento do andamento do projeto, e foi atualizado em planilha eletrônica, durante as reuniões diárias.

Para um melhor dimensionamento das estórias utilizou-se o *planning poker*, permitindo aos integrantes dar sua opinião sobre o entendimento quanto ao desenvolvimento da estória e tempo necessário para desenvolvê-la e, por isso, houve maior visualização do tempo necessário para concluir cada *sprint* e uma melhor confiabilidade na previsão de tempo para a conclusão.

O desenvolvimento do projeto CRM ocorreu no período de 03/04/2018 a 29/08/2018, sendo necessários cinco *sprints*. Com a aplicação do *Scrum* neste projeto, foi possível identificar que os *sprint* atenderam aos objetivos propostos nas reuniões de planejamento. As ferramentas de controle utilizados na aplicação - *sprint backlog*, o gráfico *burndown*, e o Trello - foram suficientes para manter todos os participantes do projeto informados acerca de seu andamento.

Com a aplicação de um questionário, identificou-se a satisfação do time de desenvolvimento quanto a aplicação do *Scrum* no projeto. Os benefícios considerados pela equipe foram: o *Scrum* proporcionou a definição exata das histórias que precisavam ser realizadas para atingir ao objetivo do *sprint*; a divisão do projeto em *sprints*, com data definida para conclusão e o tempo específico para trabalhar no projeto; a comunicação entre os integrantes da equipe através da reunião diária e o acompanhamento do *sprint* através do gráfico de *burndown*; a realização do desenvolvimento das histórias em local isolado, o que não faz parte do *Scrum* em si, mas foi uma definição do PO, e considerada importante pelo time de desenvolvimento, conforme citado nas lições aprendidas do *sprint* 1.

Quanto a utilização da ferramenta Trello, o time de desenvolvimento considerou importante devido a possibilidade de visualizar as tarefas e o andamento do projeto a qualquer hora do dia. Porém, essa não seria a melhor ferramenta para ser utilizada com o *Scrum*. Sugeriu-se que para o uso constante, a ferramenta seria mais eficiente se integrada ao sistema de chamados (GLPI) utilizado pela equipe e, que traz outras demandas não relacionada à projetos.

Portanto, é possível concluir que a utilização de uma metodologia ágil na personalização de *softwares* auxilia a equipe a ter maior eficiência na gestão dos projetos, trazendo benefícios para a equipe e instituição, bem como ao atingimento dos objetivos do projeto.

Por fim, a realização deste estudo foi de grande importância para a autora, devido ao conhecimento obtido durante do desenvolvimento do trabalho e também na relevância do tema que vem ganhando espaço nas empresas e instituições. Além disso, foi gratificante poder realizar este trabalho na própria Instituição de Ensino.

Como sugestão de continuidade da pesquisa e trabalhos futuros, sugere-se novas pesquisas utilizando outras ferramentas para representação do quadro (*Kanban*) e que possam ser integradas a um sistema de gestão de demandas. Outro estudo que pode ser indicado ao departamento é sobre como priorizar demandas em relação à projetos, ou ainda uma análise do número de profissionais necessários para atender as demandas e também novos projetos.

