

## **Lean Thinking aplicado ao gerenciamento de projetos: Um estudo de caso no setor de tecnologia da informação**

**LUIZA PAIS CORREA**

Pontifícia Universidade Católica de Minas Gerais  
luizapaiscorrea@gmail.com

**PAULA KARINA SALUME**

Pontifícia Universidade Católica de Minas Gerais  
paulasalume@hotmail.com



## **LEAN THINKING APLICADO AO GERENCIAMENTO DE PROJETOS: UM ESTUDO DE CASO NO SETOR DE TECNOLOGIA DA INFORMAÇÃO**

### **Resumo**

No contexto atual, as organizações se veem obrigadas a melhorar continuamente seus produtos e serviços, a fim de garantir máxima eficiência operacional e fidelização do cliente, por meio de uma gestão inovadora e ágil. O presente artigo tem por objetivo identificar os desperdícios no gerenciamento ágil de projetos de desenvolvimento de softwares, a fim de compreender se os princípios do *Lean Thinking* podem ser empregados na organização, possibilitando entregar mais valor para seus clientes. Com o intuito de fundamentar os aspectos observados, a metodologia utilizada inclui a revisão da literatura da área e a aplicação de pesquisa qualitativa tendo como objeto de estudo quatro projetos de uma empresa que utiliza gerenciamento ágil. Os resultados indicaram que a maneira como os projetos são gerenciados está diretamente ligada à eficiência de sua execução e, consequentemente, a geração de desperdícios. Desse modo, adotar princípios *lean* como alternativa para melhoria dos processos, não significa apenas reduzir desperdícios, mas também como adicionar valor para o cliente e para o negócio. Cumprir com todas as etapas e os requisitos de melhores práticas, se torna um grande desafio frente a constantes mudanças, seja no ambiente interno ou externo à organização.

**Palavras-chave:** Desperdícios. Gerenciamento de projetos. *Lean thinking*. *Software*

### **Abstract**

In the current context, organizations see themselves forced to continuously improve their products and services in order to ensure maximum operational efficiency and customer loyalty through innovative and agile management. The purpose of the present paper is to identify the wastes in the agile management of software projects development in order to understand if the principles of Lean Thinking can be applied in organizations to deliver more value to customers. Aiming to base the observed aspects, the methodology used includes the review of the area literature and the application of a qualitative research having as object of study four projects of a company that applies agile management. The results indicated that the way projects are managed is directly linked to their execution efficiency and consequently with waste generation. Therefore, the adoption of lean principles as an alternative to improve processes means not only reducing waste, but also adding value to the customer and business. Comply with all the steps and requirements of best practices becomes a great challenge in the face of constant changes, whether in the internal or external organization environment.

**Keywords:** Wastes. Project Management. *Lean thinking*. *Software*

### **1 Introdução**

A inovação tecnológica é um fator essencial para garantir a sobrevivência das organizações, tornando-as cada vez mais produtivas e competitivas, o que levou nos últimos anos, à ocorrência de grandes mudanças no modo de trabalho, de planejamento, de produção e



de consumo da sociedade. A partir da revolução industrial e seus impactos sobre o crescimento da produtividade, ocorreram sucessivas inovações, desde a inserção de máquinas e equipamentos até as formas de organização e gestão do trabalho. As empresas vivem e morrem em sua capacidade de descobrir novos negócios e criar valor para os clientes. A pressão competitiva está aumentando, impulsionada pelas mudanças na tecnologia e na sociedade (HUMBLE; MOLESKY; O'REILLY, 2015).

A revolução tecnológica, parte da quarta revolução industrial, transformará fundamentalmente a forma de viver, trabalhar e de se relacionar, afirma Schwab (2017). A quarta revolução industrial representa um alcance muito amplo dos avanços em áreas do domínio físico, digital e biológico, em direção a novos sistemas que foram construídos sobre a infraestrutura da revolução digital anterior (SCHWAB, 2017). A sobrevivência em longo prazo de qualquer empresa depende da sua capacidade de compreender as forças culturais e técnicas que aceleram os ciclos de inovação (HUMBLE; MOLESKY; O'REILLY, 2015).

Diante desse cenário, Conforto (2013, p.141) afirma que “existe uma necessidade inerente e constante de mudar, de se ter agilidade, seja no processo de desenvolvimento de produtos, no processo de manufatura ou toda organização”. Nessa busca, as organizações têm adotado a abordagem ágil no gerenciamento de projetos, o que envolve um alto grau de inovação e quebras de paradigmas. A agilidade nas empresas não pode ser alcançada usando processos e arquitetura tradicionais, ela deve ser incorporada nos principais processos e funções, levando em consideração a complexidade da organização (KENETT, 2013).

No 11º Relatório Anual Ágil, a empresa VersionOne (2017) identificou que a agilidade acelera a entrega do valor do negócio, e que as principais razões para adoção da abordagem ágil são: acelerar a entrega do produto, melhorar a capacidade de gerir mudanças, aumentar a produtividade, melhorar a visibilidade do projeto, reduzir o risco, melhorar o moral da equipe, reduzir o custo, entre outros.

Entre os métodos empregados para gerar maior valor para o cliente, está o pensamento enxuto (*Lean Thinking*). As organizações que aplicam princípios enxutos ao gerenciamento de programas e projetos são capazes de identificar e eliminar os processos que geram desperdícios, o que aumenta sua eficiência, criando valor para o cliente e a organização (GALE, 2013). Estudos mostram que dentre as atividades que estão sendo executadas, há 60% de resíduos, assim a abordagem enxuta para o planejamento de projetos pode eliminar grande parte desses resíduos diz (GALE, 2013).

Com o intuito de construir organizações cada vez mais eficientes e eficazes, este artigo tem por objetivo identificar os desperdícios no gerenciamento ágil de projetos de desenvolvimento de *softwares*, a fim de compreender se os princípios do *Lean Thinking* podem ser empregados na organização, possibilitando entregar mais valor para seus clientes. A experiência aqui relatada foi vivenciada em uma empresa de tecnologia da informação.

O presente trabalho está dividido em 6 (seis) seções sendo a primeira uma contextualização do tema com os objetivos de pesquisa. A segunda seção apresenta uma revisão da literatura sobre o gerenciamento ágil de projetos, o método *scrum* e o pensamento enxuto como alternativa para redução de desperdícios. A terceira e quarta seção expõem, respectivamente, a classificação e o método de pesquisa proposto, em seguida é feita apresentação da empresa e análise de resultados dos projetos, objetos de estudo. A quinta seção apresenta a discussão dos resultados levantados na análise. E por fim, a sexta seção se refere às considerações finais e sugestões de melhorias para os problemas identificados.

## 2 Referencial teórico

### 2.1 Abordagem ágil no gerenciamento de projeto



A abordagem tradicional e modelos de gestão de projetos ainda estão profundamente incorporados na maneira como gerenciamos tudo, incluindo operações, atendimento ao cliente, orçamento, governança e estratégia. Desse modo, a abordagem ágil se mostrou como uma solução capaz de quebrar paradigmas, porém deve ser conectada e apresentada de forma sistemática para toda organização (HUMBLE; MOLESKY; O'REILLY, 2015).

