UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

UM ESTUDO DE CASO SOBRE ADOÇÃO DE PRÁTICAS ÁGEIS EM UM AMBIENTE TRADICIONAL

CARLOS EDUARDO AZEVEDO COSTINHAS DA SILVA

Orientador
PROF. GLEISON DOS SANTOS SOUZA, D.SC.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2013

UM ESTUDO DE CASO SOBRE ADOÇÃO DE PRÁTICAS ÁGEIS EM UM AMBIENTE TRADICIONAL

CARLOS EDUARDO AZEVEDO COSTINHAS DA SILVA

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada por:

_____

Prof. Gleison dos Santos Souza, D.Sc. (UNIRIO)

_____

Prof. Leonardo Guerreiro Azevedo, D.Sc. (IBM Research – Brazil; PPGI-UNIRIO)

_____

Prof. Márcio de Oliveira Barros, D.Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.
DEZEMBRO DE 2013

Ao meu pai, Silvio,
à minha mãe, Renata,
às minhas irmãs Beatriz e Patricia,
e à minha esposa, Juliana,
pelo apoio incondicional.

AGRADECIMENTOS

Agradeço a Deus pela vida e pela oportunidade de seguir e concluir mais essa etapa da minha história. Agradeço também a Meishu Sama, por todos os ensinamentos que me fazem estar mais perto de Deus.

Agradeço aos meus pais, Silvio e Renata, pela educação e princípios que foram a base da formação do meu caráter, pelo apoio e amor incondicional e por todas as renúncias que fizeram pensando em meu futuro. Agradeço às minhas irmãs Beatriz e Patricia e a toda a minha família pelos pensamentos positivos e fé em mim depositada.

Agradeço à minha esposa, Juliana, pelo apoio incondicional, companheirismo, incentivo e, sobretudo pela paciência em todos estes anos de graduação.

Agradeço aos meus amigos por todas as ideias mirabolantes, pelos conselhos e puxões de orelha. Aos meus amigos do trabalho, com os quais compartilhei bons momentos, desafios e aprendizados durante minha graduação e em especial à minha equipe, que me deu espaço e liberdade para testar novos conceitos e desenvolver este projeto.

Agradeço aos todos os professores que fizeram parte da minha vida, por sua dedicação e ensinamentos. Em especial, ao meu orientador Gleison Santos por me guiar e orientar durante este projeto.

Agradeço aos membros da banca, por aceitarem participar da apresentação deste projeto de graduação.

Meus mais sinceros agradecimentos a todos estes.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

No cenário globalizado, onde o maior desafio das organizações é lidar com um grande fluxo de informações, a tecnologia da informação torna-se um importante fator que possibilita o desenvolvimento de *softwares* para gestão da informação e de seu fluxo, a fim de gerar conhecimento e oferecer recursos para a tomada de decisões. No entanto, as características e necessidades do mercado são dinâmicas e muitas vezes exigem que as organizações se adaptem a mudanças frequentes em diferentes elementos que afetam os projetos. Por exemplo, a definição de todos os requisitos do *software* antes que se inicie o projeto pode ser uma tarefa difícil e, muitas vezes, manter essa estratégia pode gerar taxas de sucesso criticamente baixas.

Para lidar com situações como estas, iniciativas foram tomadas para buscar encontrar maneiras mais eficientes para o desenvolvimento de sistemas complexos. Algumas dessas iniciativas deram origem às metodologias ágeis, que partem do princípio que a ocorrência de mudanças é uma constante e não uma exceção. Porém, apesar de frequentes relatos de sucesso no uso de metodologias ágeis para incorporar mudanças, muitas organizações ainda utilizam as metodologias convencionais para desenvolver *software*. Convencer uma organização a mudar o paradigma de desenvolvimento não é uma tarefa trivial e requer a execução de ações que possibilitem indicar as necessidades de mudança, além de planejamento e monitoração de ações coordenadas para a melhoria dos processos.

O objetivo desta monografia é apresentar uma experiência de melhoria de processos de *software*, com foco na adoção de práticas ágeis em uma equipe de desenvolvimento de *software* de uma empresa de grande porte do segmento de telecomunicações, sediada no Rio de Janeiro. Foram executados 4 ciclos de melhoria, pensados de forma a atender expectativas das equipe envolvidas e, ao mesmo tempo, minimizar a possível rejeição às mudanças. Os resultados obtidos com a execução destes ciclos foram a aproximação das áreas técnicas e de negócio; melhorias na comunicação e na gestão de atividades das equipes; otimização da capacidade de trabalho da equipe; redução da rejeição das equipes às mudanças nos processos e apoio das gerências para implantação de melhorias nos processos.

1. INTRODUÇÃO

Este capítulo apresenta a introdução da monografia com a descrição do contexto e dos problemas que motivaram e justificaram o seu desenvolvimento. Os objetivos e a metodologia aplicada são apresentados em seguida. Ao final, é apresentada a forma de organização desta monografia.

1.1. CONTEXTO

Por influência da globalização, um dos principais desafios das organizações atuais é lidar com o grande fluxo de informações, decorrentes da redução dos limites territoriais e do avanço tecnológico. Neste cenário, a tecnologia da informação torna-se um importante fator que possibilita o desenvolvimento de *softwares* para gestão da informação e de seu fluxo, a fim de gerar conhecimento e oferecer recursos para a tomada de decisões. No entanto, as características do mercado mudam rapidamente, as necessidades das organizações se alteram e novas ameaças competitivas surgem sem aviso prévio, impossibilitando, em muitos casos, a definição de todos os requisitos do *software* antes que se inicie o projeto.

Segundo HIBBS *et al.* (2009), enquanto o número de produtos e serviços relacionados a *software* aumenta drasticamente a cada ano, as taxas de sucesso dos projetos de desenvolvimento têm sido criticamente baixas durante décadas. Este fato pode ser observado no estudo intitulado CHAOS Manifesto (STANDISH GROUP, 2010), que demonstrou que apenas 32% dos projetos de desenvolvimento de software analisados são entregues dentro do custo, prazo e escopo previstos. Do restante, 44% dos projetos sofrem com atrasos no cronograma, custos elevados ou problemas de especificação e os outros 24% são cancelados. De acordo com HIBBS *et al.* (2009), esta baixa taxa de sucesso está relacionada com a dificuldade que o modelo Cascata de desenvolvimento possui para absorver mudanças de requisitos.

De acordo com BASSI (2008), o contexto citado anteriormente estimulou diversas iniciativas no ramo da engenharia de *software* que buscavam encontrar maneiras mais eficientes para o desenvolvimento de sistemas complexos. A partir da metade da década de 90, algumas dessas iniciativas deram origem a metodologias com um foco maior nos fatores humanos e na satisfação do cliente, ao invés da burocracia dos processos, sendo inicialmente chamadas de metodologias leves.

Depois de diversos resultados positivos, especialistas nestas metodologias se reuniram em 2001 para discutir suas técnicas e publicaram um manifesto que reuniu

seus valores essenciais (AGILE MANIFESTO, 2013). A partir de então, as metodologias leves passaram a ser chamadas de ágeis.

1.2. MOTIVAÇÃO

Ao abordar as mudanças de requisitos nos projetos de desenvolvimento de software que utilizam processos convencionais, PRESSMAN (2011) afirma que os custos aumentam de forma não linear conforme o andamento do projeto, como pode ser observado na Figura 1, a seguir. Isto faz com que a ocorrência de mudanças de requisitos em fases avançadas do projeto causem grande impacto nos custos e prazos definidos.

Nas metodologias ágeis, as mudanças nos requisitos são bem-vindas mesmo em fases avançadas do desenvolvimento, sendo exploradas para trazerem vantagem competitiva para o cliente (AGILE MANIFESTO, 2013).

PRESSMAN (2011) também cita que, de acordo com os defensores da agilidade, um processo ágil bem elaborado torna a curva do custo de mudança mais plana do que nos processos convencionais, como também pode ser observado na Figura 1, permitindo que as mudanças sejam assimiladas sem um impacto significativo em custo ou prazo.

Figura 1: Os custos da mudança de requisitos ao longo do projeto
Fonte: PRESSMAN (2011).

Os projetos de desenvolvimento de *software* que utilizam processos ágeis não tratam apenas de mudanças relacionadas aos requisitos. A predisposição é tratar mudanças de uma forma geral, incluindo todos os aspectos envolvidos em um projeto. De acordo com DYBA *et al.* (2009), alguns estudos apontam que estes projetos podem incorporar as mudanças com mais facilidade e demonstrar valor ao negócio de forma mais eficiente do que os projetos que utilizam processos convencionais. Além disso, as técnicas de

gerenciamento ágil de projetos podem ser combinadas com as técnicas convencionais, permitindo uma maior flexibilidade do processo de desenvolvimento.

A aplicação de práticas ágeis não é um processo trivial, pois trata de mudanças na cultura organizacional e no ambiente de trabalho. Sua adoção, portanto, deve contar com o apoio da organização e com um planejamento que torne possível a implantação de mudanças graduais, reduzindo o impacto da equipe às mudanças propostas.

Pelos motivos supracitados, acredita-se que a adoção das práticas ágeis, combinadas com as práticas dos processos convencionais já utilizadas, possam reduzir o impacto das mudanças no custo e prazo dos projetos de desenvolvimento de *software*.

1.3. OBJETIVOS

O objetivo desta monografia é apresentar uma experiência de melhoria de processos de *software*, com foco na adoção de práticas ágeis em uma equipe de desenvolvimento de *software* de uma empresa de grande porte do segmento de telecomunicações, sediada no Rio de Janeiro.

Alguns objetivos específicos dessa monografia são:

- Disseminar na equipe os conceitos e práticas propostos pelas metodologias e *frameworks* ágeis;
- Reduzir a rejeição da equipe às mudanças de procedimentos a serem adotadas;
- Evoluir o controle de projetos e atividades da equipe, visando organizar e melhorar a visibilidade das tarefas em andamento;
- Melhorar a comunicação da equipe visando reduzir o tempo de resolução de problemas, através da troca de conhecimentos dos membros mais experientes com os membros menos experientes.

1.4. A ORGANIZAÇÃO EM ESTUDO

A iniciativa de melhoria de processos de *software* foi realizada em uma empresa privada de grande porte do segmento de telecomunicações, que atua no Brasil há 18 anos e possui como foco a telefonia móvel e a banda larga. A empresa possui mais de 10.000 colaboradores em todo o país, alocados principalmente nos estados do Rio de Janeiro e São Paulo.

A área selecionada para esta melhoria foi a Gerência de Tecnologia da Informação responsável pelo desenvolvimento e manutenção de sistemas relacionados ao registro e cobrança de serviços de telefonia fixa. Esta equipe contém 4 integrantes e possui interação com uma Fábrica de *Sofware*, que possui 3 integrantes, e com uma Área de Negócios, que possui 7 integrantes.

Os problemas identificados nesta área foram a dificuldade com o planejamento da equipe, causados pela baixa eficiência dos controles utilizados na gestão de atividades; a dificuldade para a revisão dos requisitos do produto durante seu processo de desenvolvimento, que é baseado no modelo cascata; a insatisfação da Área de Negócios com os projetos de desenvolvimento de *software* realizados na empresa e a baixa taxa de resposta do processo de desenvolvimento às necessidades de mudanças no mercado.

1.5. METODOLOGIA

Para atingir os objetivos propostos, a iniciativa de melhoria realizada na empresa descrita na Seção anterior foi dividida em 7 etapas:

1) A primeira etapa consistiu em realizar uma revisão da literatura sobre as práticas, metodologias e *frameworks* ágeis, com foco no Scrum, no Kanban e no desenvolvimento enxuto de software (*Lean Software Development*), visando identificar quais conceitos e práticas poderiam ser aplicadas na equipe para atingir os objetivos desta monografia.
2) Na segunda etapa foi realizada uma pesquisa com as equipes envolvidas nos projetos de desenvolvimento de software na organização selecionada, afim de identificar o perfil das pessoas, os problemas existentes na organização e pontuar quais são os critérios considerados críticos para o sucesso dos projetos. Para auxiliar na priorização dos critérios de sucesso, foi utilizado o modelo AHP (*Analytic Hierarchy Process*) (VARGAS, 2010). A partir dos resultados desta pesquisa, foram definidos os ciclos de melhoria executados nas próximas etapas.
3) Na terceira etapa, o primeiro ciclo de melhoria foi planejado e executado, buscando identificar quais práticas ágeis poderiam ser aplicadas no ambiente para atingir os objetivos desta monografia. Neste ciclo, as práticas ágeis identificadas na revisão da literatura foram analisadas levando em consideração o perfil da organização e das equipes, com base no resultado da pesquisa realizada na segunda etapa. As práticas selecionadas foram: entregas curtas;

divisão em funcionalidades; *backlog* do produto; metáforas do sistema; cliente presente; ritmo sustentável; times multidisciplinares; visibilidade do projeto; retrospectivas e reuniões em pé.

4) Na quarta etapa, o segundo ciclo de implantação foi planejado e executado, adotando a prática de visualização das atividades proposta pelo Kanban.
5) Na quinta etapa, o terceiro ciclo de implantação foi planejado e executado, buscando atingir os demais objetivos específicos desta monografia. As práticas adotadas foram a limitação do trabalho em andamento (*WIP*), proposta pelo Kanban, e a melhoria na comunicação da equipe, com base em práticas do Scrum, em princípios do Manifesto Ágil e do desenvolvimento enxuto de software (*Lean Software Development*).
6) A sexta etapa consistiu em planejar e executar o quarto ciclo de implantação, que busca atender o objetivo principal desta monografia. Foram selecionados dois projetos piloto e aplicadas práticas relacionadas com os requisitos e desenvolvimento do produto, além de práticas relacionadas com a organização e ambiente de trabalho.
7) A sétima e última etapa consistiu em realizar uma retrospectiva com os membros participantes e verificar se a adoção das práticas atenderam ao objetivo principal desta monografia.

1.6. ESTRUTURA DO TEXTO

Esta monografia está dividida em 6 capítulos, incluindo esta introdução, onde são abordados temas relacionados a processos de desenvolvimento de software e utilização de práticas ágeis.

O segundo capítulo tem como objetivo apresentar a revisão bibliográfica sobre as práticas, metodologias e *frameworks* ágeis, abordando o Scrum, o desenvolvimento enxuto de software, o Kanban e as práticas ágeis mais comuns em projetos de desenvolvimento de software.

O terceiro capítulo capítulo apresenta a descrição da iniciativa de melhoria e do cenário inicial da empresa onde o estudo de caso foi aplicado, apresentando a estrutura organizacional, a percepção das equipes sobre o processo de desenvolvimento de *software*, os indicadores da necessidade de mudanças.

O quarto capítulo apresenta a execução dos ciclos de melhoria, abordando a aplicação dos conceitos revisados nos capítulos anteriores na organização selecionada. Neste capítulo são apresentados os 4 ciclos de melhoria propostos.

O quinto capítulo apresenta a avaliação da proposta aplicada no capítulo anterior, verificando quais foram os pontos positivos e negativos do projeto. Por fim, o sexto capítulo apresenta as conclusões deste trabalho, as contribuições para a organização e os próximos passos desta iniciativa.

2. REVISÃO DA LITERATURA

Este capítulo apresenta os conceitos relacionados às metodologias e *frameworks* ágeis, especificamente às práticas do Scrum, do Kanban e do desenvolvimento enxuto de software (*Lean Software Development*), bem como o manifesto que reuniu os valores destas. Além disso, também foram revisadas as práticas ágeis citadas por quatro artigos, para futura seleção e aplicação nos ciclos de melhoria de processos.

Os textos selecionados neste capítulo são referências sobre cada assunto e, portanto, foram utilizados como base para a revisão da literatura.

2.1. MANIFESTO ÁGIL

As baixas taxas de sucesso nos projetos de desenvolvimento de software motivaram diversas iniciativas no ramo da engenharia de software para encontrar maneiras mais eficientes de desenvolver sistemas complexos. Algumas dessas iniciativas deram origem a metodologias chamadas, inicialmente, de metodologias leves, que possuíam um foco maior nos fatores humanos e na satisfação do cliente ao invés da burocracia dos processos (BASSI, 2008). Depois de diversos resultados positivos, 17 especialistas nestas metodologias perceberam que seus modos de trabalho, além de eficazes, eram parecidos uns com os outros (BASSI, 2008). Em 2001, eles se reuniram durante um final de semana em Utah, nos Estados Unidos, para discutir suas técnicas e estabelecer uma nova metodologia de desenvolvimento de software que pudesse ser utilizada por todos os envolvidos, substituindo os métodos tradicionais (AGILE MANIFESTO, 2013).

Após dois dias de debate, o grupo não conseguiu estabelecer uma metodologia comum e concluiu que desenvolver software é algo complexo demais para ser definido por um único processo. No entanto, o grupo chegou a um consenso de que alguns princípios eram determinantes para se obter bons resultados. O resultado deste encontro foi a criação da Aliança Ágil e o estabelecimento do Manifesto Ágil (AGILE MANIFESTO, 2013), que está baseado em quatro valores fundamentais:

- Indivíduos e interações são mais importantes do que processos e ferramentas;
- Software em funcionamento é mais importante do que documentação abrangente;
- Colaboração com o cliente é mais importante do que negociação de contratos;

- Adaptação a mudanças é mais importante do que seguir o plano inicial.

Estes valores são a base para os doze princípios do desenvolvimento ágil, proposto pelo Manifesto Ágil (AGILE MANIFESTO, 2013):

- A maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de *software* com valor agregado;
- Mudanças nos requisitos são consideradas bem-vindas, mesmo ocorrendo em fases avançadas do desenvolvimento. Os processos ágeis devem aproveitá-las para oferecer vantagens competitivas para o cliente;
- Entregar frequentemente *software* funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
- As equipes da área de negócio e de desenvolvimento devem trabalhar em conjunto durante todo o projeto;
- A construção dos projetos deve ocorrer em torno de indivíduos motivados. É preciso oferecer um ambiente e o suporte necessários e confiar neles para fazer o trabalho;
- O método mais eficiente e eficaz de trocar informações com uma equipe de desenvolvimento é através de conversa face a face;
- *Software* funcionando é a medida primária de progresso;
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- A contínua atenção à excelência técnica e bom *design* aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser realizado.
- As melhores arquiteturas, requisitos e *designs* emergem de times auto-organizáveis.
- Em intervalos regulares, o time deve refletir sobre como ficar mais efetivo e então, se ajustam e otimizam seu comportamento de acordo com esta reflexão.

Tendo em vista a abordagem proposta nesta monografia estar voltada para as práticas do Scrum, do Kanban e do desenvolvimento enxuto de software, elas serão detalhadas a seguir.

2.2. SCRUM

Ao realizar uma revisão bibliográfica sobre o Scrum, é possível notar uma divergência em sua classificação. Algumas fontes o classificam como "uma metodologia ágil para gestão e planejamento de projetos de *software*", como é o caso de DESENVOLVIMENTO ÁGIL (2013). Ken Schwaber e Jeff Sutherland, idealizadores do Scrum, o definem como "um *framework* para desenvolvimento e manutenção de produtos complexos" (SCHWABER *et al.*, 2011).

Nesta monografia, o Scrum será tratado como um *framework* de processo, pois apesar de conter diretrizes obrigatórias, cada implementação pode utilizar técnicas e ferramentas distintas de acordo com a realidade do projeto, podendo ser considerada como uma instância do modelo Scrum. Todo seu detalhamento foi baseado no guia criado por Ken Schwaber e Jeff Sutherland (SCHWABER *et al.*, 2011), para divulgação do Scrum.

2.2.1. HISTÓRIA

Em 1986, Hirotaka Takeuchi e Ikujiro Nonaka publicaram o artigo "*The new new product development game*", onde descreveram uma abordagem composta de seis características que, juntas, davam origem a um processo mais ágil e flexível para desenvolvimento de novos produtos. Este processo foi comparado ao Rugby, onde o time trabalha em conjunto para percorrer o campo como uma unidade (TAKEUCHI e NONAKA, 1986, *apud* LANDIM, 2012).

Em 1991, DeGrace e Stahl foram os primeiros a se referir a isso como uma "Abordagem SCRUM" (DEGRACE e STAHL, 1991, *apud* LANDIM, 2012). No início de 1990, Ken Schwaber juntamente com Jeff Sutherland, John Scumniotales e Jeff McKenna, desenvolveram uma abordagem similar aplicando em sua empresa, a Advanced Development Methods e foram os primeiros a utilizarem a palavra única "Scrum" para se referir ao processo (LANDIM, 2012).

O Scrum, já definido como um processo, foi utilizado pela primeira vez em 1993, quando Jeff Sutherland o aplicou em uma equipe de desenvolvimento de software, na Easel Corporation (SUTHERLAND, 2004). Sua primeira apresentação ao público foi feita por Ken Schwaber em 1995, no *workshop* "*Business Object Design and Implementation*" realizado no evento OOPSLA'95, após publicar o artigo "*SCRUM Development Process*" (SCHWABER, 1995).

Nos anos seguintes o Scrum foi aprimorado e Schwaber uniu-se a Mike Beedle para publicar o livro "*Agile Software Development with Scrum*" (SCHWABER, 2002).

2.2.2. VISÃO GERAL

O Scrum é um processo empírico, que assume que o conhecimento vem da experiência e a tomada de decisões é feita com base no que é conhecido. O Scrum adota uma abordagem iterativa e incremental para otimizar a previsibilidade e melhorar o controle de riscos, se baseando nos seus três pilares: a transparência, a inspeção e a adaptação (SCHWABER *et al.*, 2011).

No Scrum, o desenvolvimento do produto é feito de forma iterativa, isto é, o trabalho é dividido em ciclos denominados *sprints*. Um *sprint* representa um período de tempo onde um conjunto de atividades deve ser executado (SCHWABER *et al.*, 2011).

As equipes no Scrum são enxutas e possuem três papéis principais desempenhados no projeto: o *Scrum master*, o *product owner* e o time de desenvolvimento (SCHWABER *et al.*, 2011).

As funcionalidades que serão implementadas no projeto são mantidas em uma lista, chamada de *backlog* do produto (*product backlog*). No início de cada *sprint*, é realizada uma reunião de planejamento (*sprint planning meeting*), onde o *product owner* prioriza os itens do *backlog* do produto e o time de desenvolvimento seleciona os itens que conseguirão desenvolver durante este *sprint*. Estes itens selecionados são transferidos do *backlog* do produto para outra lista, chamada de *backlog* do *sprint* (SCHWABER *et al.*, 2011).

Após o início de um *sprint*, o time de desenvolvimento realiza uma reunião breve no início de cada dia, denominada reunião diária (*daily Scrum*). O objetivo desta reunião é discutir o que foi feito no dia anterior, identificar os impedimentos atuais e priorizar o trabalho do dia que se inicia (SCHWABER *et al.*, 2011).

Ao final de um sprint, o time de desenvolvimento apresenta as funcionalidades implementadas em uma reunião de revisão do *sprint* (*sprint review meeting*). Por último, é realizada uma reunião de retrospectiva do *sprint* (*sprint retrospective*), com o objetivo de identificar o que funcionou bem e o que pode ser melhorado nos próximos *sprints* (SCHWABER *et al.*, 2011). E assim, o ciclo é reiniciado, conforme detalha a Figura 2:

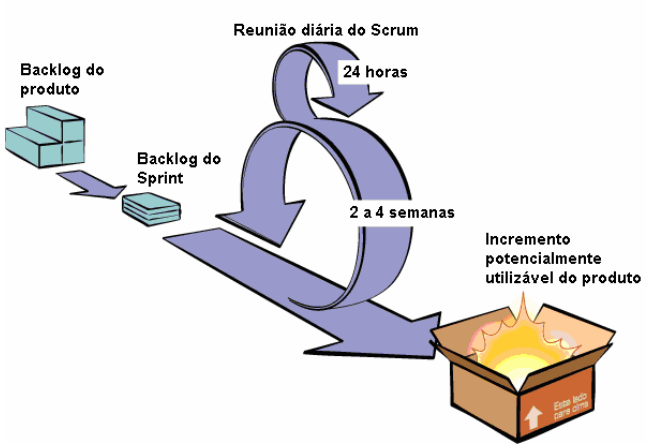

Figura 2: O ciclo de vida do Scrum
Traduzido de: (DESENVOLVIMENTO ÁGIL, 2013).

2.2.3. PILARES DO SCRUM

O Scrum é baseado em três pilares: a transparência, a inspeção e a adaptação (SCHWABER *et al.*, 2011).