REFERÊNCIAS

- ARRUDA, L. V. **Desenvolvimento Ágil de Software: uma análise sintética a partir da metodologia Kanban**. Anais. VII CONNEPI. Congresso Norte Nordeste de Pesquisa e Inovação. Palmas, Tocantins. 2012. p. 3-6.
- BRANDÃO, V. de O. **Gerência de projetos: Uma comparação entre PMBOK e APM**. Monografia (Bacharel em Ciência da Computação) - Universidade de Lavras, Lavras. 2010.
- BECK, K. **Programação Extrema Explicada – Acolha as Mudanças**. Porto Alegre: Bookman, 2004.
- CAVALCANTI, F. P.; SILVEIRA, J. A. N. **Fundamentos de Gestão de Projetos**. São Paulo: Atlas, 2016.
- CONFORTO, E. C. **Gerenciamento ágil de projetos: proposta e avaliação de método para gestão de escopo e tempo**. 2009. Dissertação (Mestrado em engenharia de produção) Escola de engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.
- CRUZ, F. **Scrum e Agile em Projetos: Guia completo**. Rio de Janeiro: Brasport, 2015.
- CRUZ, F. **PMO Ágil: Escritório ágil de gerenciamento de projetos**. Rio de Janeiro: Brasport, 2016.
- EDER, S. et al. **Estudo exploratório do conceito de agilidade: modelo teórico para aplicação no gerenciamento ágil de projetos**. SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 17., 2010. Bauru. Anais. Bauru: SIMPEP, 2010.
- EDER, S. **Práticas de gerenciamento de projetos de escopo e tempo nas perspectivas das abordagens ágil e tradicional**. 2012. Dissertação (Mestrado em engenharia de Produção) - Escola de Engenharia de São Carlos, Unidade de São Paulo, São Carlos, 2012.
- GUIMARÃES, S; CAMPOS, J. de P. **Em busca da eficácia em treinamentos**. Associação Brasileira de Treinamento e Desenvolvimento: São Paulo, 2009.
- HIRAMA, K. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro: Elsevir, 2012.
- ISTOB. Glossário de termos, versão 3. Disponível em: <http://bstqb.org.br/uploads/glossario/glossario_ctfl_at_3.1br.pdf>. Acesso em: 01 set. 2018.
- JUNGEND, D. et al. **Gestão de Projetos: teoria, prática e tendência**. Rio de Janeiro: Elsevier, 2014.

KINIBERG, H. **Scrum e XP direto das Trincheiras**. Traduzido por InfoQ - *Enterprise Software Development Community*, 2007.

MALHOTRA, N. K. **Pesquisa de Marketing**. 6. ed. Porto Alegre: Bookman, 2012.

MASSARI, V. L. **Gerenciamento ágil de projetos: Uma visão preparatória para a certificação ágil do PMI (PMI-ACP)**. Rio de Janeiro: Brasport, 2014.

MENEZES, L. C. M. **Gestão de Projetos**. 4. ed. São Paulo: Atlas, 2018.

PMBOK. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)**. 6. ed. 2017.

PMI. *Project Management Institute* Disponível em:<<https://www.pmi.org/Pulse>> Acesso em: 31 out. 2018.

PRESSMAN, R. **Engenharia do Software: Uma abordagem profissional**. 7. ed. Porto Alegre: Bookman, 2011.

PRESSMAN, R.; MAXIM, B. **Engenharia de software: Uma abordagem profissional**. 8. ed. São Paulo: Bookman, 2016.

SABBAG, R. **Scrum - Gestão Ágil para Projetos de Sucesso**. São Paulo: Casa do Código, 2014.

SBROCCO, J. H. T. de C; MACEDO, P. de. **Metodologias ágeis: engenharia de software sob medida**. 1. ed. São Paulo: Érica, 2012.

SCHWABER, K. SUTHERLAND, J. **Um Guia definitivo para o Scrum: As regras do Jogo**. 2017.

SOTILLE, M. A. et al. **Gerenciamento do escopo em projetos**. Rio de Janeiro: Editora FGV, 2014.

SOMMERVILLE, I. **Engenharia do Software**. Trad. de I. Bosnic e K. G. de O. Gonçalves. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

TELES, V. M. **Extreme Programming**. São Paulo: Novatec, 2004.

TUDESCO, A. P. K. **Gerenciamento ágil de projetos - fatores de sucesso determinantes da agilidade e facilitadores de resultados inovadores**. Dissertação (Mestre em engenharia de Produção, Sistemas de Produção) - Universidade Federal do Rio Grande do Sul. 2014.

VARGAS, L. M. **Gerenciamento ágil de projetos em desenvolvimento de software: um estudo comparativo sobre a aplicabilidade do SCRUM em conjunto com PMBOK E/OU PRINCE2**. Disponível em: <<http://www.revistagePOrg/ojs/index.php/gep/article/view/398/pdf>>. Revista de Gestão e Projetos - GeP, v. 7, n. 3, p.52-56, Set.Dez. 2016.

VERGARA, S. C. **Projetos e Relatórios de Pesquisa em Administração**. 16. ed. São Paulo: Atlas, 2016.