O método ágil busca encurtar processos e eliminar retrabalho por meio da evolução do processo baseado em ciclos com pequenas etapas para entrega, proporcionando benefícios mais cedo e com maior frequência (KENETT, 2013). Com a liberdade oferecida pelo gerenciamento ágil, as “equipes ágeis de sucesso estão produzindo *software* de maior qualidade que atendem melhor às necessidades do usuário, com maior rapidez e a um custo menor do que equipes tradicionais” (COHN, 2011, p.25).

A aplicação dos métodos de gerenciamento ágil ganhou força a partir da criação dos princípios e práticas determinadas pelo “Manifesto para Desenvolvimento Ágil de *Softwares*”. Conhecido como “Manifesto ágil”, este documento foi resultado de uma convenção com a presença de dezessete pessoas influentes da comunidade do *Extreme Programming* (XP) que buscavam entender ações vivenciadas por profissionais da área e levantar as melhores práticas utilizadas no ambiente de trabalho de desenvolvimento de software (HIGHSMITH et al. 2001). Desse modo, o documento reúne um conjunto de princípios e valores essenciais para obter os melhores resultados no desenvolvimento de softwares (HIGHSMITH et al. 2001) que são: I. os indivíduos e interações mais que procedimentos e ferramentas; II. *software* em funcionamento mais que documentação abrangente; III. a colaboração com os clientes mais que negociação de contratos; IV. responder as mudanças mais que seguir um plano.

Ao longo da última década, as práticas de gestão, sob vários rótulos como *Extreme Programming* (XP), *Scrum*, *Kanban* e *Lean*, foram testadas em campo e comprovadas em milhares de organizações ao redor do mundo. As abordagens *Scrum* e *Lean* serão discutidas com maior ênfase neste estudo.

### 2.1.1 A abordagem Scrum

O *scrum* é um “*framework* dentro do qual, pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entrega produtos com o mais alto valor possível”, explica Schwaber e Sutherland (2016, p.3). Cruz (2013) diz que o *scrum* pode ser empregado em diversos processos ou técnicas, seja no desenvolvimento de qualquer produto.

Na abordagem *scrum* os projetos são divididos em ciclos repetitivos, que possibilita modificação, correção de desvios, aperfeiçoamento da previsibilidade e o controle de riscos (SCHWABER; SUTHERLAND, 2016). Esses ciclos são chamados de *sprints*, que envolve a “definição do que é para ser construído, um plano projetado e flexível que irá guiar a construção, o trabalho e o resultado do produto” (SCHWABER; SUTHERLAND, 2016, p.8).

O foco do *scrum* é controlar os processos empíricos, aqueles em que o conhecimento vem da experiência. O controle do processo empírico se apoia em três pilares: transparência, inspeção e adaptação. A transparência, diz respeito aos processos que devem estar visíveis aos responsáveis, a fim de garantir que todos compartilhem do mesmo entendimento do que está sendo visto. A inspeção deve ser efetuada, pelos funcionários, nos artefatos *scrum* e no progresso, para detectar possíveis variações. A adaptação determina se um ou mais aspectos do processo fugiu dos limites aceitáveis, ou seja, devem ser feitos os ajustes necessários, seja no processo ou no produto (SCHWABER; SUTHERLAND, 2016, p.4).

O time *scrum* deve ser composto por três papéis principais: o *scrummaster*, *product owner* e o time. O *scrummaster* é o responsável por garantir a execução do fluxo *scrum* pelo time, removendo os riscos que interfira no objetivo do projeto. O *product owner* é responsável



pelo gerenciamento do *backlog* do produto e visa garantir o valor do trabalho realizado pelo time. O time *scrum* é composto por desenvolvedores, *testers*, arquitetos, entre outros, que devem buscar a transformação do *backlog* do produto em incrementos de funcionalidades que serão entregues ao cliente (CRUZ, 2013).

O *scrum* determina quatro eventos formais dentro da *sprint*: a reunião de planejamento da *sprint*, reunião diária (*daily*), reunião de revisão da *sprint* (*sprint review*) e a retrospectiva da *sprint*. Na reunião de planejamento da *sprint* é planejado o trabalho que será realizado na *sprint* com a participação de todo time *scrum*. A *daily* é um evento de duração de 15 (quinze) minutos, para o time *scrum* sincronizar as atividades que serão executadas no dia. A *sprint review* acontece ao final de cada *sprint*, com a participação do time e dos *stakeholders*, para inspecionar a entrega (pontos positivos e negativos) e adaptar o *backlog* caso seja necessário. A retrospectiva da *sprint* ocorre depois da *sprint review*, com o objetivo de levantar como foi o trabalho em relação às pessoas, aos relacionamentos, aos processos, ferramentas e pontos de melhorias (SCHWABER; SUTHERLAND, 2016).

Para a realização desses eventos, são utilizados como apoio os artefatos, estes “representam o trabalho ou valor para o fornecimento de transparência e oportunidades para inspeção e adaptação” (SCHWABER; SUTHERLAND, 2016, p.13). Entre os artefatos *scrum* estão: I. histórias: descrição resumida e clara de uma funcionalidade; II. gráfico de *burndown*: representa visualmente a taxa estimada de trabalho e a taxa realizada de trabalho; III. *backlog*: todos os requisitos do produto que devem ser entregue (SCHWABER; SUTHERLAND, 2016).

Para obter uma implementação bem sucedida do *scrum*, a mudança organizacional deve ocorrer tanto de cima para baixo, quanto de baixo para cima. COHN (2011, p.32) diz que o esforço da transição para o ágil gera muitos benefícios como, o time se torna mais veloz, com uma produtividade mais alta, redução de custos e entregas com maior qualidade. Além disso, a satisfação e engajamento no trabalho aumentam devido às entregas acontecerem mais cedo e em números maiores, o que leva a ganhos de produtividade e consequentemente a satisfação dos *stakeholders*. É nesse contexto que se inicia um eficiente ciclo de melhoria contínua.

## 2.2 A filosofia *Lean Thinking*

“À medida que o ritmo da mudança social e tecnológica no mundo se acelera, a pioneira Toyota se torna cada vez mais importante porque estabelece uma estratégia comprovada para prosperar na incerteza através da mudança” (HUMBLE; MOLESKY; O'REILLY, 2014, p.9). A produção enxuta ou Sistema Toyota de Produção (STP), que teve início no Japão na década de 50 com a Toyota, se refere a um conjunto de práticas “cujo objetivo é melhorar continuamente o sistema produtivo por meio da eliminação das atividades que não agregam valor ao cliente, chamadas de desperdícios” (TUBINO, 2015, p.18).