A *transparência* garante que os aspectos significativos do processo devem ser visíveis para os responsáveis pelos resultados. Esses aspectos devem ser definidos por um padrão, para que haja um entendimento comum entre os interessados. Por exemplo, uma definição de "pronto"[1] deve ser compartilhada entre os responsáveis pela criação e os responsáveis pela validação do produto. A *inspeção* sugere que os artefatos gerados e o progresso do projeto devem ser inspecionados com uma frequência suficiente para que variações inaceitáveis no processo possam ser detectadas. Porém, esta frequência não deve ser tão alta ao ponto de impactar o fluxo de trabalho. A *adaptação* afirma que, se o inspetor determinar, a partir da inspeção, que um ou mais aspectos do processo estão fora dos limites aceitáveis e que o produto resultante será inaceitável, ele deverá ajustar o processo ou o material sendo processado. Os ajustes devem ser realizados o mais rápido possível para minimizar desvios posteriores (SCHWABER *et al.*, 2011).

Existem quatro pontos para inspeção e adaptação no Scrum: a reunião de planejamento do *sprint* (*sprint planning meeting*), a reunião diária (*daily Scrum*), a

[1] Quando um item do *backlog* do produto ou um incremento for descrito como "pronto", todos os membros da equipe devem entender o que "pronto" significa (SCHWABER *et al.*, 2011). Portanto, todos os critérios que definam um item como "pronto" devem ser compartilhados entre os envolvidos no projeto. A definição de produto "pronto" para uma fábrica de software, por exemplo, pode contemplar um produto entregue com todos os testes realizados. Já na visão de uma Área de Negócios, pode contemplar um produto testado, implantado no ambiente de Produção e utilizável.

reunião de revisão do *sprint* (*sprint review meeting*) e a reunião de retrospectiva do *sprint* (*sprint retrospective*) (SCHWABER *et al.*, 2011).

2.2.4. PAPÉIS DO SCRUM

O modelo de equipe no Scrum é projetado para otimizar a flexibilidade, criatividade e produtividade, contendo três papéis principais: o *Scrum master*, o *product owner* e o time de desenvolvimento. Elas devem ser auto-organizadas, pois são responsáveis por escolher a melhor forma de realizar um trabalho, ao invés de serem dirigidos por uma pessoa que não faz parte da equipe. Também devem ser multifuncionais, pois devem possuir todas as competências necessárias para realizar o trabalho sem depender de pessoas que não fazem parte da equipe (SCHWABER *et al.*, 2011).

O *Scrum master* é responsável por garantir que a equipe siga as teorias, práticas e regras propostas pelo Scrum. Este papel é desempenhado apenas por uma pessoa por equipe, cuja principal função é remover qualquer impedimento que possa tirar o foco do time de desenvolvimento no produto, facilitando a comunicação destes com o *product owner*, buscando técnicas e ferramentas para otimizar o trabalho da equipe, facilitando as reuniões propostas pelo Scrum e auxiliando o time de desenvolvimento a se manter auto-organizado e multifuncional (SCHWABER *et al.*, 2011).

O *product owner* é responsável por gerenciar o *backlog* do produto, detalhando os itens nesta lista de forma clara e objetiva, além de garantir que a priorização dos itens atenda à missão e aos objetivos do projeto da melhor forma possível. É importante ressaltar que o papel de *product owner* é desempenhado por apenas uma pessoa, e não por uma equipe. O *product owner* pode representar uma equipe, mas apenas ele será o responsável por tomar decisões coerentes ao produto neste projeto (SCHWABER *et al.*, 2011).

O time de desenvolvimento é formado por profissionais que realizam o trabalho de entregar, ao final de cada *sprint*, uma versão incrementada e utilizável do produto, de acordo com a definição de "pronto" combinada com a equipe. Somente os membros do time podem desenvolver o produto (SCHWABER *et al.*, 2011).

Os times de desenvolvimento são estruturados e capacitados para organizar e gerenciar seu próprio trabalho. A sinergia resultante neste processo otimiza a eficiência e eficácia de toda a equipe Scrum (SCHWABER *et al.*, 2011).

O time de desenvolvimento deve ser pequeno o suficiente para permanecer ágil e grande o suficiente para completar um trabalho significativo. O indicado é que a equipe

possua de cinco a nove membros. Uma equipe com menos de três membros possui pouca interação e pode render baixa produtividade nas entregas. Já uma equipe com mais de nove membros requer muita coordenação, gerando complexidades para gerenciar o processo empírico. O *product owner* e o *Scrum master* não estão incluídos nesta contagem, a não ser que estejam participando do desenvolvimento do produto (SCHWABER *et al.*, 2011).

2.2.5. EVENTOS DO SCRUM

Os eventos prescritos pelo Scrum têm como objetivo criar uma regularidade no processo e minimizar as necessidades de reuniões extras, não definidas pelo *framework*. Isto garante que o tempo utilizado em planejamento seja adequado, sem excessos. Com exceção do *sprint*, os demais eventos são oportunidades formais para realizar inspeções e adaptações durante o processo, favorecendo a transparência (SCHWABER *et al.*, 2011).

Os eventos do Scrum são: o *sprint*; a reunião de planejamento do *sprint*; a reunião diária; a reunião de revisão do *sprint* e a reunião de retrospectiva do *sprint*.

2.2.5.1. O SPRINT

O principal evento do Scrum é o *sprint*, um período de tempo onde são criados incrementos "prontos", entregáveis e utilizáveis do produto. É recomendado que o *sprint* dure cerca de um mês ou menos, pois se o período for muito longo, seus itens podem mudar ao longo do tempo, gerando aumento da complexidade e dos riscos (SCHWABER *et al.*, 2011).

Durante um *sprint*, os itens do *backlog* do produto que foram selecionados para desenvolvimento podem ser detalhados e revisados entre o time e o *produt owner*, ou até alterados, desde que não afetem os objetivos do *sprint*. Apesar de o escopo poder ser alterado ao longo do desenvolvimento, não é recomendado alterar a composição do time, a duração ou os objetivos gerais do *sprint* (SCHWABER *et al.*, 2011).

Após ser iniciado, um *sprint* pode ser cancelado apenas pelo *product owner*, mesmo que por recomendação das partes interessadas (*stakeholders)*, do time de desenvolvimento ou do *Scrum master*. O cancelamento pode ocorrer se os objetivos do *sprint* se tornarem obsoletos, porém, em função de sua curta duração, é pouco provável que isto aconteça. Após o cancelamento, os itens do *backlog* do *sprint* que já haviam

sido desenvolvidos podem ser entregues caso o *product owner* os aprove (SCHWABER *et al.*, 2011).

2.2.5.2. A REUNIÃO DE PLANEJAMENTO DO SPRINT (SPRINT PLANNING MEETING)

Nesta reunião são planejados os itens que serão desenvolvidos durante o *sprint*, de forma colaborativa por toda a equipe do projeto. É recomendado que a reunião tenha duas fases e uma duração média de oito horas para um *sprint* de um mês, podendo ser adequada caso o *sprint* seja mais curto (SCHWABER *et al.*, 2011).

Na primeira fase, o *product owner* apresenta para o time de desenvolvimento os itens do *backlog* do produto ordenados por prioridade e toda a equipe Scrum colabora na compreensão do trabalho do *sprint*. Esta parte da reunião utiliza as seguintes informações como insumo: o *backlog* do produto; o último incremento do produto, entregue no *sprint* anterior; a capacidade estimada de produção do time de desenvolvimento; e por fim, a performance do time de desenvolvimento no último *sprint* (SCHWABER *et al.*, 2011).

Com o *backlog* do produto priorizado e a capacidade estimada de desenvolvimento, o time estima o esforço que será gasto em cada item do *backlog* e, em seguida, verifica quantos itens poderão ser desenvolvidos dentro deste *sprint*. Por fim, o time desenvolvimento define o objetivo do *sprint*, que irá orientá-los durante o desenvolvimento informando o motivo do incremento estar sendo criado (SCHWABER *et al.*, 2011).

Na segunda fase, o time de desenvolvimento decide como os itens do *backlog* do produto serão desenvolvidos neste *sprint* e detalha as estimativas de cada item selecionado, gerando o *backlog* do *sprint* (SCHWABER *et al.*, 2011).

O *product owner* também deve estar presente nesta fase da reunião, para ajudar na compreensão e priorização dos itens. Se o time de desenvolvimento verificar que foram selecionados muito ou pouco itens para este *sprint*, eles serão analisados novamente com o *product owner*. Nesta fase, pessoas de outras equipes também podem ser convidadas para ajudar no detalhamento técnico ou de funcional (SCHWABER *et al.*, 2011).

Ao final da reunião de planejamento do *sprint*, o time de desenvolvimento deverá ser capaz de explicar ao *product owner* e ao *Scrum master* como eles pretendem

trabalhar para atingir os objetivos do *sprint* e criar o incremento previsto (SCHWABER *et al.*, 2011).

2.2.5.3. A REUNIÃO DIÁRIA (DAILY SCRUM)

A reunião diária é um evento com cerca de 15 minutos de duração, para que o time de desenvolvimento possa rever o progresso do dia anterior e planejar as atividades que serão realizadas nas próximas 24 horas. É recomendado que esta reunião seja realizada sempre no mesmo local e horário, de preferência no início do dia (SCHWABER *et al.*, 2011).

Durante a reunião, cada membro do time de desenvolvimento deve responder às seguintes perguntas (SCHWABER *et al.*, 2011):

- O que foi realizado desde a última reunião?
- O que será realizado até a próxima reunião?
- Quais são os impedimentos que geram impacto em minhas atividades?

O time de desenvolvimento deve utilizar esta reunião para avaliar o progresso em direção ao objetivo do *sprint*, identificar e remover qualquer impedimento que possa impactar a performance do time e verificar como o progresso está evoluindo para a conclusão das atividades do *backlog* ao final do *sprint* (SCHWABER *et al.*, 2011).

O *product owner* não deve estar presente nesta reunião. Já o *Scrum master*, deve estar presente mas não deve participar ativamente desta reunião, pois ela deve ser conduzida pelo próprio time. Ele deve apenas garantir que o time se mantenha focado no objetivo, na frequência e na duração da reunião (SCHWABER *et al.*, 2011).

2.2.5.4. A REUNIÃO DE REVISÃO DO SPRINT (SPRINT REVIEW)

Esta reunião ocorre ao final de cada *sprint* com a participação de toda a equipe do projeto e das partes interessadas (*stakeholders*), visando inspecionar o incremento do produto gerado e adaptar o *backlog* do produto, se for necessário. É recomendado uma duração de quatro horas para um *sprint* de um mês (SCHWABER *et al.*, 2011).

Durante a reunião, o *product owner* identifica quais itens do *backlog* foram realizados no *sprint* e quais não foram. Em seguida, o time de desenvolvimento revisa os pontos positivos e negativos do *sprint*, informa como os problemas foram resolvidos e apresenta os itens do *backlog* que foram concluídos neste *sprint*. Por fim, todos os

participantes colaboram para definir quais serão os próximos passos, fornecendo insumos para a reunião de revisão do próximo *sprint* (SCHWABER *et al.*, 2011).

O resultado desta reunião é um *backlog* do produto revisado por todos os envolvidos no projeto, definindo uma provável lista de itens que serão realizados no próximo *sprint*. Se alguma necessidade do negócio surgir durante o projeto, ela pode ser incluída no *backlog* do produto neste momento (SCHWABER *et al.*, 2011).

2.2.5.5. A REUNIÃO DE RETROSPECTIVA DO SPRINT (SPRINT RETROSPECTIVE)

Diferente da reunião de revisão do *sprint*, cujo foco é o produto, a retrospectiva do *sprint* tem foco no processo, sendo uma oportunidade para toda a equipe Scrum se inspecionar e criar um plano de melhoria para o próximo *sprint*. Ela deve ocorrer após a reunião de revisão do *sprint* atual e antes da reunião de planejamento do próximo *sprint*, tendo uma duração aproximada de três horas para um *sprint* de um mês (SCHWABER *et al.*, 2011).

Esta reunião tem como objetivo inspecionar como foi o andamento do último *sprint* com relação às pessoas, aos relacionamentos da equipe, ao processo e às ferramentas utilizadas. Além disso, também busca identificar melhorias que deram certo no último *sprint* e identificar novos pontos de melhoria para realização do trabalho da equipe (SCHWABER *et al.*, 2011).

O *Scrum master* deve incentivar toda a equipe a buscar processos e práticas para tornar o trabalho mais efetivo e prazeroso, baseando-se nos pilares do Scrum (SCHWABER *et al.*, 2011).

2.2.6. ARTEFATOS DO SCRUM

Os artefatos do prescritos pelo Scrum têm como principal objetivo maximizar a transparência das informações para toda a equipe, garantindo o sucesso na entrega de um incremento de produto "pronto". Os principais artefatos são: o *backlog* do produto; o *backlog* do *sprint*; o gráfico de *burndown* e o incremento do produto (SCHWABER *et al.*, 2011).

2.2.6.1. O BACKLOG DO PRODUTO (PRODUCT BACKLOG)

O *backlog* do produto é uma lista ordenada que contém todas as características ou funcionalidades desejadas em um produto. Seu conteúdo é definido pelo *product owner*,

que também é responsável por detalhar e priorizar os itens desta lista (SCHWABER *et al.*, 2011).

O conteúdo do *backlog* do produto é dinâmico e não precisa estar completo no início do projeto. O desenvolvimento pode ser iniciado com todas as características que são mais óbvias em um primeiro momento e no decorrer do projeto, à medida que se aprende mais sobre o produto e seus usuários, seu conteúdo pode crescer e mudar (SCHWABER *et al.*, 2011).

A ordenação dos itens nesta lista geralmente é feita por riscos, prioridades ou retorno ao negócio. Os itens que estão no topo da lista são os mais prioritários e devem estar com o maior nível de detalhe possível para serem encaminhados para o desenvolvimento, incluindo o esforço estimado para desenvolvê-lo. Os itens menos prioritários podem estar com um nível de detalhamento menor, já que serão melhor estimados e detalhados quando forem priorizados (SCHWABER *et al.*, 2011).

A Figura 3 demonstra um exemplo de *backlog* do produto:

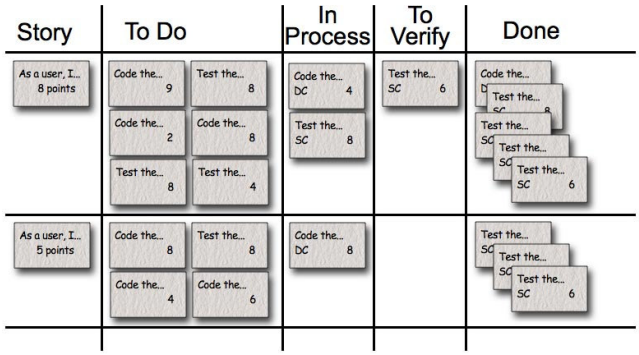

Figura 3: Exemplo de um backlog do produto.
(MONTAIN GOAT SOFTWARE, 2013)

2.2.6.2. O BACKLOG DO SPRINT (SPRINT BACKLOG)

O *backlog* do *sprint* é uma lista que contém os itens do *backlog* do produto que foram selecionados para o desenvolvimento. Na reunião de planejamento do *sprint*, o time de desenvolvimento seleciona a quantidade de itens do topo do *backlog* do produto que poderão ser desenvolvidas no próximo *sprint*, transportando-os para o *backlog* do *sprint* (SCHWABER *et al.*, 2011).

O *backlog* do *sprint* informa e torna visível todo o trabalho que o time de desenvolvimento deve realizar para transformar os itens do *backlog* do produto em um

incremento de produto “pronto”, atingindo o objetivo do *sprint* (SCHWABER *et al.*, 2011).

É possível realizar um monitoramento diário utilizando gráficos de acompanhamento do trabalho em andamento, facilitando a identificação de possíveis atrasos no desenvolvimento. Se um atraso for identificado em um dia, na reunião próxima reunião diária este desvio pode ser corrigido, reduzindo o impacto na entrega final. O gráfico de *burndown*, que será detalhado a seguir, é uma forma de realizar este acompanhamento (SCHWABER *et al.*, 2011).

2.2.6.3. O GRÁFICO DE BURNDOWN (BURNDOWN CHART)

O gráfico de *burndown* é utilizado pelas equipes Scrum para representar diariamente o progresso do trabalho em desenvolvimento no *sprint*. Ele permite à equipe ter visibilidade do seu ritmo e verificar se ele está adequado para atingir a meta do *sprint*, cumprindo com o que foi planejado (SCHWABER *et al.*, 2011).

No eixo horizontal, o gráfico marca os dias do *sprint* em ordem crescente e no eixo vertical são representadas as quantidades de trabalho a realizar (em pontos estimados, dias, horas etc.), também em ordem crescente. Uma diagonal é traçada a partir do primeiro dia, onde nenhum trabalho foi realizado e, portanto, a quantidade de trabalho pendente é a total do *backlog*, até o último dia, onde não deve restar nenhum trabalho pendente. Esta diagonal é a meta da equipe durante o *sprint*: não ter nenhum trabalho pendente no último dia do *sprint* (SCHWABER *et al.*, 2011).

Ao longo dos dias, o time de desenvolvimento marca no gráfico a quantidade de trabalho pendente. Com isto, é possível verificar se há atrasos na entrega observando a linha de trabalho realizado em relação à diagonal traçada: se a linha estiver à esquerda da diagonal, a equipe está adiantada neste dia; se a linha estiver à direita da diagonal, a equipe está atrasada em relação à meta, sendo necessário tomar alguma ação (SCHWABER *et al.*, 2011).

Este simples acompanhamento garante a visualização do progresso e permite identificar possíveis atrasos a partir do primeiro dia em que eles surgirem (SCHWABER *et al.*, 2011). A Figura 4 mostra um exemplo de gráfico de *burndown*.

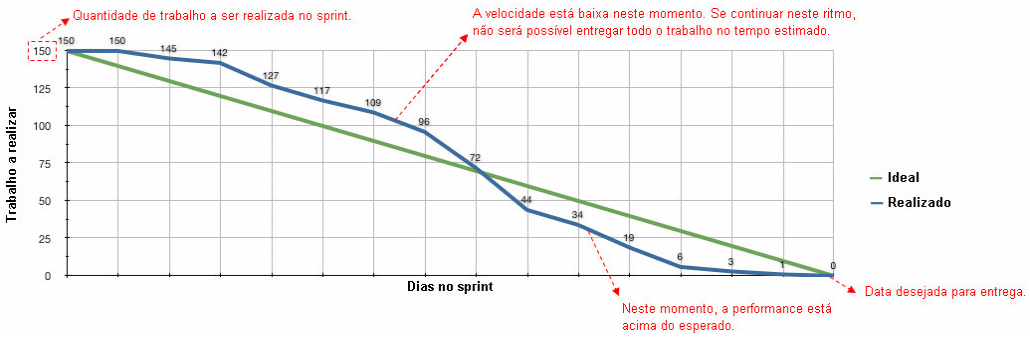

Figura 4: Exemplo de um gráfico de burndown.
Fonte: o autor.

2.2.6.4. O INCREMENTO DO PRODUTO

O incremento do produto é a soma de todos os itens do *backlog* do produto concluídos no último *sprint* e em todos os *sprints* anteriores. Ao final de um *sprint*, o novo incremento deve estar "pronto", isto é, deve estar em condições de uso e atender à definição de "pronto" definida pela equipe (SCHWABER *et al.*, 2011).

2.3. DESENVOLVIMENTO ENXUTO DE SOFTWARE

Os conceitos do desenvolvimento enxuto de software (do inglês, *lean software development*) são baseados nos conceitos de produção enxuta (do inglês, *lean manufacturing*) e desenvolvimento enxuto de produtos (do inglês, *lean product development*). A produção enxuta tem foco exclusivo na eliminação de desperdícios do processo de produção. Desperdícios podem ser considerados como qualquer esforço que não contribui com a criação de valor para o cliente. Já o desenvolvimento enxuto de produtos não tem foco exclusivo no processo de produção, mas também nas demais etapas de criação do produto, desde a criação do conceito até sua entrega final. Esta abordagem é muito mais próxima da realidade da engenharia de *software* (PETERSEN, 2010).

Em 2003, Marry e Tom Poppendieck publicaram o livro "*Lean Software Development: An Agile Toolkit*", que descrevia como os processos de produção enxuta e de desenvolvimento enxuto de produtos poderiam ser adaptados para o ambiente de desenvolvimento de *software*, dando origem ao termo "*Lean Software Development*", ou desenvolvimento enxuto de *software (*POPPENDIECK *et al.*, 2003). Este livro foi utilizado como base para detalhar os conceitos sobre o desenvolvimento enxuto de software, descritos a seguir.

2.3.1. PRINCÍPIOS APLICADOS AO DESENVOLVIMENTO DE SOFTWARE

Em seu livro, POPPENDIECK *et al.* (2003) citam as diferenças entre princípios e práticas. Princípios são idéias e percepções a respeito de alguma disciplina, enquanto as práticas são as ações realizadas para atender aos princípios. Princípios são universais, mas o entendimento sobre como eles são aplicados a um ambiente particular nem sempre é trivial. As práticas, no entanto, dão orientações específicas sobre o que deve ser realizado, mas elas precisam ser adaptadas ao domínio de aplicação (POPPENDIECK *et al.*, 2003).

Os problemas gerados ao aplicar conceitos de outras áreas de conhecimento ao desenvolvimento de *software* são causados, muitas vezes, pela tentativa de transferir as práticas ao invés dos princípios das outras disciplinas (POPPENDIECK *et al.*, 2003). Os autores abordam os sete princípios do *Lean Manufacturing* adaptados para a realidade do desenvolvimento de *software*, descritos a seguir.

2.3.1.1. PRINCÍPIO 1: ELIMINE DESPERDÍCIOS

Serão considerados como desperdícios quaisquer esforços que não adicionam valor ao produto, do ponto de vista do cliente. No pensamento enxuto, desperdícios são considerados grandes obstáculos para o sucesso da sua execução. Se um produto foi produzido e se manteve no estoque "acumulando poeira", o esforço de sua produção foi um desperdício. Da mesma forma, se um ciclo do desenvolvimento coletou requisitos e estes não foram implementados (permanecendo no "estoque", "acumulando poeira"), este esforço também foi um desperdício. Se uma fábrica produz mais elementos do que o necessário naquele momento, estes permanecerão no estoque e serão um desperdício. Da mesma forma, se os desenvolvedores codificam mais funcionalidades do que as necessárias para aquele momento, também é desperdício (POPPENDIECK *et al.*, 2003).

O ideal é identificar quais são as necessidades do cliente e, então, desenvolver e entregar exatamente o que eles precisam, o mais rápido possível. Qualquer outro esforço que não esteja relacionado a satisfazer rapidamente a necessidade do cliente é considerado desperdício (POPPENDIECK *et al.*, 2003).

2.3.1.2. PRINCÍPIO 2: ESTIMULE O APRENDIZADO

O desenvolvimento é um exercício de descoberta, enquanto a produção é um exercício de redução da variação. Por esta razão, a abordagem *lean* aplicada ao

desenvolvimento resulta em práticas diferentes das práticas voltadas à produção enxuta (POPPENDIECK *et al.*, 2003).

POPPENDIECK *et al.* (2003) utilizam a seguinte metáfora para descrever este cenário: "Desenvolver é como criar uma receita, enquanto produzir é como criar o prato". Receitas são projetadas por *chefs* experientes que desenvolveram um instinto para este trabalho e a capacidade de adaptar os ingredientes disponíveis para aplicar em cada ocasião. No entanto, até grandes *chefs* criam diferentes versões de um prato até conseguirem uma receita que tenha um bom gosto e seja fácil de reproduzir. Não é esperado que um *chef* faça uma receita perfeita na primeira tentativa, mas é esperado que ele tente algumas variações como parte de seu processo de aprendizagem (POPPENDIECK *et al.*, 2003).

O desenvolvimento de *software* é melhor concebido como um processo de aprendizagem similar ao descrito no parágrafo anterior, com o desafio adicional de que as equipes de desenvolvimento são maiores e os resultados são muito mais complexos do que uma receita. A melhor abordagem para melhorar o ambiente de desenvolvimento de software é estimular o processo de aprendizagem (POPPENDIECK *et al.*, 2003).

2.3.1.3. PRINCÍPIO 3: TOME DECISÕES O MAIS TARDE POSSÍVEL

As práticas de desenvolvimento que prezam pela tomada de decisão tardia são efetivas em ambientes que envolvem a incerteza, pois elas possuem uma abordagem baseada em opções. Em face à incerteza, muitos mercados econômicos desenvolvem opções para permitir que os investidores evitem a tomada de decisão até que o futuro seja mais próximo e fácil de prever (POPPENDIECK *et al.*, 2003).