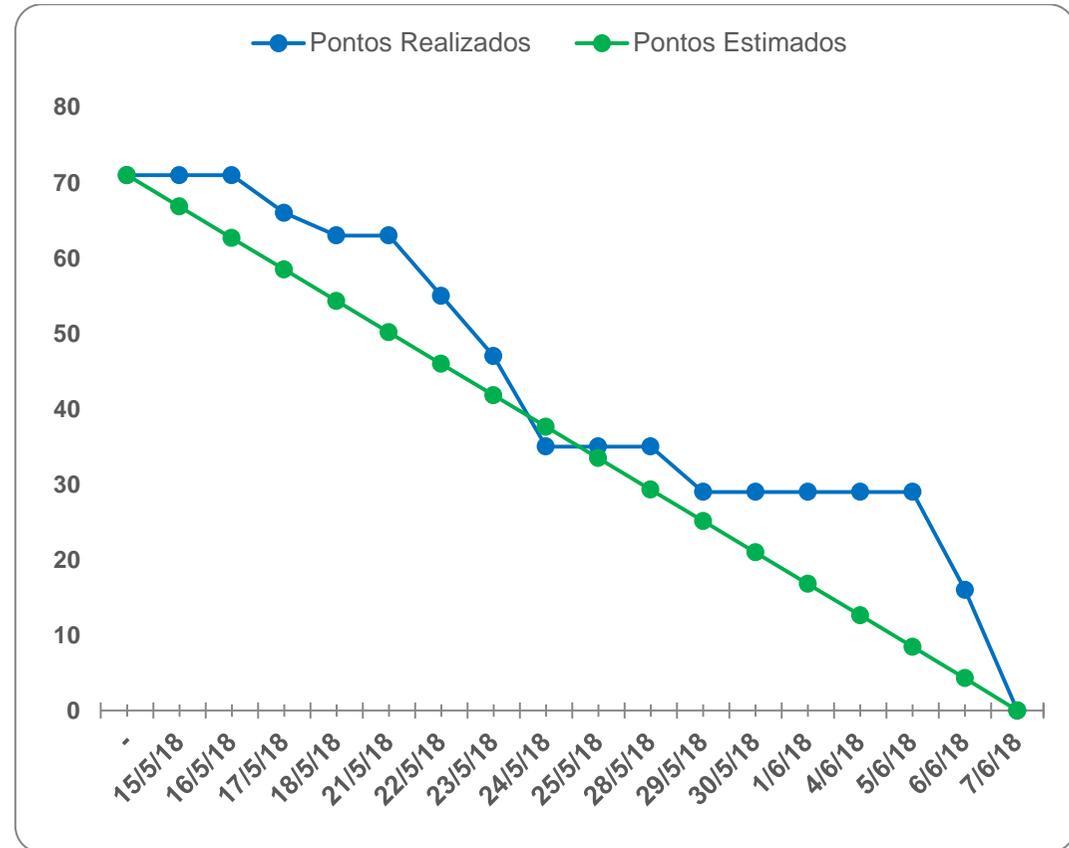
ZIKMUND W. G; BABIN, B. J. **Princípios da Pesquisa de Marketing**. Trad. de EZ2 Translate. 4. ed. norte-americana, 2.ed. brasileira. São Paulo: *Cengage Learning*, 2012.

APÊNDICE A

Pontuação para controle do gráfico *burndown*

Data	Pontos Planejados / Dia	Pontos Realizados / Dia	Pontos Realizados	Pontos Estimados
-	-	-	71	71
15/5/18	4	0	71	67
16/5/18	4	0	71	63
17/5/18	4	5	66	58
18/5/18	4	3	63	54
21/5/18	4	0	63	50
22/5/18	4	8	55	46
23/5/18	4	8	47	42
24/5/18	4	12	35	38
25/5/18	4	0	35	33
28/5/18	4	0	35	29
29/5/18	4	6	29	25
30/5/18	4	0	29	21
1/6/18	4	0	29	17
4/6/18	4	0	29	13
5/6/18	4	0	29	8
6/6/18	4	13	16	4
7/6/18	4	16	0	0

Fonte: A autora, 2018.