Womack e Jones (1996) explicam que, essa abordagem visa melhor organizar e gerenciar o relacionamento da empresa com seus clientes e sua cadeia de fornecedores. Esse sistema significa essencialmente fazer mais com menos, eliminar desperdícios e criar um fluxo de valor ao longo de um processo de transformação (LIKER; MORGAN, 2008). A Toyota define desperdício como sendo tudo além do que é essencial para a produção, seja a quantidade de equipamento, itens e recursos humanos (SLACK; JOHNSTON; JONES, 2013).

Slack, Johnston e Jones (2013) classificam os desperdícios em sete tipos: I. produção excessiva de produtos; II. tempo de espera para utilização de um equipamento e no trabalho; III. transporte de itens que não agrega valor, devido a um arranjo físico inadequado nos métodos de transporte e organização do trabalho; IV. processos mal definidos e desnecessários; V. estoque de processamento ou para consumo; VI. movimentação das





pessoas; VII. defeitos ou falhas no produto/serviço.

Além da produção sem desperdícios, o *lean* visa atender a demanda instantaneamente, com qualidade perfeita. As atividades e fluxos de trabalho são definidos e adaptados de modo a atingir um nível mais alto de desempenho e aprimorar continuamente, o que leva a empresa buscar por inovação (SLACK; JOHNSTON; JONES, 2013).

De acordo com Shingo (1996), a capacidade de eliminar perdas está associada a uma atitude positiva constante de acreditar que há outras maneiras de executar eficientemente o processo. Para a Toyota sempre haverá outra maneira para executar uma dada tarefa e eliminar completamente o desperdício, pois este não agrega valor e precisa ser descartado.

Para que ocorra a implantação das práticas do *lean* e as melhorias nos processos, Liker e Convis (2013) apontam que, é necessário o comprometimento das organizações com a criação de uma cultura de melhoria contínua. As melhorias propostas por um programa *lean* devem ir além da implantação de ferramentas, de métodos aleatórios e melhorias pontuais, ele deve sustentar as mudanças na organização que sejam contínuas e vinculadas às metas de negócios (LIKER; CONVIS, 2013). O conceito de melhoria contínua é fundamental na filosofia enxuta, é conhecido como *kaizen*, que significa melhoramento contínuo e envolve todos, entre gerentes e trabalhadores. Implantar o *kaizen* é ter garantia de que a melhoria irá ocorrer (SLACK; JOHNSTON; JONES, 2013).

### 2.2.1 A visão de valor do *Lean Thinking*

É por meio das atividades executadas em uma cadeia de valor que a empresa define a estratégia da organização e entrega valor ao cliente. Para Woiller e Mathias (2008), o valor é definido pelo consumidor, quanto mais intangível o valor, mais a satisfação do cliente deve ser trabalhada. Ele é resultado de fontes como, utilidade do produto, imagem, posicionamento de marca e até mesmo a distribuição.

Entende-se por cadeia de valor, uma abordagem que define atividades que estão associadas à estratégia utilizada pela empresa para entregar valor. Leva em consideração uma análise de *trade off*, ou seja, uma escolha que deve estar associadas à estratégia de posicionamento da empresa (WOILER; MATHIAS, 2008).

O *Lean Thinking* é uma filosofia gerencial que visa especificar e agregar valor, definir processos cada vez mais eficazes e ágeis, por meio da eliminação do desperdício. Como conceito chave para o *Lean Thinking* o valor é a capacidade de entregar exatamente o produto ou serviços que um cliente deseja (WOMACK; JONES, 1996).

Womack e Jones (1996) definem 5 (cinco) princípios do pensamento enxuto, como a chave para criação de uma empresa enxuta, que são: I. especifique valor: o valor é definido pelo cliente final. Prover o valor que o cliente deseja; II. identifique a cadeia de valor: conjunto de ações necessárias para produzir um produto específico e eliminar etapas que geram desperdícios; III. fluxo de valor: fluxo de atividades que fazem com que o valor flua nos processos para criação de produtos; IV. produção puxada: definir a produção baseada no consumo dos clientes. Deixar que os clientes puxem o valor da empresa; V. perfeição: busca pela perfeição e melhoria contínua.

Os processos podem ser definidos entre os que agregam ou não valor. As tarefas que agregam valor são aquelas que estão ligadas diretamente para a criação de produtos ou serviços que o cliente deseja. As atividades que não agregam valor são as que não contribuem diretamente no processo, sendo classificadas como desperdícios. Os processos devem evoluir a fim de reduzir e evitar esses desperdícios. Quanto maior o valor agregado, maior a eficiência da operação (WOMACK; JONES, 1996).

## 3 Metodologia



Este artigo busca identificar as fontes de desperdícios no gerenciamento de projetos de desenvolvimento de softwares, a fim de compreender se os princípios do *Lean Thinking* podem ser empregados na organização, possibilitando entregar mais valor. A experiência aqui relatada foi vivenciada em uma empresa de tecnologia da informação.

Para atender ao objetivo proposto, optou-se pela abordagem qualitativa, por meio de estudo de caso, com o intuito de analisar e interpretar os “fenômenos individuais, organizacionais, sociais, políticos ou de grupo” (YIN, 2009, p.20), estabelecendo uma discussão sobre as variáveis de estudo, por meio de entrevistas e análise de documentos. Foram incorporadas subunidades de análises em um caso único, sendo possível desenvolver um projeto mais complexo, realçando o valor das impressões em um caso único (YIN, 2009, p.67). Possui caráter descritivo que se justifica, segundo Gil (2010), por descrever as características de uma população, de um fenômeno ou experiência, proporcionando uma nova visão sobre a realidade já existente. Na fase inicial da pesquisa, como procedimento técnico, realizou-se uma pesquisa bibliográfica dos aspectos relevantes a serem avaliados no objeto de estudo, combinada com estudo de caso e entrevista.

Yin (2009) explica que existem seis fontes para coletar evidências no estudo de caso: documentos, registros em arquivos, entrevistas, observação direta, observação participante e artefatos físicos. A entrevista é um dos mais úteis instrumentos de coleta de dados, porém devem-se tratar os resultados obtidos de maneira científica, eliminando subjetividade. A pesquisa teve como base entrevista com profissionais da área descrevendo como sua empresa aborda o desenvolvimento de projetos e, análise de documentos, respeitando o princípio de coletar dados a partir de várias fontes de evidências (YIN, 2009).

Ao conduzir um estudo de caso único, uma etapa fundamental é a de definir a unidade de análise, em que se devem tomar precauções a fim de assegurar que o caso é relevante para o tema (YIN, 2009). Desse modo, foi realizada uma triagem, por meio de questionamentos, sobre cada opção que atendessem questões de interesse, assegurando a identificação de casos adequados antes da coleta formal de dados (YIN, 2009). A pesquisa foi validada considerando os seguintes aspectos: a) empresa prestadora de serviço de tecnologia da informação; b) empresa que utiliza o *scrum* como abordagem para gerenciamento de projetos; c) acesso a profissionais e a disponibilidade de informações.