Adiar decisões é um recurso valioso porque as melhores decisões são feitas baseadas em fatos, e não em especulações. Em um mercado em evolução, manter as opções de *design* abertas é mais valioso do que defini-las antecipadamente. Portanto, uma estratégia chave para adiar comprometimentos ao desenvolver um sistema complexo é torna-lo receptivo à mudanças (POPPENDIECK *et al.*, 2003).

2.3.1.4. PRINCÍPIO 4: ENTREGUE O PRODUTO O MAIS RÁPIDO POSSÍVEL

Há até pouco tempo, o desenvolvimento rápido de software não era valorizado e a abordagem "não cometa nenhum erro" parecia ser mais importante. Porém, é hora de incluir este como um mito desmascarado e identificar suas vantagens, pois sem a

velocidade, não é possível adiar decisões nem obter *feedbacks* confiáveis (POPPENDIECK *et al.*, 2003).

No desenvolvimento, o ciclo de descobertas (projetar, implementar, avaliar e melhorar) é fundamental para o aprendizado. Portanto, quanto mais curto e rápido for este ciclo, maior será o aprendizado, além de garantir que os clientes recebam o que eles precisam agora, e não o que eles precisavam ontem. A redução deste ciclo é a estratégia fundamental para eliminar desperdícios (POPPENDIECK *et al.*, 2003).

2.3.1.5. PRINCÍPIO 5: DÊ PODER DE DECISÃO À EQUIPE

O segredo da execução de alto nível está na compreensão correta dos detalhes, e ninguém entende dos detalhes melhor do que as pessoas que realmente fazem o trabalho. Portanto, envolver os desenvolvedores nos detalhes de decisões técnicas é fundamental para alcançar a excelência (POPPENDIECK *et al.*, 2003).

Os membros da equipe de desenvolvimento aliam o conhecimento técnico dos detalhes com o poder de muitas mentes. Quando possuem as competências necessárias e são guiados por um bom líder, eles poderão tomar as decisões técnicas e de processos melhor do que qualquer outra pessoa (POPPENDIECK *et al.*, 2003).

O papel de uma pessoa gerenciando as atividades da equipe não é possível neste ambiente, já que as decisões são tomadas o mais tarde possível e os ciclos de desenvolvimento são rápidos. Para evitar este gerenciamento de atividades, as práticas enxutas buscam oferecer versões cada vez mais refinadas do produto em intervalos regulares, facilitando o planejamento do trabalho, e descrevem sinalizações no local de trabalho para que todos os integrantes da equipe tenham conhecimento do que precisa ser feito, através de gráficos visíveis, reuniões diárias, integrações contínuas e testes abrangentes (POPPENDIECK *et al.*, 2003).

2.3.1.6. PRINCÍPIO 6: INCLUA A QUALIDADE NO PROCESSO

A qualidade deve ser tratada com prioridade e não é negociável, devendo estar presente em todas as etapas do processo. Para o cliente, a qualidade deve ser tanto intrínseca como explícita e quanto mais o cliente percebê-la, mais qualidade terá o produto final (POPPENDIECK *et al.*, 2003).

A qualidade explícita é chamada de integridade percebida, pois é um indicador de equilíbrio entre as funções, a usabilidade, a confiabilidade, a economia e a percepção do

cliente. Se o cliente diz: “Perfeito! Foi como eu imaginei que seria.”, então o software possui a integridade percebida (POPPENDIECK *et al.*, 2003).

A qualidade intrínseca é chamada de integridade conceitual. Neste caso, os conceitos centrais do sistema devem trabalhar em conjunto, devem ser fáceis de entender e possuir alta coesão. A integridade conceitual é um fator crítico para o sucesso da integridade percebida (POPPENDIECK *et al.*, 2003).

2.3.1.7. PRINCÍPIO 7: OTIMIZE O TODO

É necessário otimizar todas as etapas do processo, desde o começo até o final, utilizando métricas de desempenho da equipe para identificar problemas e melhorar o processo. Ao estimar o tempo das atividades, é recomendado incluir uma margem de segurança para não gerar atrasos (POPPENDIECK *et al.*, 2003).

O foco principal deve ser sempre a satisfação do cliente, buscando atender em suas reais necessidades e evitando quaisquer desperdícios (POPPENDIECK *et al.*, 2003).

2.4. KANBAN

O método Kanban surgiu no Japão com o Sistema Toyota de Produção (ou TPS, do inglês, *Toyota Production System*) (OHNO, 1997, *apud* SILVA *et al.*, 2013) para gerenciar a fabricação de automóveis. Diferente da abordagem das indústrias americanas, os japoneses implementaram um sistema de produção denominado sistema puxado (*pull system*) onde a demanda sinaliza quando deve-se produzir mais, ditando o ritmo de produção. Com isso, a indústria adaptou sua velocidade de produção de acordo com o nível de consumo dos clientes, evitando criação de estoques desnecessários (SILVA *et al.*, 2013).

O uso do Kanban para o gerenciamento de equipes de desenvolvimento de *software* foi mais frequente a partir de 2007, quando Rick Garber e David J. Anderson publicaram nas conferências “*Lean New Product Development*” e “*Agile 2007*” os resultados obtidos no uso deste método no desenvolvimento de *software*. Desde então, o uso deste método com este foco vem sendo estudado e experimentado (SILVA *et al.*, 2013).

Kanban é um termo japonês que significa sinal visual. Uma das grandes características deste método é evidenciar os problemas existentes no processo (SILVA *et al.*, 2013). Dentre as metodologias ou *frameworks* para desenvolvimento de *software* abordados nesta monografia, o Kanban é a menos prescritiva, tornando a sua utilização

mais adaptável. Com isso, as equipes que o adotam precisam estar atentas ao processo aplicado para visualizar pontos de melhoria e adaptações, para que o processo possa fluir de forma satisfatória.

Segundo KNIBERG *et al.* (2009), o Kanban tem apenas três prescrições: visualize o fluxo de trabalho atual; limite o fluxo de trabalho; acompanhe e gerencie o fluxo de trabalho.

2.4.1. VISUALIZE O FLUXO DE TRABALHO ATUAL

Para atender à primeira prescrição do Kanban é importante ressaltar que o fluxo de trabalho a ser visualizado deve ser aquele que de fato ocorre e não o que formalmente é definido pela organização. Em muitas organizações, apesar de haver um processo oficial, geralmente as equipes seguem outro modelo, definindo processos internos (KNIBERG *et al.*, 2009).

2.4.2. LIMITE O FLUXO DE TRABALHO;

Para limitar o fluxo de trabalho (também chamado de WIP, do inglês *Work in Progress*) é necessário explicitar quantos itens de trabalho devem estar em cada uma das fases do processo. Esse artifício é um dos pontos-chave do método, uma vez que ele é o responsável por definir o Kanban como sistema puxado (sua capacidade é definida pela demanda). Apenas quando um item sair de uma fase é que esta fase poderá receber outro item (KNIBERG *et al.*, 2009).

2.4.3. ACOMPANHE E GERENCIE O FLUXO DE TRABALHO.

Na terceira prescrição, é definida uma forma de medir e controlar o fluxo de trabalho. É neste ponto que as equipes de desenvolvimento realizam adaptações para, de acordo com os problemas evidenciados, propor formas de controlá-los e contorná-los (KNIBERG *et al.*, 2009).

Uma prática comum para visualização de atividades é a representação em um quadro, onde cada coluna informa a etapa do processo e cada item (ou “cartão”) corresponde a uma atividade em sua etapa respectiva (KNIBERG *et al.*, 2009), conforme exemplo da Figura 5. Os números exibidos abaixo de cada coluna representam o WIP determinado nesta etapa.

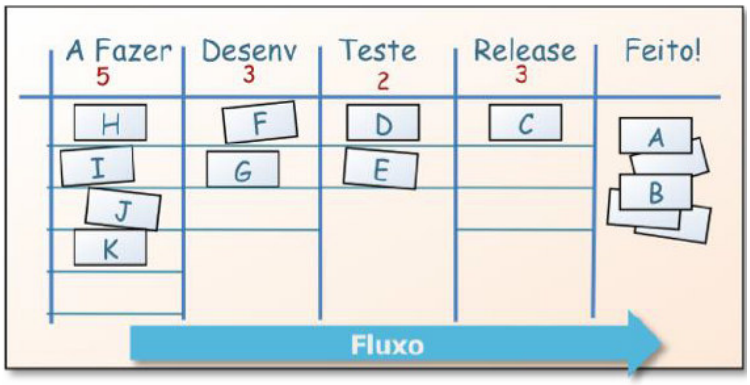

Figura 5: Exemplo de um quadro de atividades.
Fonte: (KNIBERG et al., 2009)

O Kanban, por ser mais adaptativo do que prescritivo, acaba se tornando bastante empírico. As fases do processo em questão e os valores limitados de itens de trabalho para cada fase devem ser testados pela equipe de forma a encontrarem o valor ideal do WIP. Não há uma fórmula definida para chegar a esse valor, pois cada equipe tem suas características. A equipe deve experimentar e encontrar os números que melhor se adequem à sua realidade (KNIBERG *et al.*, 2009).

Dadas as prescrições, percebe-se que o Kanban é baseado em um processo onde o fluxo de trabalho é contínuo, diferente do Scrum, controlando as entradas de itens de trabalho e a vazão que é dada de acordo com o WIP definido. A esta vazão dos itens de trabalho dá-se o nome de *lead time*, que representa o tempo de um item de trabalho desde a sua entrada no fluxo de trabalho mapeado até a sua saída (KNIBERG *et al.*, 2009).

O Kanban permite a combinação de ferramentas de diversos métodos até se obter o processo adequado, baseando-se fortemente no pensamento *Lean* (POPPENDIECK *et al.*, 2003) e estimulando a melhoria contínua do processo, de forma a tornar possível dar respostas rápidas ao cliente (KNIBERG *et al.*, 2009).

2.5. PRÁTICAS ÁGEIS

Para apoiar a identificação e aplicação das práticas ágeis no estudo de caso proposto nesta monografia, foram revisados quatro artigos que abordam as práticas ágeis utilizadas em projetos de desenvolvimento de *software*: WILLIAMS (2010), JALALI *et al.* (2010), ABRANTES *et al.* (2011) e KURAPATI *et al.* (2012).

Por consolidar as práticas ágeis mais comuns, o artigo de ABRANTES *et al.* (2011) foi utilizado como base para a comparação de todas as demais práticas, gerando a

Tabela 1, que contém todas as práticas citadas agrupadas por similaridade e características. Os termos foram mantidos em inglês pois algumas práticas similares são citadas com nomes distintos por cada autor.

Tabela 1: Relação das práticas ágeis abordadas nos artigos revisados.

ABRANTES E TRAVASSOS (2011)	WILLIAMS (2010)	KURAPATI et al (2012)	JALALI E WOHLIN (2010)
Coding standards	Code and Tests Nightly Build Ten-Minute Build	Coding standards	Code standards
Continuous integration	Continuous Integration	Continuous integration	Continuous integration
Pair programming	Pair Programming	Pair programming	Pair programming
Planning Game	Planning Poker Wideband Delphi Estimation	Planning game	Planning game
Project visibility	Informative Workspace	Tracking progress	Burndown charts Virtual scrum wall
Refactoring	Code and Tests Executable Documentation	Refactoring	Refactoring Code reviews
Small releases	Short Iterations Short Releases Sprint Incremental Design	Short / small releases Sprint / iteration	Short iteration
Stand-up meetings	Stand-Up Meeting	Stand-ups	Stand-up meetings
Test Driven Development	Acceptance Test-Driven Development Unit Test-Driven Development	Test driven development	TDD
Collective Code Ownership	Collective Code Ownership Code Ownership	Collective Ownership	
Metaphor		Metaphors	System metaphor
On-site customer	Sit Together Negotiated Scope		Close collaboration Proxy customer
Product Backlog	Release and Iteration Backlog		Backlog
Sustainable Pace (40 hour week)	Sustainable Pace Energized Work	40 hour week	
Whole team (multi-skill teams)	Whole Team	Team	
	Retrospective	Retrospective	Retrospectives
	Scrum Meeting	Sprint planning meeting Sprint review meeting Informative workshops	Planning meeting Sprint review
	Features Stories	Stories / Features	User stories
	Automation-Driven Root Cause Analysis of Failures Iteration Demonstration Inspections	Testing	Automated testing Acceptance tests Unit testing
Open workspace		Office structure that supports agile development	
Simple design		Simple design	
		Communication	Instant messages
	Done Criteria		
		Configuration and Change management	
		Documentation	
			Feature driven development
			Scrum of scrums
			Sprint demo

Como não há um consenso sobre as práticas ágeis mais utilizadas, foram selecionadas as práticas citadas pela maioria dos artigos analisados. Esta seleção resultou em 19 práticas ágeis que foram utilizadas como opções para a aplicação da proposta desta monografia, sendo 9 práticas comuns a todos os artigos analisados e 10 práticas abordadas por ao menos 3 dos 4 artigos analisados. Elas foram agrupadas de acordo com seu foco:

- **Práticas relacionadas com os requisitos do produto:**
  - Divisão em funcionalidades (*Features / Stories*)
  - *Backlog* do produto (*Product Backlog*)
- **Prática relacionada com a fase de design do produto:**

    - Metáforas do sistema (*Metaphor*)
- **Práticas relacionadas com a fase de construção do produto:**
    - Padrões de codificação (*Coding standards*)
    - Código coletivo (*Collective Code Ownership*)
    - Integração contínua (*Continuous integration*)
    - Programação em par (*Pair programming*)
    - Refatoração do código (*Refactoring*)
    - Entregas curtas (*Small releases*)
    - Desenvolvimento dirigido por testes (*Test Driven Development*)
- **Prática relacionada com a fase de testes do produto:**
    - Automação de testes (*Automated testing*)
- **Práticas relacionadas com a organização e o ambiente de trabalho:**
    - Cliente presente (*On-site customer*)
    - Ritmo sustentável (*Sustainable Pace / 40 hour week*)
    - Times multidisciplinares (*Whole team / multi-skill teams*)
- **Práticas relacionadas com o gerenciamento do projeto:**
    - Jogo do planejamento (*Planning Game*)
    - Visibilidade do projeto (*Project visibility*)
    - Retrospectivas (*Retrospective*)
    - Reuniões Scrum (*Scrum Meetings*) e Reuniões de pé (*Stand-up meetings*)

2.6. CONSIDERAÇÕES FINAIS

Este capítulo apresentou uma revisão da literatura sobre as metodologias e *frameworks* ágeis, além de revisar as práticas ágeis mais comuns em projetos de *software*. Os conceitos e práticas revisados poderão ser utilizados na proposta de melhoria de processos, de acordo com o planejamento dos ciclos apresentados nos próximos capítulos.

3. O CENÁRIO INICIAL DA ORGANIZAÇÃO

Este capítulo apresenta a descrição da iniciativa de melhoria e do cenário inicial da empresa onde o estudo de caso foi aplicado, apresentando a estrutura organizacional, a percepção das equipes sobre o processo de desenvolvimento de *software*, os indicadores da necessidade de mudanças.

3.1. DESCRIÇÃO DA INICIATIVA DE MELHORIA

Para atingir os objetivos propostos por esta monografia, a estratégia selecionada para a iniciativa de melhoria foi decompor a proposta em ciclos, para minimizar a rejeição da equipe às mudanças na rotina de trabalho e proporcionar um período de adaptação e aprendizado entre cada ciclo.

A iniciativa de melhoria foi iniciada a partir da revisão das práticas ágeis selecionadas na Seção 2.5, visando identificar quais práticas seriam aplicáveis ao ambiente da organização. Após identificação das práticas que seriam aplicáveis na organização, foram propostas melhorias na gestão de atividades da equipe, visando melhorar a visualização das atividades, facilitar a alocação de recursos e permitir a identificação de gargalos. Em seguida, o foco foi dado na melhoria da comunicação entre as equipes e no ajuste de atividades paralelas, visando trazer transparência ao processo e ajustar a capacidade de trabalho da equipe. Por fim, foram propostas mais práticas relacionadas ao ambiente de trabalho e ao desenvolvimento do produto durante a execução de dois projetos piloto no ambiente adaptado às melhorias realizadas anteriormente.

Ao final das melhorias, foi realizada uma retrospectiva com as equipes envolvidas para avaliar as mudanças propostas no ambiente e verificar quais foram os pontos positivos e negativos de cada proposta.

3.2. ESTRUTURA DA ORGANIZAÇÃO

O estudo de caso foi realizado em uma empresa privada de grande porte do segmento de telecomunicações, que atua no Brasil há 18 anos e possui como foco a telefonia móvel e a banda larga. A empresa possui mais de 10.000 colaboradores em todo o país, alocados principalmente nos estados do Rio de Janeiro e São Paulo.

A área selecionada para execução do projeto foi a Gerência de Tecnologia da Informação responsável pelo desenvolvimento e manutenção de sistemas relacionados ao registro e cobrança de serviços de telefonia fixa.

3.2.1. EQUIPES

Na área de TI, a gerência selecionada possui duas equipes com foco em diferentes etapas do ciclo de receita da empresa. Ambas possuem cinco integrantes, dentre os quais um é coordenador de sua respectiva equipe. Cada integrante é responsável por um ou mais sistemas, que podem estar relacionados a uma ou mais áreas de negócio. Como a equipe é reduzida, é comum ocorrer alocação dos integrantes em projetos que não se relacionam diretamente com o sistema sob sua responsabilidade.

O projeto foi desenvolvido em uma destas equipes, que será denominada Área Técnica, composta por um coordenador, um analista sênior e dois analistas plenos, sendo o autor desta monografia um dos analistas plenos da equipe. Os membros da equipe participam ativamente de quase todas as etapas do ciclo de desenvolvimento de *software*, realizando atividades de análise de requisitos, definição de soluções técnicas, validação de propostas de fornecedores, planejamento e execução de testes e gestão de projetos. A única etapa do processo que não é realizada ativamente pela equipe é a codificação, que é desempenhada por uma Fábrica de Software contratada pela empresa. Todos os membros da equipe participaram do projeto proposto nesta monografia.

A Fábrica de Software que atende a Área Técnica selecionada possui três integrantes, sendo dois desenvolvedores seniores e um desenvolvedor júnior. Um dos seniores também é responsável pela coordenação da equipe. Todos os membros desta equipe participaram do projeto.

A Área Técnica possui como principais clientes internos três áreas de negócio distintas: interconexão de redes, garantia de receita e gestão de riscos. A Área de Negócio convidada para participar do projeto foi a de gestão de riscos, por ser a menor e mais receptiva a experimentar novos métodos.

A área de gestão de riscos possui sete integrantes, sendo um gerente, dois analistas seniores, uma assistente sênior e três assistentes juniores. A execução deste projeto envolveu o gerente e um analista sênior, por sugestão do próprio gerente da área. O papel de usuário durante o projeto será desempenhado pelo analista sênior e o andamento será acompanhado pelo gerente da área.

A Tabela 2 resume a equipe participante da iniciativa de melhoria.

Tabela 2: Equipe participante da iniciativa de melhoria.

Área de Negócio	Área Técnica	Fábrica de Software

1 gerente 1 analista sênior	1 coordenador 1 analista sênior 2 analistas plenos	1 desenvolvedor sênior / coordenador 1 desenvolvedor sênior 1 desenvolvedor júnior
Total: 2 integrantes	**Total:** 4 integrantes	**Total:** 3 integrantes
Equipe da iniciativa de melhoria: 9 integrantes		

Para auxiliar a caracterização das equipes, foram aplicados dois questionários para obter o perfil de cada integrante e entender suas expectativas com a condução dos projetos de desenvolvimento de *software* na empresa.

O primeiro questionário, que será denominado Questionário A, foi direcionado para a equipe da Área de Negócio, visando obter a percepção dos entrevistados sobre a efetividade do processo de desenvolvimento de *software* da organização e contendo uma matriz para identificar quais os critérios mais relevantes para se obter sucesso em um projeto de desenvolvimento. Os dois membros da equipe da Área de Negócio participaram do preenchimento do questionário.

O segundo questionário, que será denominado Questionário B, foi direcionado para as equipes da Área Técnica e da Fábrica de *Software*, contendo as mesmas seções do Questionário A, além de perguntas sobre a experiência do entrevistado na área de TI, os papéis desempenhados durante sua experiência e seus conhecimentos sobre práticas, metodologias ou *frameworks* ágeis. Os quatro integrantes da equipe da Área Técnica participaram do preenchimento do Questionário B. Por falta de disponibilidade de todos os membros da equipe da Fábrica de *Software*, apenas o coordenador participou do preenchimento do Questionário B, representando a visão de toda sua equipe.

Os resultados da aplicação do Questionário B relacionados à caracterização das equipes da Área Técnica e da Fábrica de *Software* serão apresentados a seguir. Os demais resultados do Questionário B e todos os resultados do Questionário A, serão apresentados na Seção 3.3 por serem relacionados à percepção dos integrantes sobre o processo de desenvolvimento e sobre os critérios de sucesso em um projeto.

- **Experiência dos entrevistados na área de TI:**

Para caracterizar a experiência das equipes da Área Técnica e Fábrica de *Software*, foi perguntado quantos anos de experiência o entrevistado possui na área de TI. As opções disponíveis para resposta foram: Menos de 2 anos; 2 a 5 anos; 5 a 10 anos; Mais de 10 anos.

Todos os integrantes das equipes da Área Técnica e da Fábrica de Software possuem experiência superior a cinco anos na área de TI, como pode ser visto na Figura 6. Este fato auxilia a execução do projeto por contar com uma equipe com certo nível de maturidade profissional.

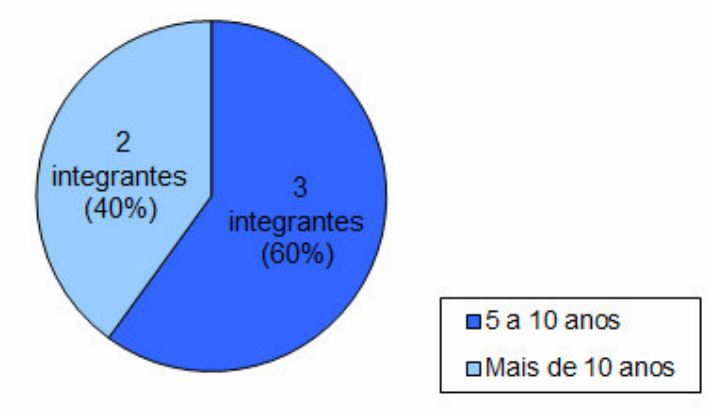

Figura 6: Experiência das equipes na área de TI.

- **Experiência e atuação dos entrevistados nos papéis de TI:**

Para caracterizar a experiência dos integrantes das equipes da Área Técnica e Fábrica de *Software* nos papéis de TI, foi solicitado que o entrevistado indicasse sua experiência para cada papel apresentado: Analista de Produção; Analista de Sistemas; Analista de Testes; Desenvolvedor; Gerente de Projeto ou Outros. As opções disponíveis para resposta foram: Nenhuma; Menos de 2 anos; 2 a 5 anos; 5 a 10 anos ou Mais de 10 anos.

Os papéis de Gerente de Projetos, Analista de Sistemas e Desenvolvedor foram os mais desempenhados pelos integrantes da Área Técnica e da Fábrica de Software entrevistados durante toda sua experiência profissional, como pode ser visto na Figura 7.

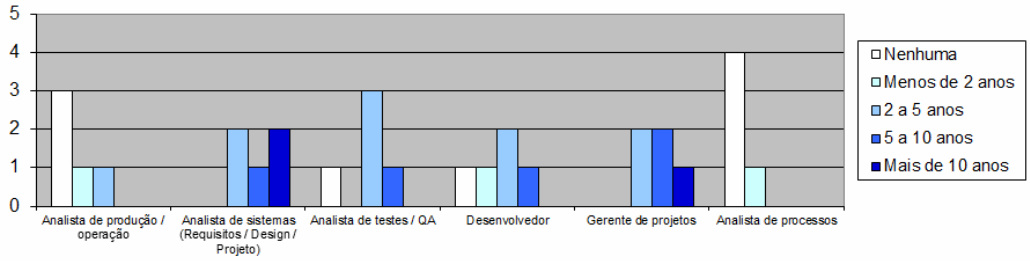

Figura 7: Experiência das equipes nos papéis da área de TI.

Para identificar os papéis mais desempenhados na organização atual, foi solicitado que o entrevistado indicasse os papéis que mais se adequassem à sua posição atual na empresa. Os papéis apresentados como opção foram os mesmos da questão anterior.

Como resultado, os papéis mais desempenhados na organização atual foram o de Gerente de Projetos, Analista de Sistemas e Analista de Testes, como pode ser visto na Figura 8.