APÊNDICE B

Lista de estórias, ID, tarefas e responsáveis e pontos das estórias do *Sprint 1*

ESTÓRIA	ID	TAREFAS	RESPONSÁVEL
Criar campos complementares de Disciplina e Turma	1	Testar campos complementares	Luís
	2	Verificar como os campos complementares aparecem no portal	Luís
	3	Copiar valor preenchido na turma/disciplina para o corpo da solicitação	Luís
	4	Validar que o aluno encontra-se matriculado na turma/disciplina escolhida	Luís
Fórmula visual para avançar etapa no processo automaticamente com pagamento	5	Conhecer processo no sistema	Felipe
	6	Conhecer estrutura no DB	Felipe
	7	Criar SQL utilizado pelo processo	Felipe
	8	Desenvolver o processo	Felipe
	9	Realizar testes	Felipe
	10	Importar no ambiente de produção	Felipe
Criar relatório de chamada para a prova	11	Agendar a execução	Felipe
	12	Criar consulta SQL	Felipe
	13	Criar o relatório	Felipe
	14	Realizar testes/ Ajustes	Felipe
	15	Importar para o ambiente de produção	Felipe
Bloquear visibilidade do arquivo para os alunos	16	Impedir que o aluno acesse o arquivo com a prova quando o professor anexá-lo na solicitação.	Luís
Criar o processo - fluxo	17	Criar o tipo de atendimento, as tarefas, definir os caminhos que o fluxo deve seguir e estar os passos.	Luís
Criar relatório de pagamento para professores	18	Criar consulta SQL	Felipe
	19	Criar relatório	Felipe
	20	Realizar testes	Felipe
	21	Importar para o ambiente de produção	Felipe
Criar Perfil	22	Criar perfil para os professores	Luís/Felipe
	23	Criar perfil para os alunos	Luís/Felipe
	24	Criar perfil para Secretaria	Luís/Felipe
	25	Importar Professores e secretários como Atendentes	Luís/Felipe
	26	Definir filtro padrão de atendimentos para Professores	Luís/Felipe
	27	Acertar Localidades, Grupos de Atendentes e Grupos de Atendimentos	Luís/Felipe
	28	Associar os Atendentes importado a uma Localidade	Luís/Felipe
	29	Associar os Atendentes importados a um Grupo de Atendentes	Luís/Felipe
	30	Replicar no ambiente de produção: as associações de atendentes importados a um grupo de atendentes; os atendentes que foram importados a uma Localidade; Localidades, Grupos de Atendentes e Grupos de Atendimentos; padrão de Atendimentos para Professores.	Luís/Felipe
Criar manuais	31	Criar manual para os Alunos	Felipe
	32	Criar manual para Professores	Felipe
Personalizar e testar e-mails	33	Testes iniciais de funcionamento do envio do e-mail	Felipe
	34	Não enviar e-mail para erp@fahor.com.br (pois ele abre chamado no CI) quando o fluxo passar pelo Servidor de processos.	Felipe
	35	Criar relatório para personalizar o e-mail enviado	Felipe
	36	Verificar porque ele não está enviando e-mail na abertura do chamado	Felipe

Fonte: A autora, 2018.

APÊNDICE C

Lista de estórias, ID, tarefas e responsáveis e pontos das estórias do *Sprint 2*