Respeitou uma lógica que permitisse um diagnóstico mais objetivo para o desenvolvimento dos projetos: (a) 2 (dois) projetos internos e (b) 2 (dois) projetos externos, sendo que todos eles utilizaram o *scrum* para sua execução. Essa escolha se deu com o objetivo de comparar a metodologia utilizada entre projetos para o ambiente interno da organização e projetos executados para clientes externos, e assim avaliar a eficiência de um em relação ao outro.

Na segunda fase, para a coleta de dados, o instrumento de pesquisa foi um roteiro de perguntas abertas, possibilitando focar diretamente nas variáveis do estudo de caso e captar inferências causais. As entrevistas foram realizadas diretamente pelo pesquisador junto aos profissionais em seus respectivos locais de trabalho. Os profissionais escolhidos foram um *scrummasters* e um desenvolvedor para cada projeto escolhido, sendo gravadas, transcritas e analisadas, permitindo associar com maior riqueza de detalhes os aspectos levantados com as variáveis de estudo. A análise documental contribuiu para validação e comprovação dos dados obtidos nas entrevistas.

As variáveis de análise estabelecida para a pesquisa se fundamentaram nas principais fontes de desperdícios do Sistema Toyota de Produção aplicado ao desenvolvimento de softwares, segundo Poppendieck e Poppendieck (2008), conforme descrito no Quadro 1.



Sistema Toyota de Produção	Desenvolvimento de Softwares
Superprodução	Funcionalidades a mais
Espera	Atrasos
Transporte	Transferência de Controle
Processamento adicional	Reaprendizagem
Estoques no processo	Trabalho inacabado
Movimentação	Troca de tarefas
Defeitos	Defeitos

**Quadro 1. Os sete desperdícios no desenvolvimento de softwares**

Fonte: Poppendieck e Poppendieck (2008)

Segundo Poppendieck e Poppendieck (2008), o foco da eliminação do desperdício está em reduzir os custos da organização e tornar os produtos mais efetivos. Eles traduzem os sete desperdícios conforme a seguir: I. **trabalho inacabado**: o fluxo de trabalho deve ser único e rápido, para isso ele deve ser dividido em pequenos lotes ou iterações; II. **funcionalidades extras**: uma funcionalidade só deve ser desenvolvida se houver uma necessidade econômica clara e atual; III. **reaprendizagem**: é fundamental ter uma gestão do conhecimento de todos trabalhadores, aproveitando a experiência de cada um; IV. **transferência de controle**: ao transferirmos o controle de um trabalho para outra pessoa, ocorre uma grande perda de conhecimento tácito, que fica apenas na mente do criador; V. **troca de tarefas**: não se devem realizar várias tarefas ao mesmo tempo, e sim uma por vez; VI. **atrasos**: mau planejamento e espera por aprovações provocam muitos atrasos na entrega do projeto; VII. **defeitos**: os códigos devem ser a prova de falhas, caso aconteça deve ser identificado o defeito o mais cedo possível.

## 4 Análise de resultados

### 4.1 Apresentação da empresa

A empresa objeto de estudo, denominada Alfa, localiza-se em Belo Horizonte, Minas Gerais. Empresa do setor de tecnologia, fundada em 2009, apresenta como missão revolucionar a experiência de engajamento entre marcas e pessoas, fortalecendo o relacionamento com clientes. O produto *core business* da empresa é uma plataforma de automação de *marketing*.

A empresa Alfa possui três núcleos de gerenciamento de projetos: o núcleo de dados, núcleo de produtos e núcleo de projetos especiais. O “núcleo de dados” é responsável pelo suporte a todos os projetos que envolvam a necessidade de tratamento e manipulação de uma massa de dados. O “núcleo de produtos” é responsável por projetos referente a produtos do portfólio, ou seja, que envolvam a plataforma de automação de marketing e projetos internos de melhoria da plataforma. O “núcleo de projetos especiais” trabalha com projetos externos à plataforma. Para o estudo de caso, foram selecionados 2 (dois) projetos do “núcleo de produtos” e 2 (dois) projetos do “núcleo de projetos especiais”.

### 4.2 Apresentação dos projetos





## 4.2.1 Análise do Projeto A

O **projeto A**, pertencente ao “núcleo de projetos especiais”, surgiu da necessidade do cliente de reformular a regra do programa de pontos do serviço. O programa de pontos passaria a funcionar conforme a seguir: os usuários acumulam pontos de acordo com a utilização do serviço durante 90 (noventa) dias, ao final desse período os 100 (cem) mil primeiros colocados recebem benefícios e a pontuação é zerada, iniciando uma nova rodada. Uma vez que a cada 3 (três) meses a pontuação é zerada, a mudança para esse nova regra só poderia ser implementada ao final de uma rodada.

Neste projeto não foi utilizada nenhuma ferramenta de gerenciamento, porém foi criado em uma planilha com o acompanhamento das entregas por meio do Gráfico de *Burndown*, para monitorar o progresso do time. O cliente disponibiliza uma plataforma que serve para registrar as *sprints*, reportar problemas e marcar datas de entregas.

Na fase de **planejamento**, para a elaboração da versão final do escopo foi realizada uma reunião entre a agência de projetos, de criação e empresa Alfa, para discutir limitações e requisitos do projeto. Em seguida, o *layout* foi elaborado e aprovado antes do início da criação da documentação técnica do projeto, uma vez validado e aprovado o *layout*, não se pode alterar as telas. Dessa forma foi possível realizar o **planejamento da sprint** e iniciar o desenvolvimento com todos os recursos necessários para realização do trabalho.

Durante a fase de **desenvolvimento** do projeto A, houve **transferência de controle** de houve uma troca de funções no projeto, e o repasse das atividades se deram de maneira gradual e direta, sem a necessidade de criar documentos de instruções sobre o projeto. Já a **troca de tarefas** ocorreu constantemente, devido à necessidade de suporte e manutenção.

As entrevistas realizadas mostraram que a empresa analisada busca sempre surpreender o cliente, buscando desenvolver funcionalidades e tecnologias surpresas. Ao analisar as informações coletadas do projeto A, observou-se que não foram criadas **funcionalidades extras**, porém foram executadas mudanças no projeto, de modo a propor melhor solução do que a prevista.

Os códigos deste projeto foram criados, testados e aceitos. Todo código foi testado e validado, não deixando passar nenhum **defeito**, visto que o projeto impactaria um grande número de usuários e financeiro para o cliente. Os testes foram executados em dois momentos: teste no ambiente do desenvolvedor (teste local) e teste no ambiente de homologação (testes da aplicação). O teste local pode ser realizado de duas formas, manual ou automatizado, e no caso do projeto A, optou-se apenas pelo teste manual.

Os entrevistados afirmaram a buscam criar códigos mais limpos, utilizar de fácil compreensão, permitindo que qualquer desenvolvedor possa interpretá-lo sem a necessidade de documentação, apenas em casos de se criar novas tecnologias é que se documenta o código. Para avaliar se o código está limpo ou não, é utilizado um *software* que determina a qualidade em uma escala de A a F, sendo A o código mais limpo.