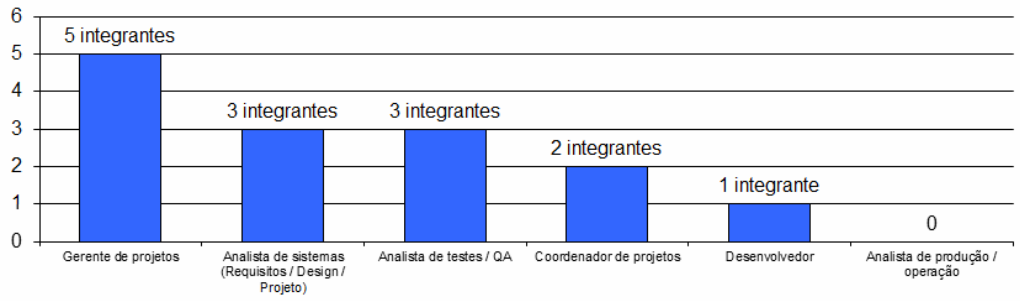

Figura 8: Papel atual das equipes na área de TI.

A experiência e papel atual dos integrantes entrevistados são um ponto positivo na execução do projeto, pois estão de acordo com as propostas da iniciativa de melhoria, que terão foco principal no processo e no projeto de desenvolvimento de *software*.

- **Conhecimento das metodologias, *frameworks* e práticas ágeis:**

Para caracterizar o conhecimento dos integrantes das equipes da Área Técnica e Fábrica de *Software* sobre as metodologias e *frameworks* ágeis, foi solicitado que o entrevistado indicasse as opções que conhecia: *Extreme Programming* (XP); Kanban; Desenvolvimento Enxuto de *Software* (*Lean)*; Scrum ou Outros.

Todos os integrantes entrevistados das equipes da Área Técnica e da Fábrica de Software conhecem, de alguma forma, o Scrum e o XP (*Extreme Programming*). Dois integrantes conhecem o Kanban. Já o desenvolvimento enxuto de *software* não é conhecido por nenhum integrante entrevistado, como pode ser visto na Figura 9.

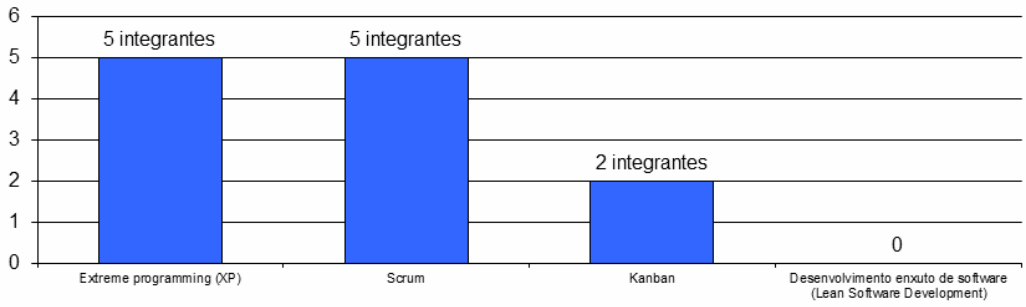

Figura 9: Conhecimento das equipes sobre as metodologias e frameworks ágeis.

Com relação ao conhecimento sobre as práticas ágeis, foi solicitado que o entrevistado indicasse quais práticas conhecia, dentre as opções apresentadas. As práticas exibidas como opção foram as mesmas identificadas na Seção 2.5.

Todas as práticas apresentadas aos entrevistados são conhecidas por ao menos um dos integrantes entrevistados. Além destas práticas, um dos entrevistados citou a definição de "pronto" como uma prática conhecida, conforme exibido na Figura 10.

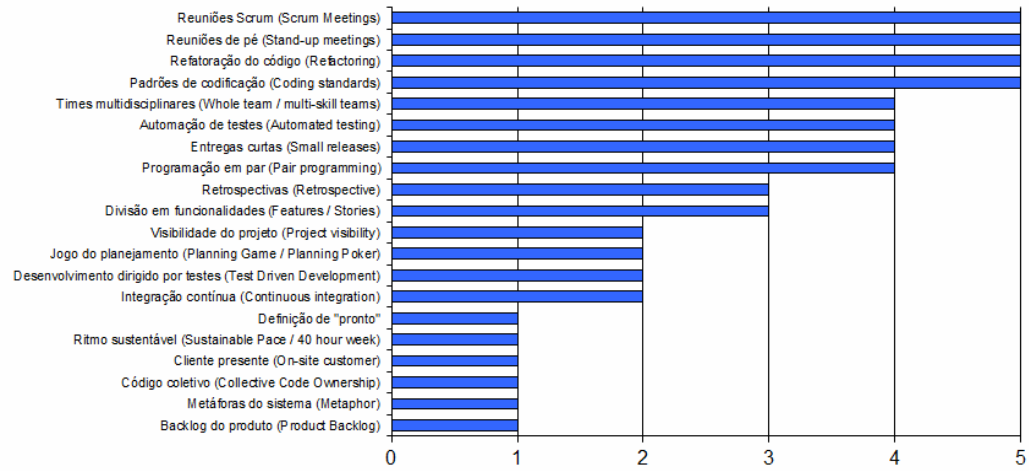

Figura 10: Conhecimento das equipes sobre as práticas ágeis.

3.2.2. PROCESSO DE DESENVOLVIMENTO

O processo de desenvolvimento de *software* utilizado na empresa é baseado no modelo cascata, conforme visão macro exemplificada na Figura 11.

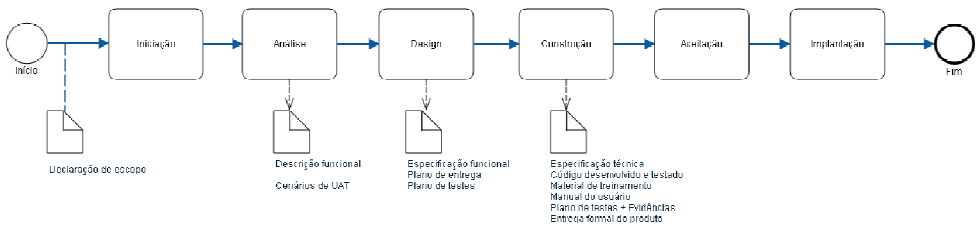

Figura 11: Visão macro do processo de desenvolvimento utilizado na empresa.

O processo inicia-se a partir do recebimento do documento de declaração de escopo da Área de Negócio, que contém todas as características desejadas do produto e serve de insumo para a primeira etapa, o estágio de iniciação. Existe um processo anterior de

priorização e liberação do documento de declaração de escopo, mas é de responsabilidade de cada Área de Negócio e não está no escopo da área de TI.

O estágio de iniciação é dividido em três fases. Na primeira fase, são definidas as áreas técnicas e de negócio afetadas por esta demanda e os responsáveis em cada área são alocados no projeto. Na segunda fase, os responsáveis das áreas técnicas realizam uma análise preliminar do documento em conjunto com as áreas de negócio correspondentes. O documento é revisado até que o escopo seja compreendido por todos os envolvidos, podendo gerar várias versões do documento. Após a conclusão da versão final do documento de declaração de escopo, as áreas técnicas realizam uma estimativa de esforço para construção deste produto, com baixíssimo nível de detalhamento. Na terceira fase desta etapa ocorre a valoração da demanda, que é a conversão das horas estimadas em custo financeiro, e a aprovação do projeto por um comitê, que analisa informações como esforço, retorno do investimento e alinhamento à estratégia da empresa. Se o projeto for aprovado por este comitê, o valor estimado é alocado no orçamento e o início do projeto é formalizado. Caso contrário, ele é incluído em uma fila de projetos e é reavaliado nas próximas reuniões do comitê.

O estágio de análise, segunda etapa do processo, é iniciado após a aprovação formal do projeto. Nesta etapa, as áreas técnicas criam o documento de descrição funcional, onde o produto é detalhado com base no documento de declaração de escopo e são descritos os requisitos funcionais e não funcionais, casos de uso e regras de negócio. Em paralelo, as áreas de negócio definem os critérios que devem ser validados para garantir que o produto final esteja de acordo com o especificado, gerando os cenários de UAT (do inglês, *User Acceptance Testing*, ou Teste de Aceitação do Usuário). Nesta etapa, ocorrem reuniões entre as áreas técnicas e áreas de negócio e, caso seja necessário, são realizadas adequações no documento de declaração de escopo.

Após a conclusão do documento de declaração funcional, o estágio de *design* é iniciado. Nesta etapa, as áreas técnicas enviam o documento de declaração funcional para as suas equipes correspondentes na Fábrica de Software, para que o projeto seja analisado e orçado. São realizadas reuniões entre as áreas técnicas e as fábricas de *software* para que o escopo e solução técnica proposta sejam esclarecidos. Em seguida, a Fábrica de Software gera um documento de especificação funcional, que contém um maior detalhamento do documento de descrição funcional, um plano de entrega do projeto, um plano de testes do produto e uma proposta com o custo estimado para construção deste produto. Em seguida, é feita uma atualização do custo deste projeto no

orçamento aprovado na terceira etapa do estágio de iniciação. Se houver uma alteração significativa do custo do projeto em relação ao custo previsto anteriormente, ocorre uma nova avaliação do comitê. Caso contrário, o processo segue para seu próximo estágio.

O estágio seguinte é o de construção do produto, que utiliza as informações e artefatos gerados nos estágios anteriores para codificação do *software*. Esta etapa é realizada somente pela equipe da Fábrica de Software, sem ação direta da equipe da Área Técnica. Ao final deste estágio, são gerados os seguintes artefatos: uma especificação técnica contendo as informações detalhadas do produto, como *layout* de arquivos, detalhamento de interfaces, estrutura do banco de dados e pseudo-código do produto; um material de treinamento e manual de utilização do usuário, caso seja aplicável; um plano de teste com as evidências dos testes realizados; as documentações necessárias para entrega formal do produto; e, por fim, o código desenvolvido e testado.

Ao final do estágio de construção, é iniciado o estágio de aceitação. Nesta etapa, a equipe técnica é responsável por verificar a qualidade do produto desenvolvido pela Fábrica de Software, utilizando os cenários de UAT solicitados pela Área de Negócio no estágio de análise e realizando os testes que julguem ser necessários (caixa branca, caixa preta, regressão, desempenho etc.). Em seguida, é realizada uma reunião de homologação com a Área de Negócio, permitindo que o produto seja utilizado em um ambiente de testes para que a Área de Negócio formalize a sua aceitação.

Após obter a aceitação formal das áreas de negócio, o produto é encaminhado para implantação no ambiente de Produção, seguindo um fluxo interno para instalação da sua nova versão. Esta atividade é realizada pela equipe de Produção, que não participa ativamente das demais etapas deste processo. Com a nova versão do produto implantada, o processo de desenvolvimento é finalizado e a fase de manutenção do produto é iniciada.

3.2.3. CONTROLES UTILIZADOS

Todos os controles de atividades, projetos e entregas da equipe da Área Técnica eram realizados através de planilhas eletrônicas.

As atividades da área eram mantidas em uma planilha dividida em três seções: atividades em andamento, concluídas e canceladas. Cada seção possuía três divisões: investigações ou suportes, demandas sem análise iniciada e projetos. Os itens eram descritos em cada linha, onde era atribuído o responsável pela atividade e inserida uma

breve descrição, conforme exemplo da Figura 12. Os valores desta figura e das seguintes não serão apresentados por motivos de confidencialidade.

Prioridade	Sistema	Status	DEMANDAS SEM ANÁLISE INICIADA	Fluxo	Entrada	Responsável em TI	Previsão de conclusão	Detalhes	Atividades já realizadas	Atividades pendentes / Próximos passos
###	###	###	#########	###	###	###	###	###	###	###
###	###	###	#########	###	###	###	###	###	###	###

Prioridade	Sistema	Status	PROJETOS	Fluxo	Entrada	Responsável em TI	Previsão de conclusão	Detalhes	Atividades já realizadas	Atividades pendentes / Próximos passos
###	###	###	#########	###	###	###	###	###	###	###
###	###	###	#########	###	###	###	###	###	###	###

Prioridade	Sistema	Status	INVESTIGAÇÕES / SUPORTES	Fluxo	Entrada	Responsável em TI	Previsão de conclusão	Detalhes	Atividades já realizadas	Atividades pendentes / Próximos passos
###	###	###	#########	###	###	###	###	###	###	###
###	###	###	#########	###	###	###	###	###	###	###

Figura 12: Planilha de controle de atividades.

Como este controle era apresentado em reuniões de acompanhamento com as áreas usuárias, era necessário manter três versões deste documento, um para cada Área de Negócio.

O controle de defeitos identificados nos sistemas era realizado através de outra planilha, onde eram mantidas informações como data de ocorrência, breve descrição, responsável e *status*, conforme exemplo da Figura 13. Cada defeito corrigido pela fábrica gera uma ou mais entregas para a Área Técnica, que são controladas em outra planilha, descrita no próximo parágrafo.

Fluxo	Data Erro	Tipo de Erro	Tipo de Teste	Categoria	Descrição do Erro Encontrado	Responsável	Responsável Fábrica	Status Entrega	Resumo da Solução	O que é preciso testar?	Status Final do Erro
####	#####	######	######	#####	###############	#####	######	######	####	######	####
####	#####	######	######	#####	###############	#####	######	######	####	######	####

Figura 13: Planilha de controle de defeitos.

As entregas realizadas pela Fábrica de Software, sejam de correção de defeitos ou de implementação de novas funcionalidades, eram controladas através de uma planilha específica. Nesta planilha, eram descritas informações como sistema correspondente, fluxo da entrega, comentários e breve descrição, conforme exemplo da Figura 14.

Sistema	Rotina	Descrição	Entrega	Tipo	Fluxo	Dependências	Comentários	Responsável
###	###	########	###	##	##	###	###	#####
			###	##	##	###	###	####
###	###	########	###	##	##	###	###	#####

Figura 14: Planilha de controle de entregas

A manutenção de todos estes controles não é produtiva, pois apesar de todas as informações estarem diretamente relacionadas, cada uma precisa ser mantida em um

controle individual, dificultando a consulta de informações e tornando a atualização dos dados muito custosa.

Isto faz com que os controles não sejam atualizados com frequência, impossibilitando a equipe de definir uma melhor alocação de recursos por não ter uma visualização completa das atividades em execução e dificultando a gestão dos projetos da área. Em alguns casos, é necessário recorrer à busca de *e-mails* para se obter as informações de projetos para a gerência.

3.3. PERCEPÇÕES SOBRE A ORGANIZAÇÃO

Após caracterização das equipes, do processo de desenvolvimento e dos controles utilizados, os questionários descritos na Seção 3.2.1 foram utilizados para se obter a percepção das equipes sobre o processo de desenvolvimento de *software* utilizado na empresa e sobre os critérios relevantes para o sucesso em um projeto de desenvolvimento.

Em seguida, com os resultados apurados nos Questionários A e B, foram descritos os indicadores da necessidade de mudanças e os fatores favoráveis para a proposta de melhoria.

3.3.1. PERCEPÇÕES SOBRE O PROCESSO DE DESENVOLVIMENTO

As percepções dos entrevistados da Área de Negócio, da Área Técnica e da Fábrica de Software sobre o processo de desenvolvimento de *software* utilizado na empresa, foram obtidas através de três perguntas distintas, que foram respondidas pelos 2 integrantes da Área de Negócio, 4 integrantes da Área Técnica e 1 integrante da Fábrica de *Software*.

Em primeiro lugar, foi perguntado qual era a percepção do entrevistado sobre o sucesso dos projetos que participou em toda sua experiência profissional. O entrevistado deveria indicar qual percentual dos projetos que participou se enquadram nas seguintes opções:

- Concluídos com sucesso, ou seja, atenderam às expectativas de custo, prazo e qualidade;
- Concluídos com problemas, ou seja, não atenderam às expectativas de custo, prazo ou qualidade;
- Falharam, ou seja, foram concluídos e nunca utilizados, ou foram cancelados.

Segundo 4 entrevistados, 26 a 50% dos projetos que participaram foram concluídos com sucesso, enquanto 3 entrevistados indicaram que apenas 0 a 25% dos projetos que participaram se enquadram neste cenário. Com relação aos projetos concluídos com problemas, 3 entrevistados indicaram que 76 a 100% dos projetos que participaram se enquadram neste cenário. Outros 3 entrevistados indicaram que 51 a 75% dos projetos que participaram se enquadram neste cenário, enquanto 1 entrevistado indicou que 0 a 25% dos projetos que participou foram concluídos com problemas. Por fim, todos os entrevistados indicaram que apenas 0 a 25% dos projetos que participaram falharam.

A segunda pergunta possui o mesmo contexto e opções da pergunta anterior, porém se refere aos projetos realizados na organização atual.

Segundo 6 entrevistados, apenas 0 a 25% dos projetos que participaram na organização foram concluídos com sucesso, enquanto 1 entrevistado indicou que 26 a 50% dos projetos se enquadram neste cenário. Com relação aos projetos concluídos com problemas, 5 entrevistados indicaram que 76 a 100% dos projetos que participaram se enquadram neste cenário. Outros 2 entrevistados indicaram que 51 a 75% dos projetos que participaram se enquadram neste cenário. Por fim, todos os entrevistados indicaram que apenas 0 a 25% dos projetos que participaram falharam.

Estes números apresentam resultados compatíveis com as baixas taxas de sucesso descritas por HIBBS *et al.* (2009) e STANDISH GROUP (2010), e citadas na Seção 1.1 desta monografia.

Na terceira pergunta, foi perguntado aos entrevistados qual era sua percepção sobre quanto o processo de desenvolvimento de *software* utilizado na organização conseguia auxiliar no atendimento dos objetivos do projeto, com relação à: Prazo; Custo; Escopo; Satisfação do Cliente; Qualidade do Produto e Mudanças de Escopo. A escolha destes critérios foram baseadas nas áreas de conhecimento do PMBOK (PROJECT MANAGEMENT INSTITUTE, 2008). As opções de resposta eram: Auxilia muito; Auxilia pouco; Não auxilia / Indiferente; Atrapalha.

Os 7 entrevistados indicaram que o processo atrapalha a obtenção dos objetivos relacionados a escopo e mudanças de escopo. Com relação à satisfação do cliente, 1 entrevistado indicou que o processo auxilia pouco, 3 entrevistados indicaram que o processo não auxilia ou é indiferente e 3 entrevistados indicaram que o processo atrapalha. Estes e os demais resultados podem ser observados na Figura 15, a seguir.

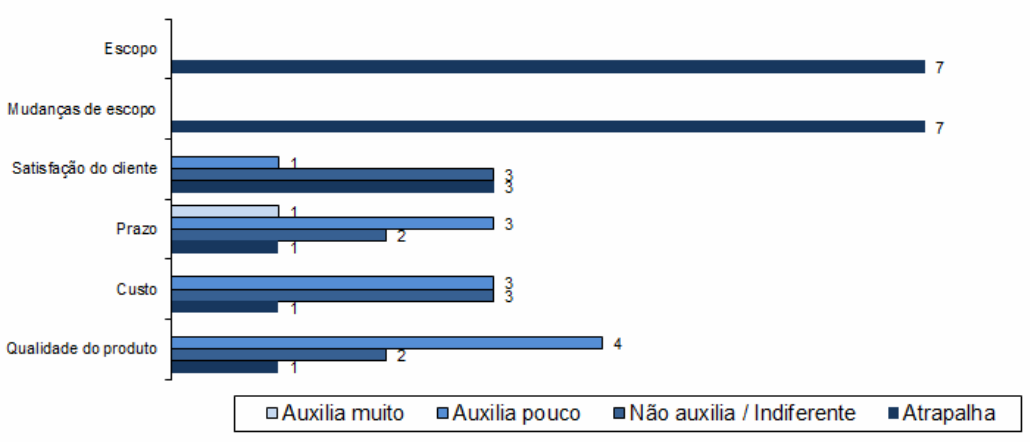

Figura 15: Percepção sobre a efetividade do processo de desenvolvimento na organização atual.

O resultado indica uma insatisfação de todos os entrevistados sobre o processo utilizado na organização, em relação ao escopo e às mudanças de escopo em um projeto de desenvolvimento. A satisfação do cliente também é impactada negativamente pelo processo, segundo 3 dos 7 entrevistados.

3.3.2. PERCEPÇÕES SOBRE OS CRITÉRIOS DE SUCESSO EM PROJETOS

Os Questionários A e B também tiveram como objetivo identificar a percepção dos integrantes das Área de Negócio, da Área Técnica e da Fábrica de *Software* sobre os critérios que são relevantes para o sucesso de um projeto de desenvolvimento.

Os critérios selecionados foram os mesmos apresentados na Figura 15, porém descritos de outra forma para facilitar o preenchimento dos questionários, conforme detalhes da Tabela 3.

Tabela 3: Critérios descritos nos Questionários A e B.

Critério	Descrição
Prazo	Entregar o produto no prazo previsto
Custo	Entregar o produto dentro do valor orçado
Escopo	Entregar o produto de acordo com o escopo definido
Satisfação do cliente	Entregar o produto que melhor atende às necessidades do cliente
Qualidade	Entregar o produto sem bugs identificados
Mudanças de escopo	Absorver mudanças de escopo durante o projeto

Os critérios foram apresentados como linhas e colunas de uma matriz, para que os entrevistados pudessem informar qual era a relevância do critério apresentado na linha com relação ao critério apresentado na coluna. Desta forma, foi possível comparar todos

os critérios um a um, seguindo a proposta do modelo AHP (*Analytic Hierarchy Process*)[2], apresentado em detalhes no Anexo II desta monografia.

Após comparação e aplicação do modelo AHP, os resultados foram analisados individualmente para cada área entrevistada.

Para a Área de Negócio, os critérios mais relevantes para o sucesso em um projeto são: entregar o produto sem *bugs* identificados, entregar o produto que melhor atende às necessidades do cliente e absorver mudanças de escopo durante o projeto, como pode ser visto na Figura 16. Estes critérios representam 86% de relevância e demonstram uma preocupação da área com a qualidade do produto e com o atendimento das necessidades do cliente e do negócio.

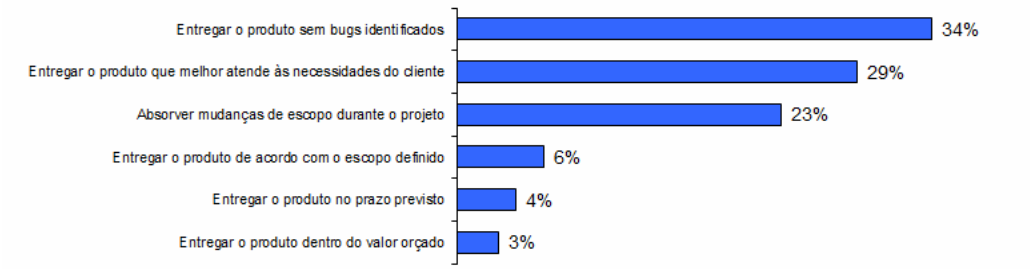

Figura 16: Relevância dos critérios de sucesso em projetos, de acordo com a Área de Negócio.

A Área Técnica considera como critérios mais relevantes para o sucesso de um projeto: entregar o produto que melhor atende às necessidades do cliente, entregar o produto de acordo com o escopo definido, absorver mudanças de escopo durante o projeto e entregar o produto sem *bugs* identificados, como pode ser visto na Figura 17. Estes critérios representam 85% de relevância e demonstram um alinhamento da Área Técnica com as expectativas da Área de Negócio ao manter o foco nas necessidades do cliente interno, na qualidade do produto e na gerência de escopo.

[2] O AHP foi desenvolvido na década de 1970 por Thomas L. Saaty e foi extensivamente estudado a partir dessa época. Atualmente é aplicado para a tomada de decisão em diversos cenários complexos, em que pessoas trabalham em conjunto para tomar decisões e onde percepções humanas, julgamentos e consequências possuem repercussão de longo prazo (BHUSHAN, 2004 *apud* VARGAS, 2010).

Figura 17: Relevância dos critérios de sucesso em projetos, de acordo com a Área Técnica.

Para a Fábrica de Software, os critérios mais relevantes para o sucesso do projeto são: entregar o produto dentro do valor orçado, de acordo com o escopo definido, sem *bugs* identificados e no prazo previsto, como pode ser visto na Figura 18. Estes critérios representam 84% de relevância e demonstram um foco da área no contrato comercial, ao buscar manter os critérios mínimos de qualidade, custo e prazo exigidos pela empresa contratante.