ESTÓRIA	ID	TAREFAS	RESPONSÁVEL	PONTOS
Incluir item no Manual dos Professores	37	Adicionar no Manual dos professores informação referente ao item "salvar/enviar"	Felipe	2
Liberar processo para os Estudantes	38	Enviar e-mail para os Estudantes	Lúis	2
Ajustar relatório de pagamento de professores	39	Realizar ajustes necessários	Felipe	0,5
Criar Processo de obtenção das provas do Sistema	40	Criar este processo	Felipe	3
Acompanhar a solicitação das provas	41	Solicitação de provas realizadas pelos Estudantes	Felipe/Lúis	5
	42	Disponibilização de provas no Sistema pelos Professores	Felipe/Lúis	5
	43	Acesso e impressão das provas - Secretaria	Felipe/Lúis	3
Liberar processo para os professores	44	Incluir perfil para os Professores	Lúis	1
	45	Enviar e-mail para Professores	Lúis	
Lançar solicitação de provas atrasadas existentes em papel para o Sistema.	46	Revisar o Perfil (atualmente está com erro de permissão)	Lúis	3
	47	Lançar as solicitações já recebidas em papel		
	48	Baixa do lançamento gerado		
Criar Manual	49	Criar manual para Secretaria	Felipe	3
O processo não está enviando o e-mail ao abrir o chamado (parece ser um bug do sistema, mas são necessários mais testes para confirmar).	50	Identificar o problema e resolvê-lo	Lúis	8
Realizar ajustes no processo responsável por enviar prova por e-mail	51	Limitar o assunto do email em 120 caracteres para corrigir o erro	Felipe	0,5
	52	Alterar estrutura do processo para enviar quantos arquivos estiverem anexados à solicitação (prova, formulario, etc)		0,5
Criar trigger para mover o arquivo anexado para a tabela de arquivos privados quando o registro for salvo.	53	Criar e agendar fórmula visual para avisar caso ela seja desativa	Lúis	5
	54	Criar e agendar fórmula visual para avisar caso ela seja desativa		
Realizar ajustes/melhorias no processo responsável por avançar etapa após o pagamento	55	Verificar estrutura da tabela "parcela Solicitação"	Felipe	3
	56	Verificar dados contidos na tabela		
	57	Realizar consulta SQL utilizada pelo processo		
Criar o processo "Trocar data da prova atrasada"	58	Criar o processo	Felipe	2
	59	Realizar os testes e ajustes		
	60	Documentar o processo		
			VELOCIDADE	46,50

Fonte: A autora, 2018.

APÊNDICE D

Lista de estórias, ID, tarefas e responsáveis e pontos das estórias do *Sprint 3*

ESTÓRIA	ID	TAREFAS	RESPONSÁVEL	PONTOS
Criar alertas para boletos	61	Incrementar a mensagem de "boleto vencendo no dia"	Felipe	3
	62	Criar mensagem de boleto vencido	Felipe	
Complementação de manuais	63	Para secretaria parametrizar o período de solicitação	Felipe/ Luis	5
	64	Desenho de fluxo e links para os manuais já existentes		
Definir prazo para os professores incluir as provas	65	Criar alerta caso a solicitação esteja com o professor por 2 dias e a prova ainda não esteja anexada;	Luis	8
	66	Após 3 dias, alertar o professor e o coordenador.		
Criar processo para cancelamento de atendimentos	67	Financeiro	Felipe	8
	68	Criar documentação do processo de cancelamento		
Solicitação de Quebra de Requisito: Criar o fluxo do processo (etapas)	69	Coordenador não aceita: encerra o chamado	Luis	5
	70	Coordenador aceita: encaminha para a secretaria		
	71	Secretaria: encerra o chamado		
Solicitação de Quebra de Requisito: Validações	72	A disciplina escolhida deve possuir pré requisitos	Luis	3
	73	A disciplina não pode estar cursada ou em curso		
	74	O aluno não cumpriu os pré requisitos		
Criar processo que envie e-mail de acompanhamento para o acadêmico, sempre que o atendimento avançar uma etapa.	75	Criar a consulta SQL	Felipe	5
	76	Montar a fórmula visual		
	77	Realizar os testes		
Solicitação de Quebra de Requisito: Preenchimento do complemento	78	Requisitos da disciplina escolhida;	Luis	8
	79	Situação do estudante neste requisito (por exemplo, se ele não cursou, se cursou e reprovou (o que pode ter acontecido múltiplas vezes))		
	80	Quando este requisito foi ofertado pela última vez;		
Solicitação de Quebra de Requisito: Teste e documentação	81	Realizar testes para Estudantes	LuisFelipe	8
	82	Realizar Cordenadores		
	83	Realizar Secretária		
	84	Documentação sobre alterações de período das solicitações		
	85	Documentação do processo como um todo		
	86	Realizar testes		
Solicitação de matrícula em menos créditos: Criar o fluxo do processo (etapas)	87	Identificar as etapas	Felipe	5
Solicitação de matrícula em menos créditos: Preenchimento do complemento	88	Justificativa do estudante	Felipe	2
Solicitação de matrícula em menos créditos: Validações	89	Disciplinas não cursadas	Felipe	3
	90	Disciplinas distintas		
	91	Validar o número de créditos total		
	92	Alguma disciplina selecionada		
Solicitação de matrícula em menos créditos: Teste e documentação	93	Realizar testes para Estudantes	Felipe	8
	94	Realizar testes para Cordenadores		
	95	Realizar testes para Secretária		
	96	Documentação sobre alteração de período das solicitações		
	97	Documentação do processo como um todo		
	98	Realizar testes		
			VELOCIDADE	71,00

Fonte: A autora, 2018.