Os entrevistados consideram que foram encontrados poucos **defeitos**, visto sua complexidade. Os **defeitos** que ocorreram com maior frequência foram: funcionamento incorreto do *ranking* de pontos e incompatibilidade da linguagem desenvolvida com o sistema utilizado pelo cliente. O novo sistema pontos foi implantado com todos os defeitos corrigidos.

Foram realizadas reuniões de *daily meeting*, para discutir sobre as tarefas do dia anterior e as que estão previstas. Já a reunião de retrospectiva ocorreu apenas ao final do projeto e, não foi documentada. não garantindo a **gestão do conhecimento** e das informações.

Por fim, é importante ressaltar que o serviço já estava pronto para atualizar as mudanças do *website*, no entanto, em função de atividades que deveriam ser executadas pelo cliente e não foram, acarretou **atraso** no projeto. Após a entrega final, a equipe apresentou o resultado para todos os funcionários da empresa Alfa.



#### 4.2.2 Análise do projeto B

O **projeto B**, pertencente ao “núcleo de projetos especiais”, surgiu da necessidade da empresa Alfa em diminuir o número de chamados diários para resolução de problemas com usuários da *website* do programa de pontos. Quando um usuário entra em contato com a central de atendimento do cliente, a equipe de atendimento do cliente deve abrir um chamado para resolução do problema. O número de chamados era alto, sendo que muitas vezes podiam ser tratadas de maneira mais rápida e simples pela equipe de atendimento. Desse modo, o **projeto B** teve como objetivo a melhoria do sistema de suporte e abertura de chamados, adicionado à plataforma, um sistema refinado de reconhecimento de problemas, que identifica a causa raiz do mesmo, direcionando o atendente sobre como proceder.

Neste projeto não se utilizou nenhuma ferramenta para seu gerenciamento, desse modo não foi possível acompanhar o progresso do projeto, além de não ter dados para alimentar nenhum indicador de desempenho. Foi utilizada apenas uma plataforma do cliente para registrar as entregas e datas das sprints.

O **planejamento** do projeto B foi elaborado pela agência de projetos com o suporte da empresa Alfa, que criou o fluxo dos possíveis problemas juntamente com o cliente. Ocorreu **atraso** de 1 (um) mês para o início do desenvolvimento do projeto, devido à demora no processo de aprovação do escopo do projeto pelo cliente. Após a aprovação, a criação do layout e a documentação técnica do projeto necessitaram ser elaborada em paralelo.

O objetivo e os requisitos do projeto foram bem definidos e estruturados, e as regras do serviço do cliente já estavam determinadas por ele, desse modo o time *scrum* desenvolveu exatamente o que estava previsto, não sendo criadas **funcionalidades a mais**.

Durante a fase de **desenvolvimento**, não ocorreu **transferência de controle**. Já a **troca de tarefas** ocorreu em vários momentos, pois a equipe também estava envolvida em outras demandas de manutenção e suporte.

Os entrevistados explicaram que no projeto B, os códigos foram criados e em torno de 50% dos códigos foram documentados. Devido ao pouco tempo para a execução de testes, a documentação de teste não foi elaborada e nem todos os códigos foram testados. Foram realizados apenas alguns testes manuais e da funcionalidade. Devido à urgência do projeto, o cliente decidiu lançar sem teste final. Os **defeitos** encontrados não eram de alto impacto, e foram todos corrigidos aos poucos. O que mais ocorreu foi erro de compatibilidade do código com o navegador do cliente.

Foram realizadas reuniões de *daily meeting* para alinhar o andamento das tarefas. Já as reuniões de retrospectiva e *review* não foram realizadas, impossibilitando à **gestão do conhecimento e da informação**. Os entrevistados alegaram que não foi possível devido ao tempo curto para execução, tornando inviável alocar tempo para essa reunião.

Os entrevistados afirmaram ter ocorrido **atraso** por parte da equipe técnica. O principal motivo do atraso foi a falta de experiência e falta de conhecimento de erros passados dos desenvolvedores. Dessa forma, a equipe teve de fazer ajustes, em um curto tempo para conseguir fazer a entrega do projeto.

#### 4.2.3 Análise do projeto C

O **projeto C**, do “núcleo de produtos”, surgiu a partir de uma necessidade interna de melhoria da plataforma. Teve por objetivo criar um ambiente de campanhas de *marketing*, em que o cliente pudesse criar réguas de relacionamento com o usuário, com o intuito de agregar valor ao cliente, oferecendo a ele sua própria autonomia para operar a criação de campanhas.

O “núcleo de produtos” utiliza algumas ferramentas para o gerenciamento do projeto:



o *Jira*, para acompanhamento e indicadores de desempenho do projeto; o *Trello*, para listagem de requisitos; plataforma desenvolvida internamente para gerenciamento das *sprints*, avaliação da *performance* de cada membro do time, gerenciamento das reuniões de revisão, além de medir a duração e complexidade do projeto.

A idealização do projeto durou em torno de 8 (oito) meses, essa demora ocorreu devido à falta de foco e definição do que se esperava do produto final. O projeto começou a ser desenvolvido sem uma definição clara do objetivo, os requisitos da nova funcionalidade e principalmente sem ter levantado as reais necessidades do cliente.

Durante a fase de **desenvolvimento**, não ocorreu **transferência de controle**. A **troca de tarefas** ocorreu apenas com o *scrummaster*, devido a sua maior experiência na área. As equipes do núcleo de produtos são fixas, potencializando a redução de troca de tarefas.

Neste projeto, foram criadas muitas **funcionalidades extras**, em decorrência da falta de planejamento do projeto, acarretando em **retrabalho** e **atrasos** na entrega. Para corrigir o atraso, a equipe passou a realizar reuniões semanais para discutir e definir próximos passos.

Os códigos foram criados e não documentados, já a documentação de teste foi parcialmente elaborada. Nesse projeto, foi realizado somente o teste manual, visto que a funcionalidade ainda não foi lançada.

Para os entrevistados, a frequência da ocorrência de **defeitos** está dentro do esperado. As falhas que mais ocorreram estão relacionadas à falta de experiência da equipe, desenvolver a funcionalidade sem entendimento e a falta de conhecimento de erros passados. O **trabalho inacabado** esteve presente em vários aspectos desse projeto, incluindo documentação do projeto que não foi criada, códigos que não foram documentados e a constante mudança de requisitos, fazendo com que várias partes do trabalho realizado fossem descartadas.

Foram realizadas reuniões de *daily meeting*. Já as reuniões de retrospectiva da *sprint* e de *review* foram executadas no mesmo momento, sendo ao final de cada *sprint* com o objetivo de avaliar como foi a execução do projeto e pontuar aprendizados. Todas essas reuniões foram documentadas na plataforma interna para acompanhar todo o progresso das *sprints* e evolução das equipes, contribuindo para a **gestão do conhecimento**.

Essa versão inicial foi encaminhada para teste, para alguns clientes, após 1 (um) ano do desenvolvimento do projeto, porém não foram recolhidos *feedbacks*, paralisando o projeto.