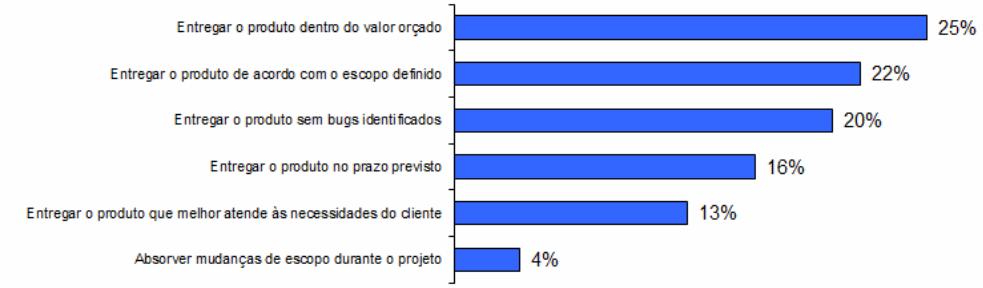

Figura 18: Relevância dos critérios de sucesso em projetos, de acordo com a Fábrica de Software.

3.3.3. INDICADORES DA NECESSIDADE DE MUDANÇAS

O processo descrito anteriormente, apesar de estruturar o desenvolvimento pelo sequenciamento de etapas e prescrever uma ampla documentação, não favorece a revisão das necessidades do negócio ao longo do processo, tornando qualquer mudança nos requisitos prejudicial ao projeto.

Com base na experiência do autor na participação nos projetos de desenvolvimento realizados pela Área Técnica selecionada, o histórico de projetos foi analisado e demonstrou problemas frequentes, decorrentes desta inflexibilidade do processo às mudanças e das dificuldades com o controle de atividades da equipe, descritos na Seção 3.2.3. Este cenário é reforçado pelo resultado dos Questionários A e B descritos na

Seção 3.3.1, que demonstram a percepção dos entrevistados sobre a baixa taxa de sucesso dos projetos de *software* realizados na empresa.

Os principais problemas identificados foram:

- **Dificuldade com o planejamento da equipe.**

Em função da baixa eficiência dos controles utilizados pela equipe, descritos na Seção 3.2.3, não é possível ter uma visão geral de todo o processo de desenvolvimento, dificultando as estimativas, as alocações da equipe e a análise de informações.

- **Os requisitos do produto não podem ser revisados durante o processo, sem que haja impacto em custo e prazo.**

Ao criar o documento de declaração de escopo, nem sempre as áreas de negócios conhecem todos os requisitos ou sabem detalhar por completo um requisito do produto. Eles possuem uma visão do produto, que deve ser detalhada para que uma versão aceitável do produto seja proposta. Este é um dos fatos que contribuem para falhas no desenvolvimento de software utilizando o modelo cascata (DEGRACE e STAHL, 1990, apud SUTHERLAND, 2004).

No processo de desenvolvimento da empresa, este fato possibilita a existência de diferenças significativas entre o primeiro orçamento, realizada no estágio de iniciação com base na declaração de escopo, e o orçamento atualizado no estágio de *design*, após detalhamento técnico do produto. Se houver uma diferença significativa entre os dois orçamentos realizados nas fases distintas, existe o risco de o projeto perder prioridade ou nem ser aprovado após a revisão do comitê, prejudicando o negócio.

- **O processo não favorece a resposta rápida às necessidades do mercado.**

A sequência de fases de detalhamentos de escopo, aprovações e priorizações tornam o processo lento, quando comparado à velocidade de mudança de informações no mercado. Este fato desperta nas áreas de negócio a iniciativa de criar processos alternativos utilizando planilhas, macros e procedimentos manuais, buscando atender as necessidades do negócio enquanto a área de desenvolvimento não consegue atendê-las a tempo. Isto, em um primeiro momento, pode oferecer como vantagem uma resposta rápida às mudanças necessárias no produto, por não depender do processo burocrático de desenvolvimento da empresa. No entanto, em contrapartida, podem também gerar processos, módulos ou até sistemas internos sem o suporte de TI, construídos sem a documentação e metodologias adequadas.

Os resultados dos Questionário A e B, descritos na Seção 3.3.1, também apontam que as mudanças de escopo são prejudicadas pelo processo de desenvolvimento utilizado na empresa, como pode ser visto na Figura 15.

- **Insatisfação da Área de Negócios com os projetos de software realizados na empresa.**

Os resultados dos Questionário A e B, descritos na Seção 3.3.1, apontam uma taxa de mais de 50% de insucesso nos projetos de desenvolvimento realizados na empresa. Além disso, os resultados apontam uma insatisfação não só da Área de Negócio, mas também da Área Técnica e da Fábrica de Software a respeito da baixa efetividade do processo com relação à gerência de escopo e satisfação do cliente.

3.3.4. FATORES FAVORÁVEIS ÀS MUDANÇAS

Durante a fase de diagnóstico e planejamento dos ciclos de melhoria, o ambiente da empresa sofreu mudanças que favorecem a iniciativa de melhoria de processos:

- **Atualização do modelo do documento de descrição funcional.**

O modelo do documento de descrição funcional, criado pela Área Técnica no estágio de análise do processo de desenvolvimento sofreu alterações para obrigar a explicitar os requisitos funcionais e não funcionais, as regras de negócio e os casos de uso, em vez de apenas listá-los na forma de texto livre, de acordo com a técnica desejada pelo Analista de Requisitos. Também se tornou obrigatório relacionar cada caso de uso com os requisitos funcionais e regras de negócio correspondentes.

Esta alteração favorece principalmente o desenvolvimento incremental, pois os casos de uso podem ser priorizados e divididos em lotes. Estes lotes podem ser entregues em várias versões do produto ao longo do projeto, aumentando a sensação de progresso da Área de Negócios e permitindo o aprendizado, evolução e replanejamento dos requisitos durante o desenvolvimento do projeto.

- **Mudança física das equipes**

As equipes citadas na Seção 3.2.1 eram alocadas em locais físicos distintos no Rio de Janeiro: a equipe da Área Técnica localizava-se no bairro de Botafogo; a equipe da Área de Negócio localizava-se no bairro da Barra da Tijuca; e a equipe da Fábrica de Software localizava-se no bairro de São Cristóvão. A partir da mudança física, todas as equipes estão localizadas no mesmo complexo, em São Cristóvão. Esta alteração

favorece a comunicação entre as equipes, facilitando a condução dos projetos e a experimentação de mudanças.

- **Interesse das equipes na experimentação de novas práticas**

Desde o início do projeto proposto nesta monografia, as equipes selecionadas na Área de Negócios, na Área Técnica e na Fábrica de Software demonstraram interesse em participar e experimentar novas práticas.

3.4. CONSIDERAÇÕES FINAIS

Neste capítulo foram descritos a iniciativa de melhoria e o cenário inicial da empresa onde o estudo de caso foi aplicado, apresentando a estrutura organizacional, a percepção das equipes sobre o processo de desenvolvimento de *software*, os indicadores da necessidade de mudanças.

Os indicadores da necessidade de mudanças mapeados foram relacionados à gestão de atividades da equipe, às mudanças de requisitos e à insatisfação da Área de Negócios com os projetos de *software* realizados na empresa. Estas informações serão utilizadas no próximo Capítulo, para o planejamento e a execução dos ciclos de melhoria.

4. EXECUÇÃO DOS CICLOS DE MELHORIA

Para atingir os objetivos propostos por esta monografia, a estratégia selecionada para a iniciativa de melhoria foi decompor a proposta em 4 ciclos, para minimizar a rejeição da equipe às mudanças na rotina de trabalho e proporcionar um período de adaptação e aprendizado entre cada ciclo.

No primeiro ciclo, as práticas ágeis identificadas na Seção 2.5 desta monografia foram analisadas, levando em consideração o perfil das equipes envolvidas, o momento atual da organização e o resultado dos questionários apresentados nas Seções 3.2 e 3.3. O objetivo deste ciclo foi identificar quais práticas ágeis seriam aplicáveis ao ambiente selecionado e descartar as práticas que não seriam aplicáveis, evitando problemas na implantação.

No segundo ciclo, foi dado foco na gestão das atividades da equipe da Área Técnica, visando melhorar a visualização das atividades, facilitar a alocação de recursos e permitir a identificação de gargalos. Neste ciclo houve uma mudança interna na gestão de atividades da equipe, refletindo positivamente no acompanhamento dos projetos da área.

O terceiro ciclo consistiu em propor melhorias na comunicação entre as equipes, buscando facilitar a troca de conhecimento e trazer transparência ao processo, além de reduzir o excesso de paralelismo de atividades (*multitasking*) através da limitação do trabalho em andamento, proposta pelo Kanban.

No quarto ciclo foram selecionados dois projetos piloto para execução, utilizando o ambiente com as melhorias realizadas nos ciclos anteriores, além da proposta de outras práticas relacionadas ao ambiente de trabalho e ao desenvolvimento do produto.

A Figura 19 apresenta o cronograma de execução dos ciclos de melhoria. O progresso exibido se refere apenas ao período de implantação. As etapas de monitoração e acompanhamento de cada ciclo foram iniciadas ao final de sua implantação e são contínuas.

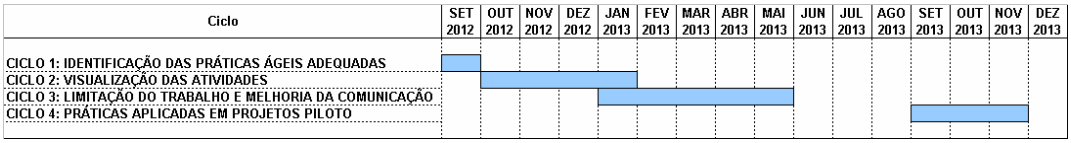

Figura 19: Cronograma de execução dos ciclos de melhoria.

Apesar do ciclo 3 ter sido concluído em Maio de 2013, o ciclo 4 só pode ser iniciado em Setembro de 2013 pois dependia da aprovação da gerência para início dos projetos

pilotos. Neste período, as práticas propostas nos ciclos 2 e 3 se mantinham em fase de melhoria contínua.

4.1. CICLO 1: IDENTIFICAÇÃO DAS PRÁTICAS ÁGEIS ADEQUADAS

O objetivo deste ciclo foi identificar quais práticas ágeis identificadas na Seção 2.5 seriam aplicáveis ao ambiente e à equipe, levando em consideração as informações e percepções obtidas nas Seções 3.2 e 3.3, e descartar as práticas que não seriam aplicáveis. O motivo de evitar a seleção de práticas que não se adequam ao ambiente é manter o macro-processo de desenvolvimento de *software*, que não pode ser alterado pois é adotado oficialmente pela empresa. O foco foi propor mudanças no micro-processo que não gerem impactos negativos no macro-processo.

As 19 práticas selecionadas na Seção 2.5 foram analisadas conforme sua característica: práticas relacionadas com os requisitos do produto; relacionadas com a fase de *design* do produto; relacionadas com a fase de construção do produto; relacionadas com a fase de testes do produto; relacionadas com a organização e o ambiente de trabalho e, por fim, relacionadas ao gerenciamento do projeto.

Dentre as práticas relacionadas com a fase de construção do produto, apenas as entregas curtas podem ser aplicadas pois não alteram o processo interno da Fábrica de S*oftware*. As demais práticas não podem ser aplicadas pois propõem mudanças no processo interno da Fábrica de *Software*, que é uma empresa terceirizada.

As práticas relacionadas com os requisitos e com a fase de *design* do produto podem ser aplicadas conforme detalhes a seguir:

- **Divisão em funcionalidades (*Features* / *Stories*):** O produto pode ser dividido utilizando os casos de uso propostos no documento de descrição funcional, prescrito no processo de desenvolvimento utilizado pela empresa.
- **Backlog do produto (*Product Backlog*):** Esta lista, proposta pelo Scrum, pode ser criada e mantida a partir dos casos de uso descritos no documento de descrição funcional.
- **Metáforas do sistema (*Metaphor*):** As metáforas podem ser utilizadas nas reuniões de entendimento do documento de declaração de escopo, na fase de iniciação do projeto. De acordo com a complexidade da funcionalidade a ser desenvolvida, estas metáforas também podem ser listadas no documento de detalhamento funcional.

A prática de automação de testes (*automated testing*) não é pertinente a ser selecionada para aplicação neste projeto, pois os testes realizados pela equipe da Área Técnica são basicamente funcionais e não há problemas identificados que justifiquem o esforço para esta automação. Os demais testes são realizados pela Fábrica de *Software*, cujo processo interno não pode ser alterado por motivos contratuais.

Todas as práticas mapeadas relacionadas com a organização e o ambiente de trabalho são aplicáveis nesta organização, de acordo com os detalhes a seguir:

- **Cliente presente (*On-site customer*):** Devido à mudança das equipes para uma mesma localização física, esta prática é pertinente de ser aplicada nos primeiros ciclos de melhoria, aproveitando esta oportunidade e a motivação da Área de Negócio.
- **Ritmo sustentável (*Sustainable Pace / 40 hour week*):** A cultura da empresa não incentiva as horas extras, tratando-as como exceção. Este posicionamento pode ser utilizado como motivação para aplicação desta prática.
- **Times multidisciplinares (*Whole team / multi-skill teams*):** Esta prática já era aplicada, de certa forma, na fase inicial dos projetos, através da definição dos pontos focais de cada equipe na participação do projeto. Neste caso, esta prática se mostra adequada porque poderia ser evoluída de acordo com as necessidades de melhorias de cada ciclo subsequente.

As práticas mapeadas relacionadas com o gerenciamento do projeto são aplicáveis nesta organização, com exceção do jogo de planejamento e das reuniões Scrum. A primeira não poderá ser aplicada neste momento, pois as estimativas são realizadas pela Fábrica de *Software*, enquanto a segunda só faz sentido se o Scrum for utilizado na íntegra, o que não é o caso deste projeto. As demais práticas poderão ser aplicadas de acordo com os detalhes a seguir:

- **Visibilidade do projeto (*Project visibility*):** Esta prática pode ser aplicada utilizando-se das mesmas oportunidades citadas na prática de cliente presente (*on-site customer*).
- **Retrospectivas (*Retrospective*):** As retrospectivas podem ser utilizadas em cada fase do projeto, ou apenas na fase de encerramento do projeto, obtendo *feedback* para cada equipe envolvida, a respeito do processo utilizado e do produto desenvolvido.

- **Reuniões de pé (*Stand-up meetings*):** Complementando as práticas de visibilidade do projeto e de cliente presente, esta prática pode ser utilizada na melhoria da comunicação das equipes visando tornar as reuniões mais objetivas.

As práticas selecionadas após esta análise inicial foram utilizadas no decorrer dos próximos ciclos de melhoria, levando em consideração seus objetivos específicos, conforme demonstrado na Tabela 4.

Tabela 4: Relação das práticas ágeis abordadas nos artigos revisados.

Etapa	Objetivos	Práticas selecionadas	Justificativas
CICLO 2: VISUALIZAÇÃO DAS ATIVIDADES	- Melhorar na visualização das atividades da equipe; - Facilitar a alocação de recursos; - Permitir a identificação de gargalos no fluxo.	- Visibilidade do projeto;	A equipe da Área Técnica possui dificuldade no planejamento e visualização das atividades da equipe, em função da baixa eficiência de seus controles. Isto gera impacto negativo na execução dos projetos.
CICLO 3: LIMITAÇÃO DO TRABALHO E MELHORIA DA COMUNICAÇÃO	- Eliminação de gargalos no fluxo;	- Ritmo sustentável; - Cliente presente; - Reuniões de pé; - Retrospectivas - Visibilidade do projeto.	Melhorar a comunicação da equipe para antecipar problemas e compartilhar conhecimento. Além de ajustar a carga de trabalho para tornar o ambiente mais saudável, evitando horas extras e atrasos por excesso de paralelismo de atividades.
CICLO 4: PRÁTICAS APLICADAS EM PROJETOS PILOTO	- Aplicar todas as práticas na execução de um projeto, e não só em seu monitoramento; - Aumentar a transparência no processo; - Melhorar a satisfação da Área de Negócios; - Reduzir os impactos das mudanças no processo.	- Ritmo sustentável; - Cliente presente; - Reuniões de pé; - Retrospectivas - Visibilidade do projeto; - Divisão em funcionalidades.	Atuar na melhoria da satisfação e sensação de progresso da Área de Negócios, além de adaptar o processo para receber melhor qualquer tipo de mudança durante a fase de desenvolvimento.

4.2. CICLO 2: VISUALIZAÇÃO DAS ATIVIDADES

O segundo ciclo de melhoria teve como foco a dificuldade de planejamento e visualização de atividades que a equipe da Área Técnica possui, em função da baixa eficiência dos controles descritos na Seção 3.2.3. A opção de atuar nestes problemas no primeiro ciclo que efetivamente propõe mudanças no ambiente foi selecionada porque a melhoria dos controles internos afetaria diretamente a gestão dos projetos de forma positiva.

A prática selecionada para aplicação foi a de visibilidade do projeto (*Project visibility*), proposta pelo Kanban, que também busca atender ao princípio da eliminação de desperdícios proposto pelo desenvolvimento enxuto de *software*. Para minimizar possíveis rejeições às mudanças propostas, não foi citado nome de nenhuma metodologia ou ferramenta, se referindo apenas como uma nova maneira de controlar as atividades.

4.2.1. QUADRO DE ATIVIDADES

Como ferramenta de trabalho, foi selecionado o quadro de atividades do Kanban para gerenciar o portfólio de projetos da Área Técnica, onde cada atividade do quadro

representaria um projeto e todas as etapas do processo de desenvolvimento seriam representadas no quadro.

Em função de limitações de espaço para a utilização de um quadro físico, foi utilizada uma ferramenta *web* gratuita (KANBANTOOL, 2013) onde um quadro virtual é mantido e compartilhado. A utilização do quadro virtual traz vantagens para a aplicação, como a utilização remota do controle das atividades, o compartilhamento destas atividades com a área usuária em tempo real, a possibilidade de manter informações associadas a cada cartão do quadro, entre outras.

O quadro virtual foi criado contendo todas as etapas do processo de desenvolvimento da empresa, descrito na Seção 3.2.2, e suas atividades foram criadas com base nas informações da planilha de controle de atividades, descrita na Seção 3.2.3. O resultado inicial foi o quadro exibido na Figura 20. As informações não estão visíveis por questões de confidencialidade, mas é possível notar o primeiro gargalo identificado, destacado em vermelho: o desenvolvimento dos produtos era concluído, mas eles ficavam represados nesta etapa aguardando disponibilidade da equipe da Área Técnica para iniciar os testes.

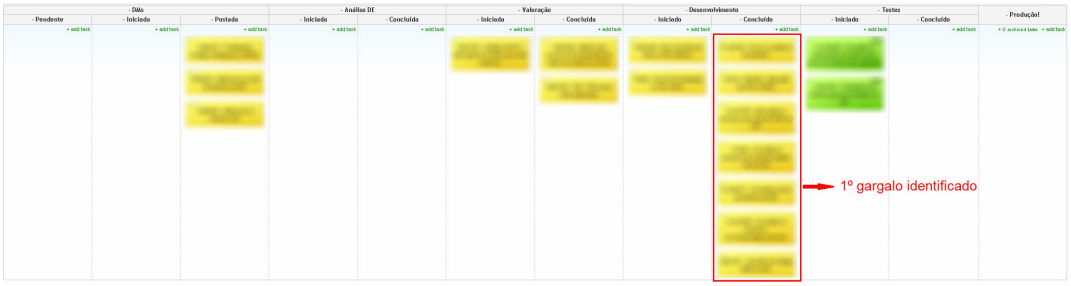

Figura 20: Primeira versão do quadro virtual de atividades.

A partir deste momento, o quadro virtual passou a ser atualizado no dia a dia da equipe, de acordo com a evolução das atividades. Porém, em um primeiro momento de adaptação da equipe, foram mantidos os dois controles em paralelo: o quadro virtual e a planilha de controle de atividades.

A manutenção desses dois controles em paralelo gerou, em um primeiro momento, um esforço maior do que o aplicado na situação anterior a esta proposta. Porém, no decorrer de 2 semanas de trabalho, a equipe da Área Técnica foi migrando aos poucos para a utilização apenas do quadro virtual após perceber que o controle das atividades era mais rápido e intuitivo.

As Figuras 21 e 22 demonstram, respectivamente, o recurso de comentários e histórico do quadro virtual, sendo utilizado para facilitar a comunicação e o armazenamento de informações em cada projeto. Cada projeto contido no quadro de atividades possui suas informações armazenadas individualmente.

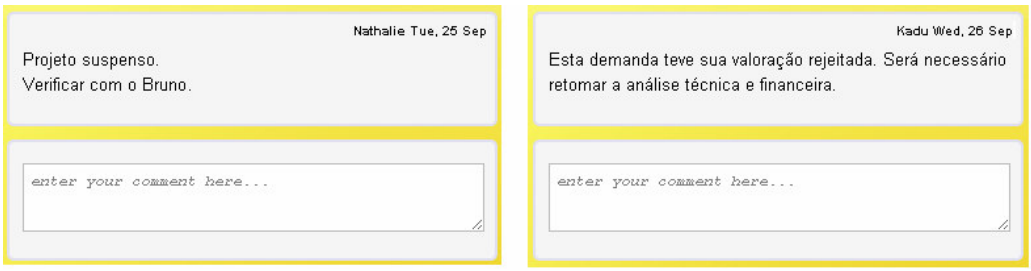

Figura 21: Registro de comentários em cada projeto do quadro

Figura 22: Consolidação de informações em cada projeto do quadro

Em 3 semanas de trabalho, o quadro Kanban substituiu completamente o controle feito através de planilhas, facilitando a visualização completa do fluxo de trabalho e permitindo a equipe a identificar e atuar no primeiro gargalo identificado.

4.2.2. GARGALOS IDENTIFICADOS

O primeiro gargalo foi identificado na etapa de testes e homologação, pois a etapa de desenvolvimento era concluída pela Fábrica de Software e a equipe da Área Técnica não tinha capacidade para testar todas as entregas realizadas, represando o fluxo neste ponto. A ação realizada pelo coordenador foi mobilizar toda a equipe para priorizar a conclusão dos testes em função das demais atividades.

Após esta ação, o quadro Kanban foi sendo adaptado às necessidades diárias e evoluindo conforme Figuras 23 e 24, a seguir. Nestas figuras, algumas informações foram ocultas por motivos de confidencialidade.

O quadro da Figura 23 apresenta uma redução no gargalo na etapa de testes e homologação em relação ao quadro da Figura 20, demonstrando que a mobilização realizada pelo coordenador da equipe para reduzir a quantidade de testes pendentes estava sendo efetiva.

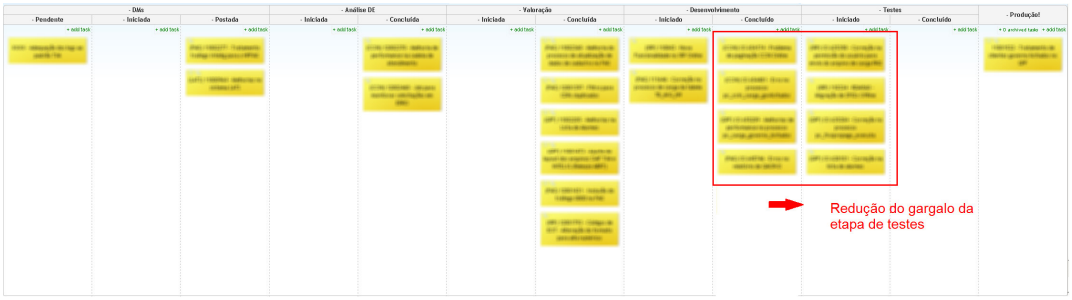

Figura 23: Visão do quadro kanban após 1 mês de utilização.

A Figura 24 apresenta uma visão do quadro de atividades obtida 1 mês após a visão da Figura 23. Foram destacados três gargalos nesta imagem:

- O primeiro, sinalizado pelo número 1, representa as atividades de análise técnica que estavam sendo realizadas pela Área Técnica. Nota-se uma grande quantidade de atividades paralelas, gerando uma grande quantidade de atividades represadas nesta etapa;
- O gargalo sinalizado pelo número 2 representa as atividades que estavam em andamento na Fábrica de *Software*, demonstrando outra etapa com alto grau de paralelismo e muitas atividades represadas;
- O gargalo sinalizado pelo número 3 representa as atividades de testes que estavam sendo realizadas pela Área Técnica. Nota-se uma grande quantidade de atividades em paralelo, inclusive com as atividades do gargalo 1.