APÊNDICE E

Lista de estórias, ID, tarefas, responsáveis e pontos das estórias do *Sprint 4*

ESTÓRIA	ID	TAREFAS	RESPONSÁVEL	PONTOS
Padronizar os nomes	99	Padronizar os nomes dos tipos de solicitação	Felipe	1
Adicionar no e-mail para o aluno	100	Incluir Status "Deferido/Indeferido"	Felipe	5
Aumentar o número de caracteres (atualmente é de 255)	101	Testar o que está causando o problema	Felipe	8
	102	Contornar o problema		
Ver a melhor forma de organizar o texto dos requisitos	103	Padronizar o texto	Felipe	5
Anexar currículo acadêmico nas solicitação de atendimento	104	Menos de 12 créditos	Felipe	13
	105	Quebra de requisitos	Felipe	
			VELOCIDADE	32,00

Fonte: A autora, 2018.

APÊNDICE F

Lista de estórias, ID, tarefas, responsáveis e pontos das estórias do *Sprint 5*

ESTÓRIA	ID	TAREFAS	RESPONSÁVEL	PONTOS
SQL não exibe lista das disciplinas ofertadas	106	Exibir lista de disciplinas	Felipe	2
ACGs - Criar fluxo do processo	107	Criar o processo	Felipe	10,5
	108	Criar validações		
	109	Anexar currículo		
	110	Documentação e testes		
Solicitação de Revisão de prova - Criar o fluxo do processo (com geração de boleto)	111	Criar o processo	Luis	10,5
	112	Criar validações		
	113	Anexar currículo		
	114	Documentação e testes		
Solicitação de Cancelamento de matrícula	115	Criar o processo	Felipe	16
	116	Anexar o currículo		
	117	Documentação e testes		
Solicitação de Trancamento de matrícula	118	Criar o processo	Luis	8,5
	119	Anexar o currículo		
	120	Documentação e testes		
Solicitação de Transferência	121	Criar o processo	Luis	8,5
	122	Anexar o currículo		
	123	Documentação e testes		
Solicitações que geram boleto	124	Incluir o "termo de aceite"	Felipe Luis	3
			VELOCIDADE	59,00

Fonte: A autora, 2018.

APÊNDICE G

Avaliação do time de desenvolvimento quanto a aplicação da metodologia

RESPOSTA 1

Aplicação do Scrum

Este questionário visa analisar a satisfação do time de desenvolvimento com a aplicação do Scrum no projeto CRM.

1. Você considera que a utilização do Scrum trouxe benefícios para o resultado do projeto?*

	1	2	3	4	5	
Poucos Benefícios	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muitos Benefícios

Justifique. *

Sim, pela definição exata do que deve ser feito, e em sua grande maioria, pelo isolamento realizado para trabalhar no projeto.

2. A utilização das ferramentas do Scrum foram importantes durante o projeto?

	1	2	3	4	5	
Pouco Importante	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muito Importante

Justifique. *

Foram bem importantes, para a gente revisarmos informações e acompanharmos o caminho de cada etapa do projeto.

3. As entregas dos Produtos foram melhoradas com a utilização do Scrum, se comparado aos projetos anteriores? *

	1	2	3	4	5	
Não Houve Diferença	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Muita Diferença

Justifique. *

Entendo que sim, mas acredito que mais devido o trabalho que estamos desenvolvendo junto a você, para atendermos o prazo, mas assim, acabávamos sacrificando algumas outras tarefas.