#### 4.2.4 Análise do projeto D

O projeto D do “núcleo de produtos” surgiu a partir de solicitações de clientes que tinham a necessidade de obter informações mais detalhadas de suas campanhas. Desse modo, teve por objetivo criar tela de *report* para o cliente, em que pudesse ter visão de todos os dados e manipulá-los para analisar informações específicas dos usuários.

O projeto D utilizou as seguintes ferramentas para o gerenciamento de projetos: *Jira*, *Trello* e plataforma interna para gerenciamento de projetos. Desse modo, foi possível acompanhar o projeto por meio de indicadores de desempenho.

O projeto começou a ser desenvolvido sem a definição clara do objetivo, sem fazer um **planejamento** adequado e principalmente sem ter levantado as informações que os clientes precisam extrair do relatório. O planejamento da *sprint* foi realizado sem saber ao certo onde se queria chegar, levando ao **descarte** de toda a produção. Desse modo foram desenvolvidas muitas **funcionalidades a mais**, várias telas de *report* foram criadas e descartadas.

Foi necessário reiniciar o projeto, após 5 (cinco) meses do seu início, e então planejando bem suas ações. O projeto tinha previsão de duração de 4 (quatro) meses, porém devido ao surgimento de novas prioridades e outros **atrasos**, foi entregue após um 1 (um) ano. Nesse momento, passaram a ser realizadas reuniões de *daily meeting*, reuniões de retrospectiva e *sprint review*, sendo todas documentadas na plataforma interna para



acompanhamento do progresso das *sprints* e evolução da *performance* das equipes.

Durante a fase de desenvolvimento, não ocorreu **transferência de controle**, entre os membros do time. Já a **troca de tarefas** ocorreu com maior frequência com o *Scrummaster*, devido a sua maior experiência, acabava por interromper uma atividade para se dedicar a tarefas mais prioritárias, não mantendo foco.

O **trabalho inacabado** esteve presente em vários aspectos desse projeto, incluindo documentação do projeto que não foi criada, códigos que não foram documentados e a constante mudança de requisitos, que levou ao descarte de várias entregas do projeto.

Os códigos foram criados e não documentados, já a documentação de teste foi elaborada. Para os entrevistados, o número de **defeitos** encontrados estava na média dos projetos da empresa. As falhas que mais ocorreram estão relacionadas à falta de clareza no planejamento, desenvolver a funcionalidade sem entendimento e a falta de conhecimento de erros passados.

Essa versão foi encaminhada para teste com alguns clientes. Em seguida foram recolhidos alguns *feedbacks* e implantadas melhorias, porém não foi publicado para todos os clientes na plataforma.

## 5 Discussão de resultados

Nesta seção são apresentados os resultados obtidos no que tange a identificação dos desperdícios durante o desenvolvimento de projetos de software utilizando o *scrum* como método ágil. Os fatos descritos pelos entrevistados evidenciaram que a utilização do *scrum* nem sempre ocorre como prescreve a teoria. Na prática, vários processos, entregas e padrões de qualidade exigidos não são cumpridos, contribuindo para a geração de desperdícios.

Na **empresa Alfa** a realização da prática dos 4 (quatro) eventos do *scrum*, conforme determina Schwaber e Sutherland (2016), se mostrou parcialmente presente no cotidiano dos projetos. O líder da equipe realiza o **planejamento da sprint** com os responsáveis, seja interno ou externo, sobre o que deverá ser entregue, em seguida repassa as decisões para o time *scrum* em outra reunião, com o objetivo de divulgar o objetivo do projeto, o que deve ser entregue, quem fará cada tarefa e os resultados esperados.

Durante o **desenvolvimento** foram realizadas as *dailys meeting* para alinhamento e sincronização das atividades que serão executadas no dia. Percebeu-se que as reuniões de *sprint review* e retrospectiva da *sprint* são realizadas em conjunto, às vezes ao final de cada *sprint* ou ao final do projeto, de acordo com sua duração. Essa prática é empregada erroneamente, visto que as duas reuniões possuem objetivos distintos. Segundo Cruz (2013), a reunião de *sprint review* deve acontecer antes da retrospectiva, e tem por objetivo revisar os itens concluídos do *backlog*, analisando o que está sendo entregue e o que deveria, de fato, ter sido entregue. A reunião de retrospectiva, tem por objetivo avaliar o que funcionou bem e o que pode ser melhorado, em relação a pessoas, processos e ferramentas. Para os **projetos internos**, a reunião aconteceu ao final de cada *sprint* e todas elas foram documentadas, o que possibilita a gestão da informação e do conhecimento, e a redução de trabalhos por **reaprendizagem**. Já para os **projetos externos**, não é uma prática muito utilizada.

Toda funcionalidade criada deve ser justificada, “no mundo do produto, lançar um produto com somente as funcionalidades corretas, nem mais nem menos, demonstra que a empresa realmente compreende o que os clientes desejam” (POPPENDIECK; POPPENDIECK, 2008, p. 90). No caso dos **projetos externos**, ficou evidenciado que a empresa realmente compreende o que o cliente deseja, entregando o que foi solicitado, agregando seu valor ao produto e superando suas expectativas. Já no caso dos **projetos internos**, percebe-se um alto desperdício com criação de **funcionalidades a mais** e *sprints* reformuladas devido à má definição de requisitos. Cabe ressaltar que essas soluções de





melhoria foram propostas sem ouvir o cliente e entender sua real necessidade, indo contra o que afirma Poppendieck e Poppendieck (2008, p.95): “se não existe uma necessidade econômica clara e atual para uma funcionalidade, ela não deve ser desenvolvida”.

Os dados mostraram que grande parte dos **atrasos** ocorreu devido a falta de planejamento adequado, priorização de outros projetos, correção de falhas e retrabalho. Nos **projetos externos**, percebe-se uma dependência grande de validações do cliente para seu andamento. Nos **projetos internos**, as maiores fontes de atrasos foram objetivos mal definidos e falta de planejamento, o que também levou a grandes mudanças de requisitos. A empresa deve saber lidar neste contexto de alta variação de requisitos e alto grau de imprevisibilidade, porém quando processos não são coordenados e sua capacidade não é dimensionada adequadamente durante o **planejamento**, a programação inicial será irrealista e propensa a atrasos, e estes devem ser tratados como desperdícios (OEHMEN; REBENTISCH, 2010).

A empresa Alfa aloca pouco tempo para execução de processos de preservação de conhecimento gerado no projeto, levando em muitos casos a necessidade de **reaprendizagem**. A empresa possui uma plataforma interna onde são arquivadas todas as reuniões de *reviews* dos **projetos internos**, com os registros dos pontos positivos e de melhorias. Nos **projetos externos**, as reuniões não foram documentadas, perdendo pontos importantes de gestão do conhecimento. Muitos erros cometidos no desenvolvimento dos projetos foram erros já discutidos em projetos anteriores. Kennet (2013) argumenta que a gestão do conhecimento é crítica, desse modo a disseminação de uma cultura baseada em melhores práticas entre equipes se faz necessária, a fim de solucionar problemas de captura de padrões de processo, garantindo que as necessidades de reutilização de conhecimento ocorram.