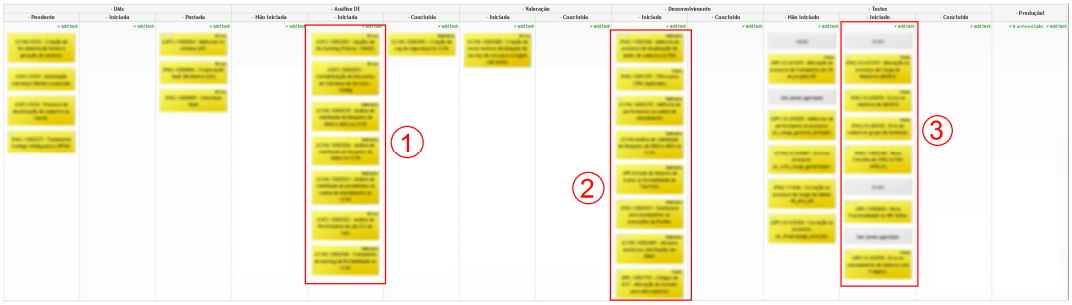

Figura 24: Visão do quadro kanban após 2 meses de utilização.

Com base nas imagens e ações citadas anteriormente, é possível observar que a utilização do quadro de atividades facilitou a identificação e resolução de gargalos no fluxo de atividades.

4.2.3. BENEFÍCIOS PERCEBIDOS

Os benefícios percebidos após aplicação desta prática foram:

- Melhoria na visualização dos projetos e atividades da área, em suas respectivas fases;
- Melhoria na alocação da equipe, ao permitir identificar facilmente as atividades sob responsabilidade de cada integrante;
- Identificação visual de gargalos no fluxo;
- Facilitação e armazenamento da comunicação em cada projeto especifico, através dos comentários exibidos no exemplo da Figura 24;
- Eliminação dos controles realizados anteriormente através de planilhas, citados na Seção 3.1.2, através da centralização das informações nas descrições de cada projeto, conforme exemplo da Figura 25.

As dificuldades encontradas durante esta aplicação foram:

- A resistência inicial da equipe às mudanças em sua rotina, que foi minimizada ao evitar citar o nome dos métodos e práticas utilizadas e incentivar a equipe a participar ativamente das propostas de melhoria;
- O período de manutenção de dois controles em paralelo, enquanto o novo controle não era utilizado por toda a equipe.

O sucesso na aplicação desta prática tornou a equipe mais receptiva a novas mudanças e preparou o ambiente para a aplicação das práticas de cliente presente, buscando uma maior transparência do processo, e ritmo sustentável, buscando uma maior qualidade no trabalho e satisfação das equipes.

Os passos seguintes para melhoria do processo foram a melhoria da comunicação das equipes e redução do excessivo paralelismo de atividades (*multitasking*) de cada membro da equipe, apresentados a seguir.

4.3. CICLO 3: LIMITAÇÃO DO TRABALHO E MELHORIA DA COMUNICAÇÃO

Com os benefícios gerados após implantação do ciclo 2, a equipe da Área Técnica identificou novos problemas que motivaram a implantação deste novo ciclo de melhoria. Este ciclo consistiu em propor melhorias na comunicação entre as equipes,

buscando facilitar a troca de conhecimento e trazer transparência ao processo, além de reduzir o excesso de paralelismo de atividades (*multitasking*) através da limitação do trabalho em andamento, proposto pelo Kanban.

Após a melhoria da visualização das atividades, foi possível identificar gargalos nas etapas de testes, conforme quadros das Figuras 20 e 24, e na etapa de análise técnica, conforme o quadro da Figura 24.

4.3.1. ANÁLISE DE GARGALOS E DEFINIÇÃO DO TRABALHO EM ANDAMENTO

Os gargalos citados anteriormente foram analisados com mais detalhes pela equipe da Área Técnica, que identificou que sua causa era a elevada quantidade de atividades paralelas, relacionadas a assuntos distintos. Esta causa é um problema comum no mercado de trabalho e já foi tema de um estudo publicado na *Harvard Business Review*, que concluiu que projetos são finalizados mais rapidamente se a organização executar poucos projetos ao mesmo tempo (ADLER, 1996, *apud* MAHER, 2011).

O recurso proposto para buscar resolver este problema foi a limitação do trabalho em andamento (ou WIP, do inglês, w*ork in progress*), proposto pelo Kanban, visando a prática de ritmo sustentável (*sustainable pace/40 hour week)*. Desta forma, se não há capacidade para testar muitas entregas ao mesmo tempo, não é indicado alocar todas estas horas de desenvolvimento de uma só vez. Isto gera um desperdício financeiro, pois mesmo tendo contratado um grande volume de entregas, os projetos não estavam sendo implantados em Produção com a mesma velocidade.

Como a capacidade de entrega da equipe (ou velocidade do time, segundo o Scrum) não era conhecida, o primeiro WIP utilizado para a etapa de testes foi de 7, seguindo a fórmula 2n-1 (onde *n* corresponde ao número de membros da equipe, ou seja, 4; e *-1*, foi utilizado para encorajar a colaboração e ter uma margem de segurança). Desta forma, cada membro seria responsável por até duas tarefas em paralelo, seguindo a indicação do estudo de CLARK e WHEELWRIGHT (1993, *apud* MAHER, 2011) sobre a produtividade de acordo com a quantidade de atividades paralelas, demonstrado na Figura 25.

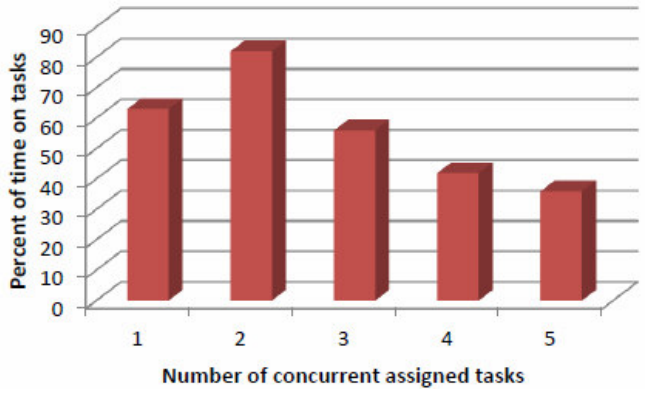

Figura 25: Produtividade de acordo com a quantidade de atividades paralelas.
(CLARK e WHEELWRIGHT, 1993, *apud* MAHER, 2011)

Após duas semanas de ajustes, a equipe da Área Técnica decidiu aumentar este limite para 3 tarefas por membro, já que algumas atividades dependiam da ação de outras equipes e, com o limite anterior, alguns membros acabavam ficando ociosos.

4.3.2. MELHORIA DA COMUNICAÇÃO ENTRE A EQUIPE DA ÁREA TÉCNICA

A segunda proposta deste ciclo foi a melhoria da comunicação interna da equipe da Área Técnica. O foco nesta melhoria teve como objetivo facilitar o compartilhamento de informações entre os membros da equipe, antecipando possíveis problemas e favorecendo a troca de conhecimento. Desta forma, seria possível buscar o incentivo ao aprendizado, proposta pelo desenvolvimento enxuto de software, além dos princípios de troca de informações e trabalho em conjunto, base do Manifesto Ágil.

Para melhorar a comunicação interna da equipe, foram selecionadas as práticas de reuniões diárias, reuniões em pé, retrospectivas e visibilidade do projeto. Todas estas práticas foram consolidadas em uma reunião diária, onde seria realizada uma retrospectiva do dia anterior.

Esta reunião seria realizada no início do dia e teria duração máxima de 10 minutos, com todos os participantes de pé para forçar a objetividade e foco no tema. Cada participante iria falar sobre o progresso das suas atividades no dia anterior, sobre o que pretendia realizar no decorrer do dia e quais eram os problemas identificados que poderiam impactar seu planejamento. Desta forma, todos os membros estariam cientes das atividades em andamento e poderiam compartilhar soluções para os problemas identificados.

Após a primeira semana de aplicação desta reunião, a equipe decidiu realizá-la semanalmente, pois as atividades da área não eram tão dinâmicas ao ponto de terem muitas atualizações a compartilhar diariamente. A comunicação continuou sendo realizada diariamente entre os membros para compartilhar problemas e soluções, porém fora de uma reunião.

A partir da terceira reunião semanal, a equipe estava com suas atividades organizadas (proposta de melhoria do ciclo 2), sua capacidade de trabalho estava sendo monitorada e ajustada (primeira proposta deste ciclo, com o ajuste do WIP) e a comunicação interna estava sendo facilitada (segunda proposta deste ciclo, com as reuniões semanais). O ambiente, portanto, estava pronto para aplicação do próximo passo: a melhoria da comunicação com as demais equipes.

4.3.3. MELHORIA DA COMUNICAÇÃO COM AS EQUIPES DA ÁREA DE NEGÓCIO E FÁBRICA DE SOFTWARE

Seguindo o mesmo modelo da reunião de acompanhamento da equipe, foram propostas mais duas reuniões semanais, uma com a Área de Negócio e outra com a Fábrica de *Software*. Estas não seriam realizadas de pé, mas teriam duração estipulada de 30 minutos para evitar desperdícios.

Com relação aos usuários, cada ponto focal da Área Técnica seria responsável pela reunião com sua respectiva Área de Negócio. O quadro virtual de atividades foi compartilhado com as áreas de negócio, permitindo que os usuários pudessem consultar as informações e comentários dos projetos em tempo real através da aplicação *web*, porém sem alterar nenhuma informação. A reunião semanal seria apenas uma oportunidade para revisar as prioridades e tratar problemas pontuais com um ponto focal da Área de Negócios, reforçando a transparência do processo.

O acompanhamento com a Fábrica de *Software* seria realizado de forma similar ao da Área de Negócios. Porém, como todos os pontos focais da Área Técnica são atendidos por uma mesma equipe da Fábrica de *Software*, os participantes desta reunião semanal seriam os membros da Área Técnica e o coordenador da Fábrica de Software. Nesta reunião, as atividades em andamento e o planejamento das próximas atividades seriam revisadas semanalmente.

O principal ponto positivo do envolvimento das demais áreas com as práticas propostas neste ciclo foi o interesse Área de Negócios em utilizar o Kanban para gestão

de suas atividades não relacionadas com TI. Este interesse favoreceu a troca de conhecimento entre as áreas e pode reforçar a importância da melhoria de processos na organização.

4.3.4. BENEFÍCIOS E DIFICULDADES PERCEBIDOS

Os demais benefícios percebidos com a aplicação destas práticas foram:

- Conhecimento da velocidade da equipe em cada etapa do processo de desenvolvimento;
- Redução da ociosidade e da sobrecarga de trabalho através do ajuste e monitoramento constante do WIP;
- Antecipação de atrasos e resolução de problemas ao favorecer a comunicação entre todos os membros da equipe;
- Compartilhamento de informações entre membros da equipe e outras áreas da empresa;
- Aumento da sensação de progresso da Área de Negócios ao acompanhar mais de perto as atividades da Área Técnica;
- Aumento da confiança da Área de Negócios na Área Técnica ao permitir a transparência dos processos e atividades.
- Revisão semanal de prioridades do negócio e atividades relacionadas na Área Técnica, garantindo que os esforços da área de TI estão alinhados com a estratégia da Área de Negócios.

As dificuldades encontradas durante este ciclo foram:

- Entendimento e correta aplicação do conceito de WIP pela equipe, que foi solucionado com o compartilhamento de conteúdos de apoio (sites, blogs, apresentações e artigos na internet);
- Resistência inicial de um membro da equipe com o acompanhamento de atividades sob sua responsabilidade. Esta resistência foi tratada e solucionada pelo coordenador da área;
- Período de adaptação da Área de Negócio aos novos conceitos apresentados. Para reduzir este período, foram realizadas palestras para compartilhar o conhecimento das práticas utilizadas, apresentando um conteúdo adaptado à realidade da Área de Negócio, utilizando a prática de metáforas;

- Limitação do plano gratuito da ferramenta utilizada (quadro virtual kanban) para utilização plena dos recursos necessários. Estas limitações foram contornadas com adaptações do uso de informações nos projetos.

Os próximos passos mapeados para a melhoria contínua do processo foram a aplicação de práticas relacionadas ao desenvolvimento do produto e ao gerenciamento do projeto em um ou mais projetos piloto, conforme detalhes na próxima seção.

4.4. CICLO 4: PRÁTICAS APLICADAS EM PROJETOS PILOTO

Apesar das melhorias propostas nos ciclos 2 e 3 terem sido aplicadas na prática, elas foram executadas ao longo de 8 meses, envolvendo a gestão de projetos já iniciados. No entanto, ainda era necessário executar um ou mais projetos por completo, utilizando todas a práticas propostas em conjunto para avaliar seu impacto e adequabilidade na organização.

Diante dos resultados positivos dos ciclos anteriores, a gerência responsável pela Área Técnica autorizou a condução de dois pequenos projetos, para dar sequência às ações de melhoria propostas. A única restrição foi que o custo envolvido não fosse superior a 20% do valor já aprovado.

O primeiro projeto, que será denominado "Projeto A" por motivos de confidencialidade, consistia em adicionar novas funcionalidades a um módulo já existente, com escopo considerado de baixa complexidade e estimado em 9 dias de documentação, desenvolvimento, testes e entrega da fábrica. O segundo projeto, que será denominado "Projeto B", tratava da criação de um novo módulo, com funcionalidades consideradas de baixa a média complexidade e esforço estimado em 14 dias de documentação, desenvolvimento, testes e entrega da fábrica.

Para melhor aproveitar a oportunidade de testar a aplicação de novas práticas em projetos piloto, foram adotadas estratégias distintas para cada projeto. Estas estratégias serão avaliadas ao fim dos projetos e será possível verificar quais foram mais efetivas. O detalhe de cada projeto será descrito individualmente, a seguir:

4.4.1. EXECUÇÃO DO PROJETO A

Conforme descrito anteriormente, o projeto denominado "Projeto A" consistia em adicionar novas funcionalidades a um módulo já existente no sistema. A estimativa

inicial, orçada e aprovada antes da proposta de aplicação deste ciclo está detalhada na Tabela 5, a seguir:

Tabela 5: Estimativa inicial para o Projeto A.

Projeto A	
Fase	**Dias**
Documentação	1 dia
Construção	5 dias
Testes	2 dias
Entrega formal	1 dia
Total	**9 dias**

Na fase de documentação seriam geradas as especificações funcionais e técnicas, e o plano de testes. O esforço estimado foi de um dia, por contar com uma baixa complexidade das funcionalidades alteradas. Na fase de entrega formal, são realizados os procedimentos internos para cadastro da demanda e criação dos pacotes de instalação. O esforço desta fase é sempre calculado com um dia por módulo, conforme contrato definido com a empresa.

Como o escopo envolvia apenas um módulo, com funcionalidades simples e poucos dias de construção, a prática de entregas curtas não foi aplicável a este projeto, pois o projeto em si já era curto. No entanto, a prática explorada foi a de cliente presente, adaptada para visitas diárias do usuário da Área de Negócios ao ambiente de desenvolvimento, com o objetivo de manter a transparência no desenvolvimento e manter a sensação de progresso que a prática de entregas curtas pretende gerar ao usuário.

Para que a experiência fosse positiva para a Área de Negócios, mas não prejudicasse as atividades da Fábrica de *Software*, foi negociado com a fábrica a alocação de uma hora por dia durante os 8 dias iniciais do desenvolvimento, totalizando um dia útil no final do projeto. Neste período, o usuário acompanharia as fases de documentação, construção e testes.

A dinâmica das visitas seria similar à das reuniões de acompanhamento da Área Técnica, com duração máxima de 15 minutos, onde seriam abordados o progresso do dia anterior, os problemas identificados e o planejamento para o dia corrente. Os 45 minutos de diferença entre o tempo efetivamente gasto na visita e a hora negociada, seriam utilizados como uma compensação pela rotina da fábrica ter sido alterada.

Após esta negociação, a estimativa foi alterada conforme Tabela 6 a seguir:

Tabela 6: Estimativa alterada para o Projeto A.

Projeto A	
Fase	**Esforço**
Documentação	1 dia
Construção	5 dias
Testes	2 dias
Visitas do usuário	1 dia
Entrega formal	1 dia
Total	**10 dias**

Apesar de esta proposta representar um aumento de um dia no esforço inicial, cerca de 11%, esta diferença se manteve inferior ao limite de 20% estipulado pelo gerente da Área Técnica.

4.4.2. EXECUÇÃO DO PROJETO B

O segundo projeto, denominado "Projeto B", tratava da criação de um novo módulo de um sistema já existente. A estimativa inicial, orçada e aprovada antes da proposta de aplicação deste ciclo está detalhada na Tabela 7 a seguir:

Tabela 7: Estimativa inicial para o Projeto B.

Projeto B	
Fase	**Dias**
Documentação	1 dia
Construção	9 dias
Testes	3 dias
Entrega formal	1 dia
Total	**14 dias**

O esforço estimado para a fase de documentação foi de um dia, por contar com funcionalidades de baixo a média complexidade. A fase de entrega formal também foi estimada em um dia, pois apenas um módulo seria previsto para entrega.

Durante a criação do documento de descrição funcional pela Área Técnica, os requisitos foram detalhados e o módulo que era inicialmente unitário foi dividido em três módulos mais simples, de complexidade similar. Apesar da fase de construção ser estimada em poucos dias, esta divisão dos módulos foi uma oportunidade para experimentar a prática de entregas incrementais, já que não foi possível utilizá-la no Projeto A.

A negociação com a Fábrica de *Software* contou com a divisão do produto nestes três módulos e em entregas parciais durante o desenvolvimento. A documentação ainda seria descrita da mesma forma inicial, mantendo-se em um dia de esforço.

As entregas também foram negociadas, pois como o produto agora possuía três módulos, por contrato, a fábrica poderia cobrar três dias para a fase de entrega. A proposta feita foi a realização das duas primeiras entregas de maneira informal, sem conter os padrões para aplicação dos módulos no ambiente de Produção, e a da última entrega de maneira formal, contendo todos os padrões para aplicação dos três módulos no ambiente de Produção. Desta forma, as entregas intermediárias não foram cobradas.

O planejamento final deste projeto foi alterado segundo a Tabela 8 a seguir:

Tabela 8: Estimativa alterada para o Projeto B.

Projeto B	
Fase	**Esforço**
Documentação	1 dia
1ª entrega	
Construção	3 dias
Testes	1 dia
Entrega informal	-
2ª entrega	
Construção	3 dias
Testes	1 dia
Entrega informal	-
3ª entrega	
Construção	2 dias
Testes	1 dia
Entrega formal	1 dia
Total	**13 dias**

Como o módulo previsto inicialmente foi dividido em módulos mais simples que possuíam características similares, houve um ganho de 1 dia na fase de construção da 3ª entrega em função da reutilização de código. O esforço final foi de um dia a menos que o inicial, compensando o aumento de um dia no esforço do Projeto A, após a proposta realizada.

Na proposta inicial, a Área de Negócios iria ter seu primeiro contato com o produto desenvolvido no mínimo 14 dias depois do início do desenvolvimento, após a entrega oficial, quando seriam iniciados os testes de aceitação com a Área Técnica. Como o projeto já estaria concluído na Fábrica de *Software*, nenhum ajuste poderia ser feito em tempo de desenvolvimento sem custos de mudança.

Com esta nova proposta, a primeira versão do produto seria entregue 5 dias após o início do projeto e o usuário já poderia ter contato com o produto no ambiente de testes, enquanto a Área Técnica estivesse realizando os testes de aceitação. A construção da segunda versão seria iniciada no 6° dia de projeto, e o usuário poderia rever alguma característica do produto antes do início da terceira versão, no 9° dia de projeto, ainda em tempo de desenvolvimento.

4.4.3. RESULTADOS DA EXECUÇÃO DOS PROJETOS PILOTO

As execuções dos projetos A e B foram realizadas sem maiores problemas, conforme o cronograma previsto. Os detalhes da execução de cada projeto serão apresentados a seguir.

As visitas realizadas durante o Projeto A foram realizadas de acordo com os limites estipulados. A primeira visita foi a mais longa e utilizou os 15 minutos reservados, pois a Fábrica de *Software* realizou uma breve apresentação de seu ambiente (sistema de controle de versão, ambientes de desenvolvimento e testes etc.) ao usuário da Área de Negócios, que nunca havia visitado um ambiente com esta característica. As demais visitas duraram pouco menos de 10 minutos e foram produtivas, pois algumas sugestões de *layout* foram validadas com o usuário em tempo de desenvolvimento.

Um ponto negativo para o projeto, mas positivo para a experiência foi o atraso de um dia na etapa de testes, em função da indisponibilidade do ambiente de testes. Neste caso, o usuário teve contato com as dificuldades que podem surgir durante o desenvolvimento e que em certos casos não são percebidas na entrega final, pois geralmente o atraso é absorvido pela fábrica através das horas extras da equipe.

Para o Projeto B, as entregas parciais também trouxeram benefícios durante a execução do projeto. Após a entrega do primeiro módulo pela fábrica, a Área Técnica disponibilizou esta versão do produto no seu ambiente de testes e compartilhou com o usuário da Área de Negócios, que já conseguiu utilizar o novo produto no 5° dia de projeto, validando as características deste módulo que foram solicitadas no documento de declaração de escopo.

Após a entrega do segundo módulo e antes do início da construção do terceiro módulo, foi realizada uma reunião de revisão de escopo entre o usuário da Área de Negócios, o ponto focal da Área Técnica e o responsável da Fábrica de Software. A alteração de escopo foi pequena, adequando uma consulta que foi entregue no segundo

módulo e que foi corrigida na entrega do terceiro módulo, ainda em tempo de desenvolvimento.

Esta alteração de escopo, apesar de simples, já demonstrou um ponto positivo desta proposta, pois se o desenvolvimento fosse realizado em apenas um módulo, a identificação desta adequação seria feita após a conclusão do desenvolvimento na fábrica, o que iria gerar custos para uma nova entrega corrigida.

4.5. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os ciclos de melhoria propostos para aplicação na organização. O primeiro ciclo permitiu selecionar as práticas ágeis que foram aplicáveis ao ambiente de trabalho, utilizadas nos demais ciclos. O segundo ciclo manteve o foco na melhoria da gestão das atividades, permitindo que os controles da equipe fossem otimizados e que fosse possível identificar os gargalos no fluxo de atividades. O terceiro ciclo apresentou novas melhorias na gestão de atividades através do ajuste do trabalho em andamento, reduzindo o excesso de paralelismo de atividades, além de trazer melhorias na comunicação entre as equipes. O quarto e último ciclo utilizou as práticas já propostas nos ciclos anteriores para gestão de dois projetos pilotos, além da proposta de práticas com foco na transparência e maior contato com a Área de Negócios.

Os resultados percebidos na aplicação destas propostas de melhoria foram positivas, porém, deve-se executar uma retrospectiva com as equipes envolvidas para se obter os pontos positivos e negativos de cada ciclo. Este será o objetivo do próximo capítulo.

5. AVALIAÇÃO DA PROPOSTA

Ao final da aplicação dos 4 ciclos de melhoria, descritos no capítulo anterior, foi realizada uma análise crítica dos resultados de cada ciclo e dos efeitos da aplicação das práticas ágeis propostas por eles. O objetivo desta análise é validar se as práticas planejadas para cada ciclo foram aplicadas e se atingiram seus objetivos, conforme previsto na Tabela 4, apresentada no final da Seção 4.1.

Além disso, foram realizadas reuniões de retrospectiva com as três equipes participantes desta iniciativa de melhoria. O objetivo destas retrospectivas foi obter a percepção dos envolvidos sobre os resultados de cada ciclo de melhoria, identificando os pontos positivos e negativos e, por fim, verificar se o esforço realizado conquistou apoio das gerências envolvidas para continuação das propostas de melhoria no processo.

5.1. ANÁLISE CRÍTICA DOS CICLOS DE MELHORIA

A Tabela 4, apresentada no final da Seção 4.1, apresentou o planejamento dos ciclos de melhoria, com as práticas previstas e o objetivo de cada ciclo proposto.

Após a aplicação de todos os ciclos de melhoria previstos, este planejamento foi analisado para validar se todas as práticas planejadas foram aplicadas e se os objetivos propostos foram cumpridos, gerando a Tabela 9, a seguir:

Tabela 9: Análise das práticas aplicadas em cada ciclo de melhoria.