4. Você acredita que o Scrum pode ser utilizado em outros projetos do departamento? *

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Justifique. *

Acredito que se todas as atividades do departamento fossem planejadas, e entrassem em um sprint, iria melhorar a forma de se organizar e trabalhar. O que impediria o funcionamento 100%, seriam os problemas com alto grau de urgência que surgem diariamente.

5. O Scrum trouxe benefícios para a equipe? *

	1	2	3	4	5	
Poucos Benefícios	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muitos Benefícios

Justifique. *

Sim, tivemos um maior grau de integração e muito conhecimento absorvido.

6. Você considera a utilização do Trello importante durante os sprints? *

	1	2	3	4	5	
Pouco Importante	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Muito Importante

Justifique. *

É importante, pois entrega de forma visual o andamento do sprint, a qualquer momento do dia. Acredito que não seja a melhor ferramenta para ser utilizada com o Scrum, mas atende a necessidade.

RESPOSTA 2

Aplicação do Scrum

Este questionário visa analisar a satisfação do time de desenvolvimento com a aplicação do Scrum no projeto CRM.

1. Você considera que a utilização do Scrum trouxe benefícios para o resultado do projeto? *

	1	2	3	4	5	
Poucos Benefícios	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muitos Benefícios

Justifique. *

Sim. Ao meu ver, o maior benefício foi o de sermos "forçados" a quebrar o projeto em partes e agrupá-las em sprints, com cada sprint produzindo algo para ser entregue em uma data especificada. Isso, aliado à alocação de um tempo específico para trabalhar no projeto, foram essenciais para o sucesso dele.

2. A utilização das ferramentas do Scrum foram importantes durante o projeto?

	1	2	3	4	5	
Pouco Importante	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muito Importante

Justifique. *

Acredito que o maior benefício é a comunicação entre os integrantes da equipe promovido pela reunião diária e o gráfico que permite acompanhar o progresso.

3. As entregas dos Produtos foram melhoradas com a utilização do Scrum, se comparado aos projetos anteriores? *

	1	2	3	4	5	
Não Houve Diferença	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muita Diferença

Justifique. *

A quebra do projeto em sprints permitiu deixar mais claro o que seria entregue e quando a entrega seria feita. A apresentação do sprint também provê uma oportunidade para fazer uma "entrega oficial" já que, nos outros projetos, a entrega normalmente era feita documentando-se o uso e enviando e-mail com os links dos manuais para quem iria utilizar o processo.

4. Você acredita que o Scrum pode ser utilizado em outros projetos do departamento? *

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Justifique. *

Sim, pode ser utilizado em outras situações, desde que elas possam ser desmembradas em outras tarefas com tempo relativamente "previsível". Se não for possível quebrar em subtarefas (ou se essas tarefas forem muito rápidas de serem executadas ou mesmo executadas todas por uma mesma pessoa), acredito que o overhead de fazer todo o processo do scrum para ela não compense (para estes, uma lista simples com um "feito"/"não feito" talvez seja mais eficaz por não requerer alocar tanto tempo para o framework em si).

5. O Scrum trouxe benefícios para a equipe? *

	1	2	3	4	5	
Poucos Benefícios	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Muitos Benefícios

Justifique. *

Para a equipe, acredito que o maior benefício foi ter o gráfico para acompanhar o andamento. Este mesmo gráfico também permitiu que, em interrupções, pudéssemos justificar o adiamento do atendimento de outra solicitação de fora do projeto para que o projeto não sofresse atrasos (ou mais atrasos), já que muitas das datas de entregas estavam atreladas à algum evento no calendário de eventos da instituição (ex: as solicitação de prova atrasada, obrigatoriamente, precisariam estar disponíveis algumas semanas antes da realização da prova, que já estava definida previamente no calendário de eventos).

6. Você considera a utilização do Trello importante durante os sprints? *

	1	2	3	4	5	
Pouco Importante						Muito Importante
	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Justifique. *

É uma boa forma de visualizar as tarefas. Entretanto, para uso mais constante, seria necessário identificar alguma forma de integrá-lo ao sistema de chamados já existentes, pois fazer o registro duplicado (no sistema de chamados + Trello) é um processo chato e bastante sujeito a erros (lançar uma coisa em um e outra coisa em outro, por engano).

Google Formulários