A **transferência de controle** é o desperdício que menos ocorreu nos projetos estudados. Os métodos ágeis não orientam que ocorra a substituição de pessoas no desenvolvimento de *software*, isto por que no trabalho de desenvolvimento de software a competência pessoal e criatividade são muito importantes. No caso em que ocorreu a transferência de controle, esta se deu por meio do repasse de instruções de forma direta, que é o ideal, visto que a transferência de controle sempre resulta em perda de conhecimento, a melhor maneira de se repassar o conhecimento é por meio da orientação, troca de experiências e comunicação face a face (POPPENDIECK; POPPENDIECK, 2008).

Poppendieck e Poppendieck (2008, p.97) dizem que a **troca de tarefa** “não somente gera distração, como toma tempo, e frequentemente, pioram os resultados de ambas as tarefas”. Esse desperdício ocorreu com maior frequência nos **projetos externos**, visto que a equipe possui demandas diárias desse mesmo cliente de manutenção e suporte. No caso dos **projetos internos**, observou-se que a troca de tarefas acontecia com maior frequência com o *Scrummaster*, pois este detém mais experiência para lidar com diversas demandas.

Aspectos relacionados ao **trabalho inacabado** demonstram maior ocorrência em **projetos internos**. Entende-se que esse desperdício está ligado a fatores como a complexidade, gravidade, impacto e urgência para entrega. Esse desperdício pode ser identificado no código não testado, código não documentado, inexistência de escopo de projeto, falta de indicadores de desempenho, entre outros.

No que se refere aos testes, “toda base de código deve incluir um conjunto de testes à prova de falha que não permita defeitos no código, tanto em nível de unidade quanto em nível de aceitação” explicam Poppendieck e Poppendieck (2008, p.100). A empresa Alfa possui um fluxo de criação de código e teste bem formulado. Nos **projetos internos**, os testes foram escritos, porém os códigos não foram documentados. Já nos **projetos externos**, a documentação de teste não foi elaborada, e apenas códigos que envolviam novas tecnologias foram documentados. Segundo Poppendieck e Poppendieck (2008, p.100) “escrever testes de unidade antes de escrever o código implica um código mais limpo, mais compreensível e mais



testável”, o que não aconteceu em todos os casos.

Percebe-se que a realização do método correto de testes leva em consideração o tempo de execução do projeto, a qualidade exigida do produto e o impacto gerado para a organização. No **projeto externo A**, que envolviam um número grande de pessoas afetadas, todos os processos de teste aconteceram corretamente e nenhum *bug* foi para produção. Já no **projeto externo B** devido à necessidade de uma solução rápida, o cliente optou por não esperar pela finalização dos testes. Da mesma forma, nos **projetos internos** essas práticas não foram respeitadas.

A empresa Alfa afirma manter a criação de uma base de códigos simples, limpa e pequena, pois assim reduzem os **defeitos** e o retrabalho, aumentando a qualidade do código escrito, o que é reafirmado por Poppendieck e Poppendieck (2008, p. 89) “um código complexo é frágil e se quebra facilmente, tornando quase impossível modificá-lo de forma segura”. Para a empresa, assim que um **defeito** é encontrado ele deve ser corrigido, o que comprova ser uma boa prática realizada, visto que se deve “testar o código mais cedo e mais frequentemente para encontrar tantas dessas falhas inesperadas quantas forem possíveis” (Poppendieck; Poppendieck, p.100). Nos **projetos externos** há uma maior preocupação com a existência de falhas, visto que o serviço envolve milhares de usuários. Para os **projetos internos**, os defeitos apareceram principalmente pela falta de definição do objetivo e clareza no que precisava ser desenvolvido. Além disso, é comum entre todos os projetos a ocorrência de defeitos repetidos e falta de entendimento do que é a funcionalidade.

Como resultado, foi possível identificar desperdícios em todas as etapas do desenvolvimento de software da empresa, conforme o Quadro 2.

Desperdícios identificados nos projetos objetos de estudo					
Desperdícios	Etapas	Projeto A	Projeto B	Projeto C	Projeto D
Funcionalidades a mais	Desenvolvimento	Não ocorreu. Porém foi desenvolvida uma solução melhor do que esperada.	Não ocorreu. Teve objetivo claro e definido.	Sim. Devido a falta de definição clara do objetivo e escopo do projeto.	Sim. Devido a falta de definição clara do objetivo e escopo do projeto.
Atrasos	Planejamento/ Criação do layout/ Desenvolvimento	Sim. Atraso por parte do cliente devido à falta de priorização interna.	Sim. Atraso na aprovação do cliente; Atraso na entrega final devido a falhas.	Sim. Devido a falta de priorização e entendimento do que tinha que ser entregue.	Sim. Devido a falta de priorização e entendimento do que tinha que ser entregue.
Transferência de Controle	Desenvolvimento	Ocorreu apenas uma vez.	Não ocorreu.	Não ocorreu.	Não ocorreu.
Reaprendizagem	Planejamento/ Desenvolvimento	Ocorreram mesmos erros que em outros projetos.	Ocorreram mesmos erros que em outros projetos.	Ocorreram mesmos erros que em outros projetos.	Ocorreram mesmos erros que em outros projetos.
Trabalho inacabado	Planejamento/ Desenvolvimento	Sim. Código não documentado.	Sim. Código parcialmente documentado; Código não testado.	Sim. Código parcialmente documentado; Código não testado; Requisitos instáveis.	Sim. Código parcialmente documentado; Código não testado; Requisitos instáveis.
Troca de tarefas	Todo etapas	Sim. Em vários momentos	Sim. Em vários momentos	Sim. Maior frequência com o líder do projeto	Sim. Maior frequência com o líder do projeto
Defeitos	Planejamento/ Desenvolvimento / Implementação	Sim. Poucos bugs, considerando a complexidade do projeto.	Sim. Foi com bugs para produção; N° de bugs na média.	Sim. Foi com bugs para teste com cliente; N° de bugs na média.	Sim. Foi com bugs para teste com cliente; N° de bugs na média.

Quadro 2. Desperdícios nos projetos da empresa analisada

Fonte: elaborado pela autora

Percebe-se uma relação entre o modo como o projeto de desenvolvimento de softwares é gerenciado e os desperdícios em cada etapa. Dessa forma, adotar medidas para redução dos desperdícios em seu ambiente poderá tornar a empresa mais competitiva, e elevar seus



resultados em termos de tempo, desempenho, custo e aprovação dos *stakeholders* (OEHMEN et al., 2014).

## 6 Considerações finais

Com o presente estudo foi possível identificar as fontes de desperdícios que mais ocorreram e em qual etapa do desenvolvimento de softwares. No cenário em questão, considera-se que é possível adotar princípios *Lean Thinking* visando a diminuição das fontes de desperdícios, possibilitando entregar mais valor para seus clientes, além de garantir vantagem em agilidade no gerenciamento de projetos.