Etapa	Objetivos	Práticas selecionadas	Discussão
CICLO 2: VISUALIZAÇÃO DAS ATIVIDADES	- Melhorar na visualização das atividades da equipe; - Facilitar a alocação de recursos; - Permitir a identificação de gargalos no fluxo.	- Visibilidade do projeto;	Todos os objetivos deste ciclo foram atingidos, através da implantação do quadro Kanban e foco da equipe na identificação de gargalos no fluxo de atividades.
CICLO 3: LIMITAÇÃO DO TRABALHO E MELHORIA DA COMUNICAÇÃO	- Eliminação de gargalos no fluxo;	- Ritmo sustentável; - Cliente presente; - Reuniões de pé; - Retrospectivas - Visibilidade do projeto.	Todos os objetivos deste ciclo foram atingidos. As reuniões periódicas foram implementadas, facilitando a comunicação entre as equipes e os gargalos foram reduzidos através da redução do excesso de paralelismo de atividades, utilizando o WIP.
CICLO 4: PRÁTICAS APLICADAS EM PROJETOS PILOTO	- Aplicar todas as práticas na execução de um projeto, e não só em seu monitoramento; - Aumentar a transparência no processo; - Melhorar a satisfação da Área de Negócios; - Reduzir os impactos das mudanças no processo.	- Ritmo sustentável; - Cliente presente; - Reuniões de pé; - Retrospectivas - Visibilidade do projeto; - Divisão em funcionalidades.	O objetivo da redução do impacto de mudanças no processo foi atingido parcialmente, pois a proposta só pode ser aplicada em dois projetos pilotos. Será necessário a gestão de mais projetos utilizando todas as práticas para verificar se a proposta contribuiu com este objetivo. Os demais objetivos foram atingidos, através da iniciativa de aproximar a Área de Negócios do ambiente de desenvolvimento, utilizando as visitas diárias e entregas curtas.

Ao observar as informações da Tabela 9, nota-se que todos os ciclos atingiram os objetivos propostos, com uma observação no ciclo 4. Neste caso, será necessário aplicar as práticas propostas em mais projetos para avaliar a capacidade de adequação do processo às mudanças no ambiente.

Nas seções a seguir, foram coletadas as percepções de todos os envolvidos neste iniciativa de melhoria em reuniões de retrospectiva com cada equipe.

5.2. AVALIAÇÃO DA PROPOSTA SEGUNDO A ÁREA DE NEGÓCIO

A primeira reunião foi realizada com a equipe da Área de Negócio envolvida na iniciativa de melhoria: o gerente da área, que acompanhou toda a aplicação das práticas, e o analista sênior que desempenhou o papel de usuário durante os projetos A e B. Esta equipe participou dos ciclos 3 e 4, além do questionário de caracterização das equipes, descritos no Capítulo 3.

Para dar uma visão geral do projeto realizado, foi apresentado um resumo de todo o planejamento e execução dos ciclos desta iniciativa de melhoria, inclusive dos ciclos que não envolveram diretamente a Área de Negócio. Em seguida, foi solicitado que os participantes da reunião indicassem os pontos positivos, negativos e comentários sobre os ciclos que participaram. Os resultados serão apresentados a seguir:

Em relação à aplicação do questionário de caracterização das equipes, foram identificados:

- **Pontos positivos:**
  - Iniciativa da área de TI para buscar feedback da área usuária sobre a condução dos projetos;
  - Disponibilidade para tirar dúvidas no preenchimento e explicar detalhes sobre cada critério abordado.
- **Pontos negativos:**
  - Dificuldade para o preenchimento da matriz de comparação dos critérios.
- **Comentários:**
  - Para a Área de Negócios, a iniciativa de buscar feedback das áreas usuárias deveria ser uma prática comum da área de TI. A sugestão dada foi a aplicação trimestral de uma pesquisa de satisfação com características similares a deste questionário, porém sem uma nova comparação dos critérios de sucesso.

Em relação ao ciclo 3 (Limitação do trabalho e melhoria da comunicação), foram identificados:

- **Pontos positivos:**
  - Transparência da área de TI com a Área de Negócio;

- Retorno das reuniões de acompanhamento de projetos e atividades, que não ocorria há algum tempo;
- Apresentação de novas ferramentas e métodos que podem ser utilizados pela Área de Negócios para gestão das atividades e apoio à tomada de decisão.

- **Pontos negativos:**

  Não foram citados.

- **Comentários:**
  - A equipe da Área de Negócio demonstrou interesse em conhecer mais detalhes sobre o Kanban, pois pretende utilizá-lo para gestão de suas atividades internas (atividades que não relacionadas aos projetos de TI).
  - No início do ano de 2014 será realizado um programa de formação de analistas na área, voltado aos atuais assistentes. A Área Técnica foi convidada para ministrar um treinamento conjunto com a Área de Negócios sobre gestão ágil de projetos.

Em relação ao ciclo 4 (Práticas aplicadas em projetos piloto), foram identificados:

- **Pontos positivos:**
  - Transparência das áreas de TI com a Área de Negócio;
  - Sensação de maior envolvimento e trabalho em equipe durante as visitas ao ambiente de desenvolvimento (Projeto A);
  - Conhecimento de etapas e problemas que não eram visíveis para a Área de Negócio antes das visitas ao ambiente de desenvolvimento (Projeto A);
  - Possibilidade de pequenos ajustes no escopo durante o desenvolvimento (Projetos A e B);
  - Redução da ansiedade e curiosidade da Área de Negócio ao antecipar o contato com a nova versão do produto (Projeto B);
- **Pontos negativos:**
  - Frequência de visitas ao ambiente de desenvolvimento (Projeto A). As visitas diárias nem sempre são produtivas pois não ocorrem grandes atualizações entre um dia e outro. Esta frequência poderá ser avaliada de acordo com cada projeto.
- **Comentários:**

- A Área de Negócio relatou que as duas estratégias utilizadas nos projetos A e B foram positivas, demonstrando interesse em utilizá-las novamente nos próximos projetos, de acordo com sua característica.

Levando em consideração todos os pontos citados pela Área de Negócio, é possível concluir que a aplicação desta proposta contribuiu positivamente para os critérios de satisfação do usuário e mudanças de escopo, citados como um dos mais relevantes para o sucesso de um projeto, de acordo com o resultado do questionário aplicado na Seção 3.3.2, descrito na Figura 16. O critério de qualidade, também citado na Figura 16 como um dos mais relevantes, não foi abordado pois não foi caracterizado como um problema relevante, segundo as percepções da Área de Negócio sobre o processo atual, descrito na Seção 3.3.1, na Figura 15.

5.3. AVALIAÇÃO DA PROPOSTA SEGUNDO A ÁREA TÉCNICA

A segunda reunião foi realizada com a equipe da Área Técnica envolvida na iniciativa de melhoria: o coordenador da área, o analista sênior e os dois analistas plenos, sendo um deles o autor desta monografia. Esta equipe participou dos ciclos 2, 3 e 4, além do questionário de caracterização das equipes, descritos no Capítulo 3.

Para dar uma visão geral do projeto realizado, foi apresentado um resumo de todo o planejamento e execução dos ciclos desta iniciativa de melhoria, inclusive dos ciclos que não envolveram diretamente a Área Técnica. Em seguida, foi solicitado que os participantes da reunião indicassem os pontos positivos, negativos e comentários sobre os ciclos que participaram. Os resultados serão apresentados a seguir:

Em relação à aplicação do questionário de caracterização das equipes, foram identificados:

- **Pontos positivos:**
  - Permitiu uma visão geral sobre a percepção das áreas envolvidas nos projetos de desenvolvimento de empresa;
  - Apresentou práticas que não eram conhecidas por toda a equipe;
  - Confirmou o foco da Área Técnica no atendimento das necessidades da Área de Negócio;
  - Permitiu identificar pontos de melhoria no processo de desenvolvimento utilizado na empresa.

- **Pontos negativos:**
    - O preenchimento da matriz de comparação dos critérios não era intuitivo. Foram necessárias algumas tentativas e maiores esclarecimentos para realizar o correto preenchimento;
    - A pesquisa poderia envolver as demais áreas de negócio que se relacionam com a Área Técnica.
- **Comentários:**
    - O ponto negativo indicado sobre a matriz de comparação de critérios foi relacionado apenas ao seu modo de apresentação e preenchimento. A equipe entende a importância desta comparação e da aplicação do modelo AHP.

Em relação ao ciclo 2 (Visualização das atividades), foram identificados:

- **Pontos positivos:**
    - Otimizou a gestão de atividades e visualização dos projetos da área, reduzindo o trabalho manual para manutenção das informações;
    - Permitiu a identificação de gargalos e realização de ajustes da capacidade da equipe, facilitando a alocação de recursos e reports gerenciais;
    - O quadro virtual permite que a gestão de atividades e coordenação de projetos seja realizada de forma mais fácil em equipes remotas, já que o compartilhamento de informações é feito via Internet;
    - Adoção de uma ferramenta gratuita, que atendeu às necessidades da área sem depender de custos externos;
    - Mostrou para a equipe as mudanças na rotina podem trazer pontos positivos.
- **Pontos negativos:**

  Não foram citados.
- **Comentários:**
    - A equipe se comprometeu a continuar utilizando o quadro Kanban para gestão das atividades, após verificarem as vantagens deste controle quando comparado aos controles feitos através de planilhas;
    - Um dos membros da equipe ficou interessado em conhecer mais sobre as práticas ágeis, solicitando a indicação de livros e sites sobre o assunto.

Em relação ao ciclo 3 (Limitação do trabalho e melhoria da comunicação) foram identificados:

- **Pontos positivos:**
    - Identificação da capacidade de trabalho da equipe, que pode ser utilizada para solicitar mais recursos à gerência;
    - Possibilidade de adequar a quantidade de demandas em andamento à capacidade de resolução da equipe, evitando atrasos;
    - Possibilidade de visualizar qualquer ociosidade na equipe;
    - Melhoria da comunicação interna e externa das equipes;
    - Aumento da confiança da Área de Negócios nos projetos de TI;
    - Melhoria no acompanhamento e priorização de atividades da Fábrica de Software.
- **Pontos negativos:**

  Não foram citados.
- **Comentários:**
    - A dificuldade para encontrar a quantidade de atividades em paralelo (WIP) foi citada inicialmente como um ponto negativo da implantação. Porém, a equipe chegou ao consenso de que esta é uma característica da prática e não um ponto negativo da proposta ou da própria prática;
    - O fato da visualização da ociosidade na equipe, citada como ponto positivo pelo coordenador da área, causou uma certa polêmica durante a reunião, pois um dos analistas questionou um possível "excesso" de monitoramento nos membros da equipe. Este ponto foi esclarecido pelo coordenador até o fim da reunião.

Em relação ao ciclo 4 (Práticas aplicadas em projetos piloto), foram identificados:

- **Pontos positivos:**
    - Visível satisfação da Área de Negócios em participar mais ativamente do projeto;
    - Permitir, de certa forma, a revisão ou validação de escopo durante a fase de desenvolvimento, ou entre cada entrega;
    - Aumentar a sensação de progresso do projeto, para Área de Negócio;

- Adiantar a execução de testes de aceitação de uma entrega, enquanto a próxima entrega está em desenvolvimento.

- **Pontos negativos:**
  - Aumento da carga de testes para a Área Técnica, pois um módulo com n entregas irá gerar n testes. Em alguns casos poderá ocorrer retrabalho, pois um mesmo plano de testes poderá ser executado em mais de uma entrega;
  - Necessidade de controle da ansiedade da Área de Negócios, pois ao participar diariamente (ou com maior frequência) do desenvolvimento, é comum solicitarem alterações fora do escopo inicial, além de prestarem "consultoria técnica" para construção das funcionalidades.

- **Comentários:**

  Não foram citados comentários adicionais para este ciclo.

Levando em consideração todos os pontos citados pela Área Técnica, é possível concluir que a aplicação desta proposta contribuiu positivamente para os critérios de satisfação do usuário e gerência de escopo, citados como um dos mais relevantes para o sucesso de um projeto, de acordo com o resultado do questionário aplicado na Seção 3.3.2, descrito na Figura 17.

5.4. AVALIAÇÃO DA PROPOSTA SEGUNDO A FÁBRICA DE SOFTWARE

A terceira reunião foi realizada apenas com o coordenador da equipe da Fábrica de *Software* envolvida na iniciativa de melhoria, representando os demais integrantes da equipe, que não tiveram disponibilidade para comparecer à reunião. Esta equipe participou dos ciclos 3 e 4, além do questionário de caracterização das equipes, descritos no Capítulo 3.

Para dar uma visão geral do projeto realizado, foi apresentado um resumo de todo o planejamento e execução dos ciclos desta iniciativa de melhoria, inclusive dos ciclos que não envolveram diretamente a Fábrica de *Software*. Em seguida, foi solicitado que o participante da reunião indicasse os pontos positivos, negativos e comentários sobre os ciclos que sua equipe participou. Os resultados serão apresentados a seguir:

Em relação à aplicação do questionário de mapeamento das equipes, foram identificados:

- **Pontos positivos:**

- Obter feedback da Fábrica de Software sobre o processo de desenvolvimento da empresa.

- **Pontos negativos:**
  - Como somente o coordenador preencheu o questionário, representando a visão de sua equipe, não foi possível apurar a real percepção da equipe com relação aos critérios abordados.
- **Observações:**
  - Assim como reportado pelas demais áreas, houve um pouco de dificuldade no preenchimento da comparação dos critérios de sucesso em projetos.

Em relação ao ciclo 3 (Limitação do trabalho e melhoria da comunicação), foram identificados:

- **Pontos positivos:**
  - Agilidade nas reuniões de acompanhamento de demandas em andamento, pois a Área de Negócios já possuía todo o histórico necessário. O foco da reunião passou a ser somente a atualização das informações e não foi mais necessário resgatar informações em e-mails e planilhas.
- **Pontos negativos:**

  Não foram citados.
- **Comentários:**

  Não foram citados comentários adicionais para este ciclo.

Em relação ao ciclo 4 (Práticas aplicadas em projetos piloto), foram identificados:

- **Pontos positivos:**
  - Compartilhar com todas as áreas envolvidas no projeto os problemas que podem ocorrer durante o desenvolvimento. Este fato auxilia na resolução dos problemas, pois em alguns casos a Fábrica de Software precisa solicitar prioridade para demais áreas da empresa (exemplo da indisponibilidade do ambiente de teste, que é solucionado por uma área de infraestrutura) e não obtém resposta;
  - Facilidade para solucionar dúvidas de entendimento que podem surgir durante o desenvolvimento, com as áreas técnica e de negócio mais

presentes. Antes estas dúvidas eram encaminhadas por e-mail e em alguns casos a resposta demorava para ser recebida;

- Aplicação de novos métodos com a preocupação de não impactar os processos internos da fábrica.

- **Pontos negativos:**
  - A periodicidade das visitas realizadas não foi considerada produtiva, pois o desenvolvedor precisava parar suas atividades para reportar pouco progresso, levando tempo para retomar a sua atividade. A sugestão da equipe foi realizar esta visita com maior periodicidade.
  - Na visão da equipe da fábrica, alguns problemas no projeto não deveriam envolver a Área de Negócio, pois estes podem não compreender a situação como a Área Técnica compreende. Isto poderia gerar um possível desentendimento entre as equipes.
- **Comentários:**

  Não foram citados comentários adicionais para este ciclo.

Levando em consideração todos os pontos citados pela Fábrica de Software, é possível concluir que a aplicação desta proposta não gerou impacto negativo nos seus processos internos, que não poderiam ser alterados por questões contratuais. Ainda assim, todos os ciclos puderam ser executados trazendo ganhos para a Área Técnica e Área de Negócio.

5.5. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os resultados da aplicação da iniciativa de melhoria, através de uma análise crítica sobre o planejamento dos ciclos e seus objetivos iniciais, além de reuniões de retrospectivas com as equipes envolvidas, buscando obter a percepção de cada integrante sobre os pontos positivos e negativos de cada ciclo.

Nota-se, através dos relatos, que a experiência de todos os envolvidos foi positiva e que iniciativas como estas são bem vindas no ambiente analisado.

O próximo capítulo apresenta as conclusões sobre o trabalho aplicado, as principais contribuições para a organização e as iniciativas que podem ser exploradas no futuro.

6. CONCLUSÕES

Neste capítulo, são apresentadas as conclusões finais do trabalho, com foco nas contribuições e trabalhos futuros na organização.

6.1. CONSIDERAÇÕES FINAIS

O objetivo desta monografia foi apresentar uma experiência de melhoria de processos de *software*, com foco na adoção de práticas ágeis em uma equipe de desenvolvimento de *software* de uma empresa de grande porte do segmento de telecomunicações, sediada no Rio de Janeiro.

A estratégia utilizada para minimizar a rejeição às mudanças foi incentivar o envolvimento das equipes no planejamento das propostas de melhoria e na execução de todas as atividades, além de não citar nomes de metodologias ou *frameworks*, antes da prática relacionada ser aplicada. Para reduzir a concorrência da aplicação das propostas com as atividades diárias, a duração do projeto foi prolongada e o esforço diário foi reduzido para que todos os ciclos propostos pudessem ser aplicados.

Com base nos resultados descritos no capítulo anterior, nota-se que é possível adotar práticas ágeis em um ambiente denominado tradicional, com processos e metodologias já definidas, sem a necessidade de adoção de uma metodologia ou *framework* por completo.

O critério chave para o sucesso desta implantação foi o incentivo das gerências envolvidas e a receptividade das áreas participantes às novas propostas. As principais dificuldades identificadas foram a rejeição natural da equipe durante a proposta do primeiro ciclo de melhoria e a concorrência das atividades diárias com as propostas deste projeto. As principais contribuições para a organização serão descritas na seção a seguir.

6.2. PRINCIPAIS CONTRIBUIÇÕES

Todos os objetivos secundários desta monografia foram atendidos: foram disseminadas novas práticas e conceitos entre as equipes envolvidas; a rejeição às mudanças de rotina foram minimizadas; houve evolução na gestão das atividades e a comunicação entre as equipes foi otimizada. Os controles evoluídos durante este projeto continuarão sendo aplicados pelas áreas após a conclusão do projeto, conforme indicado nas retrospectivas realizadas no Capítulo 5.

O objetivo principal desta monografia, que foi apresentar uma experiência de melhoria de processos de *software* com foco na adoção de práticas ágeis, foi atendido em pequena escala, através da aplicação em dois projetos curtos, facilitando a revisão de escopo durante o desenvolvimento e antecipando quaisquer mudanças de prioridade. Como próximo passo, a organização deverá adaptar e aplicar estas práticas em projetos com maior duração ou complexidade, para verificar se os ganhos da aplicação em projetos curtos serão mantidos.

6.3. TRABALHOS FUTUROS

Durante o desenvolvimento e a aplicação da proposta, algumas possibilidades de trabalhos futuros foram identificadas:

- Aplicação do escopo deste projeto envolvendo outras áreas técnicas, de negócio e equipes da Fábrica de Software;
- Evolução da utilização do Kanban na Área de Negócio, para otimizar sua a gestão de atividades;
- Manutenção e evolução da cultura de melhoria contínua na equipe da Área Técnica;
- Realização de treinamentos com demais equipes para compartilhar o conhecimento adquirido e a experiência desta implantação;
- Generalizar as etapas realizadas durante esta proposta em um processo que possa ser aplicado em outras organizações.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ABRANTES, J. F.; TRAVASSOS, G. H. Common agile practices in software processes. In: Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on. IEEE, 2011. p. 355-358.

Agile Manifesto. Disponível em <http://agilemanifesto.org>. Acesso em 22 nov. 2013.

BASSI FILHO, D. L. Experiências com desenvolvimento ágil. 2008. Tese de Doutorado. Universidade de São Paulo.

DYBA, T; DINGSOYR, T. What do we know about agile software development?. Software, IEEE, v. 26, n. 5, p. 6-9, 2009.

HIBBS, C.; JEWETT, S.; SULLIVAN, Mike. The art of lean software development: a practical and incremental approach. O'Reilly, 2009.

JALALI, S.; WOHLIN, C.. Agile practices in global software engineering-A systematic map. In: Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on. IEEE, 2010. p. 45-54.

KANBANTOOL. Disponível em <http://kanbantool.com/>. Acesso em 22 nov. 2013.

KNIBERG, H.; SKARIN, M. Kanban and Scrum making the most of both. Managing Editor: Diana Plesa. Enterprise software development series. InfoQ. USA. ISBN 978–0–557–13832–6, 2010.

KURAPATI, N; MANYAM, V. S. C.; PETERSEN, K. Agile software development practice adoption survey. In: Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2012. p. 16-30.

LANDIM, H. F. Uma Abordagem de Monitoramento dos Fatores e Condições que influenciam nas Práticas Ágeis. 2012. Tese de Mestrado. Universidade de Fortaleza.

MAHER, R. Increasing Team Productivity: A project focus. 2011.

Mountain Goat Software, Scrum Task Board. Disponível em: <http://www.mountaingoatsoftware.com/scrum/task-boards>. Acesso em 22 nov. 2013.

PETERSEN, K. Implementing Lean and Agile software development in industry. 2010.

PROJECT MANAGEMENT INSTITUTE. A Guide to the Project Management Body of Knowledge: PMBOK® Guide, 2008.

POPPENDIECK, M.; POPPENDIECK, T. Lean Software Developement. Number ISBN 0-321-15078-3 in The Agile Software Developement Series. 2003.

PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7ª Edição. Ed: McGraw Hill, 2011.

RIBEIRO, B. B.; MODESTO, D. M.; SANTOS, G. Avaliação da Importância dos Fatores Técnicos e Ambientais do Método Pontos por Caso de Uso com Base no Método AHP. In: CIbSE. 2012. p. 210-223.

SCHWABER, K; SUTHERLAND, J. The Scrum Guide–The Definitive Guide to Scrum: The Rules of the Game. Scrum. org, 2011.

SCHWABER, K. SCRUM Development Process. In: Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA). 1995.

SCHWABER, K; BEEDLE, M. Agile software development with Scrum. Upper Saddle River: Prentice Hall, 2002.

SILVA, D. V. S.; SANTOS, F. A. O.; NETO, Pedro Santos. Os benefícios do uso de Kanban na gerência de projetos de manutenção de software.

The Standish Group: CHAOS Summary for 2010. Disponível em <http://insyght.com.au/special/2010CHAOSSummary.pdf>. Acesso em 22 nov. 2013.

SUTHERLAND, J. Agile development: Lessons learned from the first scrum. Cutter Agile Project Management Advisory Service: Executive Update, v. 5, n. 20, p. 1-4, 2004.

DESENVOLVIMENTO ÀGIL, Scrum. Disponível em: <http://desenvolvimentoagil.com.br/scrum>. Acesso em 22 nov. 2013.

VARGAS, R. V.; IPMA-B, P. M. P. Utilizando a programação multicritério (Analytic Hierarchy Process-AHP) para selecionar e priorizar projetos na gestão de portfólio. In: PMI Global Congress. 2010. p. 1-22.

WILLIAMS, L. Agile software development methodologies and practices. Advances in Computers, v. 80, p. 1-44, 2010.

ANEXO I – QUESTIONÁRIOS APLICADOS

O Questionário A, encaminhado para a Área de Negócios será exibido a seguir.

Questionário inicial

Este questionário faz parte de um estudo cujo objetivo é verificar como a adoção das práticas propostas pelas metodologias ágeis podem ajudar a reduzir o impacto de mudanças de requisitos no custo e prazo dos projetos de desenvolvimento de software.

Como resultado deste questionário, pretende-se obter a percepção dos envolvidos neste processo sobre a relevância de cada um dos critérios apresentados em relação aos demais. Além disso, o processo de desenvolvimento utilizado na sua organização atual também será avaliado.

Caso tenha alguma dúvida no preenchimento, consulte as instruções contidas em cada aba da planilha.

Ao final do preenchimento salve a planilha e devolva ao solicitante para que seus dados sejam adicionados a pesquisa. Os dados serão sigilosos e não será revelada a origem da informação ou informante.

Critérios de sucesso na execução de projetos de desenvolvimento de software

O objetivo desta sessão é identificar sua percepção sobre a relevância de cada critério apresentado em relação aos demais.

Será exibida uma matriz comparando os critérios entre si, para que você possa indicar qual o grau de importância de um critério sobre o outro (sempre comparando o critério da linha com o das colunas correspondentes). Caso tenha dúvidas com a escala de importância apresentada, a aba "***Escala de comparação***" contém uma descrição detalhada de cada grau de importância.

Ao final desta comparação, serão identificados os critérios mais relevantes para o sucesso de projetos de desenvolvimento de software, de acordo sua opinião.

01. Para cada critério apresentado na matriz abaixo, escolha um valor nas células em branco que identifique qual a importância, na sua opinião, dos critérios apresentados nas linhas com relação aos critérios apresentados nas colunas.

Para facilitar o preenchimento, o próximo campo que você deverá preencher estará colorido em amarelo.

Por exemplo: Se na sua opinião o critério "***Entregar o produto de acordo com o escopo definido***", apresentado na linha 3, é um pouco mais importante que o critério "***Entregar o produto dentro do valor orçado***", apresentado na coluna 2, então a resposta selecionada para a célula correspondente à linha 3 e coluna 2 deverá ser "***Importância pequena***".

		1	2	3	4	5	6
		Entregar o produto no prazo previsto	**Entregar o produto dentro do valor orçado**	**Entregar o produto de acordo com o escopo definido**	**Entregar o produto que melhor atende às necessidades do cliente**	**Entregar o produto sem bugs identificados**	**Absorver mudanças de escopo durante o projeto**
1	**Entregar o produto no prazo previsto**						
2	**Entregar o produto dentro do valor orçado**						
3	**Entregar o produto de acordo com o escopo definido**						
4	**Entregar o produto que melhor atende às necessidades do cliente**						
5	**Entregar o produto sem bugs identificados**						
6	**Absorver mudanças de escopo durante o projeto**						

Sucesso dos projetos de software

O objetivo desta sessão é identificar a taxa de sucesso dos projetos de desenvolvimento de software que você já participou em toda sua vida profissional e na sua organização atual.

02. Com base nos projetos de desenvolvimento de software que você participou em toda sua experiência profissional, indique sua percepção sobre o percentual de projetos que se enquadram nas situações a seguir:

	0 a 25%	26 a 50%	51 a 75%	76 a 100%
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	○	○	○	○
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	○	○	○	○
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	○	○	○	○

03. Com base nos projetos de desenvolvimento de software que você participou em sua organização atual, indique sua percepção sobre o percentual de projetos que se enquadram nas situações a seguir:

	0 a 25%	26 a 50%	51 a 75%	76 a 100%
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	○	○	○	○
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	○	○	○	○
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	○	○	○	○

04. Qual a sua percepção sobre quanto o processo utilizado para o desenvolvimento de software em sua organização atual consegue auxiliar no atendimento dos objetivos do projeto com relação à:

	Auxilia muito	Auxilia pouco	Não auxilia / Indiferente	Atrapalha
Prazo	○	○	○	○
Custo	○	○	○	○
Escopo	○	○	○	○
Satisfação do cliente	○	○	○	○
Qualidade do produto	○	○	○	○
Mudanças de escopo	○	○	○	○

Conclusão

Muito obrigado por sua disponibilidade ao preencher este questionário!

Por favor, salve a planilha e devolva ao solicitante para que seus dados sejam adicionados a pesquisa. Os dados serão sigilosos e não será revelada a origem da informação ou informante.

Os resultados apurados para estas equipes foram consolidados, conforme detalhes a seguir:

01. Critérios de sucesso (Matriz Eigen)

		Entrevistado 1	Entrevistado 2
1	**Entregar o produto no prazo previsto**	0,033964375	0,051632827
2	**Entregar o produto dentro do valor orçado**	0,033964375	0,027172907
3	**Entregar o produto de acordo com o escopo definido**	0,070321134	0,055115155
4	**Entregar o produto que melhor atende às necessidades**	0,155193388	0,42081126
5	**Entregar o produto sem bugs identificados**	0,456237754	0,227842259
6	**Absorver mudanças de escopo durante o projeto**	0,250318975	0,217425592

02. Sucesso nos projetos em toda sua experiência profissional:

	Entrevistado 1	Entrevistado 2
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	0 a 25%	0 a 25%
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	76 a 100%	76 a 100%
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	0 a 25%	0 a 25%

03. Sucesso nos projetos em sua organização atual:

	Entrevistado 1	Entrevistado 2
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	0 a 25%	0 a 25%
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	76 a 100%	76 a 100%
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	0 a 25%	0 a 25%

04. Percepção quanto ao processo da organização atual:

	Entrevistado 1	Entrevistado 2
Prazo	Auxilia pouco	Não auxilia / Indiferente
Custo	Não auxilia / Indiferente	Auxilia pouco
Escopo	Atrapalha	Atrapalha
Satisfação do cliente	Auxilia pouco	Atrapalha
Qualidade do produto	Auxilia pouco	Não auxilia / Indiferente
Mudanças de escopo	Atrapalha	Atrapalha

O Questionário B, encaminhado para a Área Técnica e para a Fábrica de *Software* será exibido a seguir.

Questionário inicial

Este questionário faz parte de um estudo cujo objetivo é verificar como a adoção das práticas propostas pelas metodologias ágeis podem ajudar a reduzir o impacto de mudanças de requisitos no custo e prazo dos projetos de desenvolvimento de software.

Como resultado deste questionário, pretende-se obter a percepção dos envolvidos neste processo sobre a relevância de cada um dos critérios apresentados em relação aos demais.

Caso tenha alguma dúvida no preenchimento, consulte as instruções contidas em cada aba da planilha.

Ao final do preenchimento salve a planilha e devolva ao solicitante para que seus dados sejam adicionados a pesquisa. Os dados serão sigilosos e não será revelada a origem da informação ou informante.

Seu perfil profissional

O objetivo desta sessão é conhecer seu perfil profissional.

01. Você possui quantos anos de experiência na área de TI?

- ◯ Menos de 2 anos
- ◯ 2 a 5 anos
- ◯ 5 a 10 anos
- ◯ Mais de 10 anos

02. Por favor, indique sua experiência nos seguintes papéis (Marque todas as opções que se aplicam):

	Nenhuma	Menos de 2 anos	2 a 5 anos	5 a 10 anos	Mais de 10 anos
Analista de produção / operação	◯	◯	◯	◯	◯
Analista de sistemas (Requisitos / Design / Projeto)	◯	◯	◯	◯	◯
Analista de testes / QA	◯	◯	◯	◯	◯
Desenvolvedor	◯	◯	◯	◯	◯
Gerente de projetos	◯	◯	◯	◯	◯
Outro (especifique):	◯	◯	◯	◯	◯

03. Quais papéis abaixo se adequam à sua posição na empresa atual? (Marque todas as opções que se aplicam)

- ☐ Analista de produção / operação
- ☐ Analista de sistemas (Requisitos / Design / Projeto)
- ☐ Analista de testes / QA
- ☐ Desenvolvedor
- ☐ Gerente de projetos
- ☐ Outros (especifique): _____

Práticas ágeis

O objetivo desta sessão é verificar seus conhecimentos nas práticas propostas por diferentes metodologias / frameworks ágeis.

04. Quais são as metodologias / frameworks ágeis que você conhece?

- ☐ Extreme programming (XP)
- ☐ Kanban
- ☐ Desenvolvimento enxuto de software (Lean Software Development)
- ☐ Scrum
- ☐ Outros (especifique): _____

05. Quais são as práticas ágeis que você conhece?

☐ Divisão em funcionalidades (Features / Stories)	☐ Desenvolvimento dirigido por testes (Test Driven Development)
☐ Backlog do produto (Product Backlog)	☐ Automação de testes (Automated testing)
☐ Metáforas do sistema (Metaphor)	☐ Cliente presente (On-site customer)
☐ Padrões de codificação (Coding standards)	☐ Ritmo sustentável (Sustainable Pace / 40 hour week)
☐ Código coletivo (Collective Code Ownership)	☐ Times multidisciplinares (Whole team / multi-skill teams)
☐ Integração contínua (Continuous integration)	☐ Jogo do planejamento (Planning Game / Planning Poker)
☐ Programação em par (Pair programming)	☐ Visibilidade do projeto (Project visibility)
☐ Refatoração do código (Refactoring)	☐ Retrospectivas (Retrospective)
☐ Entregas curtas (Small releases)	☐ Reuniões Scrum (Scrum Meetings)
☐ Reuniões de pé (Stand-up meetings)	☐ Outras (especifique): _____

Critérios de sucesso na execução de projetos de desenvolvimento de software

O objetivo desta sessão é identificar sua percepção sobre a relevância de cada critério apresentado em relação aos demais.

Será exibida uma matriz comparando os critérios entre si, para que você possa indicar qual o grau de importância de um critério sobre o outro (sempre comparando o critério da linha com o das colunas correspondentes). Caso tenha dúvidas com a escala de importância apresentada, a aba "***Escala de comparação***" contém uma descrição detalhada de cada grau de importância.

Ao final desta comparação, serão identificados os critérios mais relevantes para o sucesso de projetos de desenvolvimento de software, de acordo sua opinião.

06. Para cada critério apresentado na matriz abaixo, escolha um valor nas células em branco que identifique qual a importância, na sua opinião, dos critérios apresentados nas linhas com relação aos critérios apresentados nas colunas.

Para facilitar o preenchimento, o próximo campo que você deverá preencher estará colorido em amarelo.

Por exemplo: Se na sua opinião o critério "***Entregar o produto de acordo com o escopo definido***", apresentado na linha 3, é um pouco mais importante que o critério "***Entregar o produto dentro do valor orçado***", apresentado na coluna 2, então a resposta selecionada para a célula correspondente à linha 3 e coluna 2 deverá ser "***Importância pequena***".

		1	2	3	4	5	6
		Entregar o produto no prazo previsto	Entregar o produto dentro do valor orçado	Entregar o produto de acordo com o escopo definido	Entregar o produto que melhor atende às necessidades do cliente	Entregar o produto sem bugs identificados	Absorver mudanças de escopo durante o projeto
1	Entregar o produto no prazo previsto						
2	Entregar o produto dentro do valor orçado						
3	Entregar o produto de acordo com o escopo definido						
4	Entregar o produto que melhor atende às necessidades do cliente						
5	Entregar o produto sem bugs identificados						
6	Absorver mudanças de escopo durante o projeto						

Sucesso dos projetos de software

O objetivo desta sessão é identificar a taxa de sucesso dos projetos de desenvolvimento de software que você já participou em toda sua vida profissional e na sua organização atual.

07. Com base nos projetos de desenvolvimento de software que você participou em toda sua experiência profissional, indique sua percepção sobre o percentual de projetos que se enquadram nas situações a seguir:

	0 a 25%	26 a 50%	51 a 75%	76 a 100%
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	○	○	○	○
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	○	○	○	○
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	○	○	○	○

08. Com base nos projetos de desenvolvimento de software que você participou em sua organização atual, indique sua percepção sobre o percentual de projetos que se enquadram nas situações a seguir:

	0 a 25%	26 a 50%	51 a 75%	76 a 100%
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	○	○	○	○
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	○	○	○	○
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	○	○	○	○

09. Qual a sua percepção sobre quanto o processo utilizado para o desenvolvimento de software em sua organização atual consegue auxiliar no atendimento dos objetivos do projeto com relação à:

	Auxilia muito	Auxilia pouco	Não auxilia / Indiferente	Atrapalha
Prazo	○	○	○	○
Custo	○	○	○	○
Escopo	○	○	○	○
Satisfação do cliente	○	○	○	○
Qualidade do produto	○	○	○	○
Mudanças de escopo	○	○	○	○

Conclusão

Muito obrigado por sua disponibilidade ao preencher este questionário!

Por favor, salve a planilha e devolva ao solicitante para que seus dados sejam adicionados a pesquisa. Os dados serão sigilosos e não será revelada a origem da informação ou informante.

Os resultados apurados para estas equipes foram consolidados, conforme detalhes a seguir:

01. Você possui quantos anos de experiência na área de TI?

Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
5 a 10 anos	5 a 10 anos	Mais de 10 anos	Mais de 10 anos	5 a 10 anos

02. Por favor, indique sua experiência nos seguintes papéis (Marque todas as opções que se aplicam):

	Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Analista de produção / operação	Nenhuma	Nenhuma	Menos de 2 anos	2 a 5 anos	Nenhuma
Analista de sistemas (Requisitos / Design / Projeto)	5 a 10 anos	2 a 5 anos	Mais de 10 anos	Mais de 10 anos	2 a 5 anos
Analista de testes / QA	2 a 5 anos	2 a 5 anos	Nenhuma	2 a 5 anos	5 a 10 anos
Desenvolvedor	Menos de 2 anos	Nenhuma	2 a 5 anos	2 a 5 anos	5 a 10 anos
Gerente de projetos	5 a 10 anos	2 a 5 anos	5 a 10 anos	Mais de 10 anos	2 a 5 anos
Outros: Analista de processos	Menos de 2 anos				

03. Quais papéis abaixo se adequam à sua posição na empresa atual? (Marque todas as opções que se aplicam)

Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Analista de sistemas	Gerente de projetos	Analista de sistemas	Analista de sistemas	Analista de testes
Analista de testes		Analista de testes	Gerente de projetos	Desenvolvedor
Gerente de projetos		Gerente de projetos	Coordenador de projetos	Gerente de projetos
		Coordenador de projetos		

04. Quais são as metodologias / frameworks ágeis que você conhece?

Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Extreme programming	Extreme programming	Extreme programming	Extreme programming	Extreme programming
Kanban	Kanban	Scrum	Scrum	Scrum
Scrum	Scrum			

05. Quais são as práticas ágeis que você conhece?

Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Divisão em funcionalidades (Features / Stories)		Divisão em funcionalidades (Features / Stories)		Divisão em funcionalidades (Features / Stories)
Backlog do produto (Product Backlog)				
Metáforas do sistema (Metaphor)				
Padrões de codificação (Coding standards)	Padrões de codificação (Coding standards)	Padrões de codificação (Coding standards)	Padrões de codificação (Coding standards)	Padrões de codificação (Coding standards)
				Código coletivo (Collective Code Ownership)
		Integração contínua (Continuous integration)	Integração contínua (Continuous integration)	
Programação em par (Pair programming)	Programação em par (Pair programming)		Programação em par (Pair programming)	Programação em par (Pair programming)
Refatoração do código (Refactoring)	Refatoração do código (Refactoring)	Refatoração do código (Refactoring)	Refatoração do código (Refactoring)	Refatoração do código (Refactoring)
Entregas curtas (Small releases)	Entregas curtas (Small releases)		Entregas curtas (Small releases)	Entregas curtas (Small releases)
Reuniões de pé (Stand-up meetings)	Reuniões de pé (Stand-up meetings)	Reuniões de pé (Stand-up meetings)	Reuniões de pé (Stand-up meetings)	Reuniões de pé (Stand-up meetings)
			Desenvolvimento dirigido por testes (Test Driven Development)	Desenvolvimento dirigido por testes (Test Driven Development)
Automação de testes (Automated testing)	Automação de testes (Automated testing)		Automação de testes (Automated testing)	Automação de testes (Automated testing)
Cliente presente (On-site customer)				
Ritmo sustentável (Sustainable Pace / 40 hour week)				
Times multidisciplinares (Whole team / multi-skill teams)		Times multidisciplinares (Whole team / multi-skill teams)	Times multidisciplinares (Whole team / multi-skill teams)	Times multidisciplinares (Whole team / multi-skill teams)
Jogo do planejamento (Planning Game / Planning Poker)				Jogo do planejamento (Planning Game / Planning Poker)
Visibilidade do projeto (Project visibility)			Visibilidade do projeto (Project visibility)	
Retrospectivas (Retrospective)			Retrospectivas (Retrospective)	Retrospectivas (Retrospective)
Reuniões Scrum (Scrum Meetings)	Reuniões Scrum (Scrum Meetings)	Reuniões Scrum (Scrum Meetings)	Reuniões Scrum (Scrum Meetings)	Reuniões Scrum (Scrum Meetings)
Outras: Definição de "pronto"				

06. Critérios de sucesso (Matriz Eigen)

	Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Entregar o produto no prazo previsto	0,043569934	0,039827572	0,051373294	0,06160809	0,162095997
Entregar o produto dentro do valor orçado	0,028393782	0,046611053	0,164693393	0,177637398	0,254403933
Entregar o produto de acordo com o escopo definido	0,065975748	0,322855083	0,160973155	0,179451621	0,218689647
Entregar o produto que melhor atende às necessidades do	0,366378572	0,137040876	0,512319781	0,45245072	0,12596592
Entregar o produto sem bugs identificados	0,145432424	0,354906365	0,030195158	0,034524401	0,203436671
Absorver mudanças de escopo durante o projeto	0,35024954	0,098759051	0,080445217	0,09432777	0,035407831

07. Sucesso nos projetos em toda sua experiência profissional:

	Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	26 a 50%	26 a 50%	26 a 50%	26 a 50%	0 a 25%
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	51 a 75%	51 a 75%	51 a 75%	26 a 50%	76 a 100%
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	0 a 25%	0 a 25%	0 a 25%	0 a 25%	0 a 25%

08. Sucesso nos projetos em sua organização atual:

	Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Concluídos com sucesso (Atenderam as expectativas de custo, prazo e qualidade)	0 a 25%	26 a 50%	0 a 25%	0 a 25%	0 a 25%
Concluídos com problemas (Não atenderam as expectativas de custo, prazo ou qualidade)	76 a 100%	51 a 75%	76 a 100%	51 a 75%	76 a 100%
Falharam (Foram cancelados ou foram concluídos e nunca utilizados)	0 a 25%	0 a 25%	0 a 25%	0 a 25%	0 a 25%

09. Percepção quanto ao processo da organização atual:

	Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4	Entrevistado 5
Prazo	Auxilia pouco	Atrapalha	Auxilia muito	Auxilia pouco	Não auxilia / Indiferente
Custo	Não auxilia / Indiferente	Atrapalha	Auxilia pouco	Auxilia pouco	Não auxilia / Indiferente
Escopo	Atrapalha	Atrapalha	Atrapalha	Atrapalha	Atrapalha
Satisfação do cliente	Atrapalha	Atrapalha	Não auxilia / Indiferente	Não auxilia / Indiferente	Não auxilia / Indiferente
Qualidade do produto	Auxilia pouco	Atrapalha	Auxilia pouco	Auxilia pouco	Não auxilia / Indiferente
Mudanças de escopo	Atrapalha	Atrapalha	Atrapalha	Atrapalha	Atrapalha

ANEXO II – APLICAÇÃO DO MODELO AHP PARA ANÁLISE DE CRITÉRIOS

O AHP (*Analytic Hierarchy Process*) é uma técnica estruturada para tomada de decisão em ambientes complexos em que diversas variáveis ou critérios são considerados para a priorização e seleção de alternativas ou projetos (VARGAS, 2010).

A utilização do AHP se inicia pela decomposição do problema em uma hierarquia de critérios mais facilmente analisáveis e comparáveis de modo independente. A partir da construção desta hierarquia lógica, os tomadores de decisão avaliam sistematicamente as alternativas por meio da comparação, de duas a duas, dentro de cada um dos critérios. Essa comparação pode utilizar dados concretos das alternativas ou julgamentos humanos como forma de informação subjacente (SAATY, 2008, *apud* VARGAS, 2010).

O primeiro passo para aplicação do modelo AHP é a determinação dos critérios que serão utilizados na comparação. Os critérios selecionados estão relacionados a seguir e serão comparados quanto a sua contribuição para o sucesso na execução de um projeto de desenvolvimento de software:

- **Relacionado ao prazo:** Entregar o produto no prazo previsto.
- **Relacionado ao custo:** Entregar o produto dentro do valor orçado.
- **Relacionado ao escopo:** Entregar o produto de acordo com o escopo definido.
- **Relacionado à satisfação do cliente:** Entregar o produto que melhor atende às necessidades do cliente.
- **Relacionado à qualidade:** Entregar o produto sem bugs identificados.
- **Relacionado às mudanças de escopo:** Absorver mudanças de escopo durante o projeto.

No segundo passo, os critérios foram avaliados dois a dois, visando determinar a importância relativa entre eles e sua contribuição para o objetivo selecionado. A escala de comparação dos critérios utilizada foi a proposta por Saaty (SAATY, 2008, *apud* RIBEIRO *et al,* 2012), exibida na Tabela 10. A avaliação foi realizada utilizando os questionários contidos no Anexo I desta monografia, na Seção "Critérios de sucesso na execução de projetos de desenvolvimento de software", através da matriz comparativa exibida na Tabela 11.

Tabela 10: Escala de comparação utilizada para o modelo AHP.

Definição	Intensidade de importância	Exemplificação
Importância absoluta do critério da linha sobre o critério da coluna	9	Favorecimento do critério apresentado na linha em relação ao critério apresentado na coluna com o mais alto grau de certeza
Importância muito grande do critério da linha sobre o critério da coluna	7	O critério apresentado na linha é fortemente favorecido em relação ao critério apresentado na coluna, e sua dominação é demonstrada na prática
Importância grande do critério da linha sobre o critério da coluna	5	Sua experiência e julgamento favorecem fortemente o critério apresentado na linha em relação ao critério apresentado na coluna
Importância pequena do critério da linha sobre o critério da coluna	3	Sua experiência e julgamento favorecem levemente o critério apresentado na linha em relação ao critério apresentado na coluna
Mesma importância entre os critérios	1	Os dois critérios contribuem igualmente para o sucesso do projeto
Importância pequena do critério da coluna sobre o critério da linha	1/3	Sua experiência e julgamento favorecem levemente o critério apresentado na coluna em relação ao critério apresentado na linha
Importância grande do critério da coluna sobre o critério da linha	1/5	Sua experiência e julgamento favorecem fortemente o critério apresentado na coluna em relação ao critério apresentado na linha
Importância muito grande do critério da coluna sobre o critério da linha	1/7	O critério apresentado na coluna é fortemente favorecido em relação ao critério apresentado na linha, e sua dominação é demonstrada na prática
Importância absoluta do critério da coluna sobre o critério da linha	1/9	Favorecimento do critério apresentado na coluna em relação ao critério apresentado na linha com o mais alto grau de certeza

Tabela 11: Matriz comparativa dos critérios avaliados.

	Prazo	Custo	Escopo	Satisfação do cliente	Qualidade	Mudanças de escopo
Prazo	1					
Custo		1				
Escopo			1			
Satisfação do cliente				1		
Qualidade					1	
Mudanças de escopo						1
Total						

Após os entrevistados preencherem a matriz comparativa dos critérios, foi necessário normalizar todas as matrizes para interpretar e dar pesos relativos a todos os critérios. A normalização é feita através da divisão de cada valor da tabela pelo valor total de sua coluna.

Em seguida, é feita a determinação da contribuição de cada critério das matrizes para o objetivo citado (sucesso em projetos de desenvolvimento de software). Esta contribuição é calculada a partir do vetor de prioridade, também chamado de vetor de Eigen. O vetor de Eigen apresenta os pesos relativos entre os critérios e é obtido de modo aproximado através da média aritmética dos valores de cada um dos critérios, ou seja, de cada linha da matriz normalizada exibida na Tabela 11, gerando uma tabela conforme a Tabela 12.

Tabela 12: Vetor de Eigen.

	Vetor de Eigen (% de contribuição)
Prazo	
Custo	
Escopo	
Satisfação do cliente	
Qualidade	
Mudanças de escopo	

O cálculo exato do vetor de Eigen é determinado apenas em casos específicos. A maioria dos casos práticos utiliza essa aproximação visando simplificar o processo de cálculo, uma vez que a diferença entre o valor real e o valor aproximado é inferir a 10% (KOSTLAN, 1991, *apud* VARGAS, 2010).

O próximo passo é a verificação de inconsistências nos dados. A verificação visa captar se os tomadores de decisão foram consistentes nas suas opiniões para a tomada de decisão. (TEKNOMO, 2006, *apud* VARGAS, 2010). Se, por exemplo, um entrevistado afirmar que o Custo é mais importante do que o Escopo e que o Escopo é mais importante do que Prazo, seria uma inconsistência na tomada de decisão se este entrevistado afirmasse que o Prazo é mais importante do que o Custo (se $A > B$ e $B > C$ seria inconsistente afirmar que $A < C$).

O índice de inconsistência tem como base o número principal de Eigen, que é calculado através do somatório do produto de cada elemento do vetor de Eigen calculado no passo anterior (Tabela 12) pelo total da respectiva coluna da matriz comparativa original (Tabela 11). A Tabela 13 apresenta o cálculo do número principal de Eigen (λ max):

Tabela 13: Cálculo do número principal de Eigen.

	Prazo	Custo	Escopo	Satisfação do cliente	Qualidade	Mudanças de escopo
Vetor Eigen						
Total						
Valor principal Eigen (λ_{max})						

O cálculo do índice de consistência é dado pela equação exibida na Figura 26, onde *CI* corresponde ao índice de consistência e *n* é o número de critérios avaliados (SAATY, 2005 *apud* VARGAS, 2010):

$$CI = \frac{\lambda_{Max} - n}{n - 1}$$

Figura 26: Cálculo do índice de consistência.
(SAATY, 2005 apud VARGAS, 2010)

Para verificar se o valor encontrado do índice de consistência (CI) é adequado, (SAATY, 2005, *apud* VARGAS, 2010) propôs o que foi chamado de taxa de consistência (CR). Ela é determinada pela razão entre o valor do índice de consistência (CI) e o índice de consistência aleatória (RI). A matriz será considerada consistente se a razão for menor que 10%.

O valor do índice de consistência aleatória (RI) é fixo e tem como base o número de critérios avaliados, conforme a Tabela 14:

Tabela 14: Valores do índice de consistência aleatória.
Fonte: (SAATY, 2005, *apud* VARGAS, 2010)

N	1	2	3	4	5	6	7	8	9	10
RI	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

Com a aplicação destas técnicas, foi possível identificar a relevância dos critérios apresentados para o sucesso de um projeto de *software*, segundo as áreas de negócio, técnica e Fábrica de Software.