A falta ou a falha no **planejamento** acarretam em desperdícios desde o início do processo (KENNET, 2013). Os dados evidenciam que em **projetos internos** são dedicadas poucas horas em seu planejamento inicial, criando-se objetivos mal estabelecidos, que leva ao **retrabalho**. Nos **projetos externos**, o escopo vem pronto da agência de projeto, o que faz com que o objetivo e os requisitos estejam bem definidos.

As melhores práticas executadas pelas equipes ágeis devem ser formalizadas por meio de um mapeamento dos processos *scrum*, para torná-la conhecida para todo o time, promovendo maior integração entre ferramentas e *softwares* utilizados, a fim de se obter um fluxo rápido e contínuo. Desse modo, destaca-se que a empresa analisada possui processos dentro do que se espera de uma abordagem ágil, porém, na prática, esses processos não ocorrem ou não são claramente entendidos sobre sua utilização, levando a desperdícios e custos para a organização. Quanto mais distanciado do método, maior será o desperdício e menor será o valor gerado, o que se confirma nos resultados dos projetos internos, visto que possuíam grandes desperdícios, falta de foco e alto retrabalho.

A divergência na execução de projetos internos e externos ficou evidenciada. Para projetos internos, não existe uma prática para avaliar sua relevância, benefícios futuros e retorno financeiro, mesmo sabendo que impactam em melhorias significativas para o produto da empresa. Estes projetos também não definiram com clareza seus objetivos, e principalmente, não buscaram ouvir as reais necessidades dos clientes. Para projetos externos, percebeu-se maior adequação aos processos e principalmente, ouvir e entender bem as necessidades dos clientes, o que se tornou diferencial para obtenção de resultados positivos.

Para determinar como a empresa adiciona valor para o cliente e o negócio, sugere-se a utilização de facilitadores enxutos, conhecidos como *lean enablers*. Esses facilitadores enxutos contribuem para que a empresa identifique seu fluxo de valor e entenda onde pode atuar de maneira mais enxuta, tornando-se mais assertiva e competitiva no mercado. De acordo com Oehmen et al. (2012), podem ser combinados e integrados com o gerenciamento de projetos, contribuindo para a redução de desperdícios e garantindo melhores resultados. As categorias dos *lean enablers* estão associadas aos seis princípios *Lean*: I. facilitadores *Lean* para tratar pessoas como seu ativo mais importante (princípio *Lean* 6); II. Facilitadores *Lean* para maximizar o valor de programas e projetos (princípio *Lean* 1); III. Facilitadores *Lean* para otimizar o fluxo de valor (princípio *Lean* 2); IV. Facilitadores *Lean* para criar o fluxo de programas (princípio *Lean* 3); V. facilitadores *Lean* para criar a produção puxada no programa (princípio *Lean* 4); VI. Facilitadores *Lean* para perseguir a perfeição no programa (princípio *Lean* 5). Exemplo mais utilizados e recomendados pelas organizações são segundo Oehmen et al. (2012): construir uma cultura de programa baseada no respeito pelas pessoas; envolver as partes interessadas ao longo do ciclo de vida do projeto; desenvolver plano de comunicação; designar uma função de gerente de projeto para liderar e integrar do início ao fim; gerenciar proativamente a incerteza e o risco para maximizar seus benefícios; dar uma visão concreta dos objetivos a serem alcançados.

A adoção de técnicas e diretrizes para o desenvolvimento de uma cultura ágil pode ser



considerada um importante avanço para a empresa. Porém, considerando-se os dados disponíveis, puderam ser feitas apenas constatações importantes sobre a associação dos desperdícios a cada fase do desenvolvimento de software, advindas da análise descritiva dos dados. Desse modo recomenda-se uma pesquisa mais profunda, visando a obtenção de dados mais concretos dos processos da organização e a causa raiz dos principais problemas. Pesquisas futuras podem ser conduzidas tendo como foco a investigação da cadeia de valor da empresa, o mapeamento do fluxo de valor e a aplicação de facilitadores enxutos com o objetivo de disseminar a cultura ágil e seus princípios, visando obter melhores resultados.

## 7 Referências

COHN, Mike. **Desenvolvimento de software com Scrum**. Porto Alegre: Bookman, 2011.

CONFORTO, Edivandro Carlos. **Modelo e ferramenta para avaliação da agilidade no gerenciamento de projetos**. 2013. 376 f. Tese (Doutorado). Programa de doutorado em Engenharia de Produção, Universidade de São Carlos, São Carlos, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18156/tde-24042013-143146/pt-br.php>>. Acesso em: 15 mar. 2017.

CRUZ, Fabio. **Scrum e PMBOK unidos no Gerenciamento de Projetos**. Rio de Janeiro: Brasport, 2013.

DENNING, Steve. The Best-Kept Management Secret On The Planet: Agile. **Forbes**, 2012, USA. Disponível em: <<https://www.forbes.com/sites/stevedenning/2012/04/09/the-best-kept-management-secret-on-the-planet-agile/#40ff6d1a5bc2>>. Acesso em: 03 abr. 2017.

EDER, Samuel. **Prática de gerenciamento de projetos de escopo e tempo nas perspectivas das abordagens ágil e tradicional**. 2012. 192 f. Dissertação (mestrado). Programa de Mestre em Engenharia de Produção, Universidade de São Paulo, São Carlos, 2012. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18156/tde-23092012-220004/pt-br.php>>. Acesso em: 20 abr. 2017.

GALE, Sarah Fister. Lean into savings: Using lean principles can drive waste and Inefficiencies out of programs — and save millions in the process. **PM Network**, v. 27, n.6, p.30-35, jun. 2013.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5.ed. São Paulo: Atlas, 2010.

HIGHSMITH, Jim; SUTHERLAND, Jeff; SCHWABER, Ken. *et. al.* **Manifesto for Agile Software Development**. 2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 04 mar. 2017.

HUMBLE, Jez; MOLESKY, Joanne; O'REILLY, Barry. **Lean Enterprise: how high performance organizations innovate at scale**. Sebastopol: O'Reilly Media, 2015.

KENETT, Ron S. Implementing scrum using business process management and pattern analysis methodologies. **Dynamic Relationships Management Journal**, New York, 2013. Disponível em: <<http://www.sam-d.si/Upload/Articles/DRMJv02n02a03.pdf>>. Acesso em: 20 mar. 2017.





LIKER, Jeffrey K; CONVIS, Gary L. **O modelo toyota de liderançalean**: como conquistar e manter a excelência pelo desenvolvimento de lideranças. Porto Alegre: Bookman, 2013.

LIKER, Jeffrey K; MORGAN, James M. **Sistema toyota de desenvolvimento de produto**: integrando pessoas, processo e tecnologia. Porto alegre: Bookman, 2008.

OEHMEN, Josefet al. **The guide to lean enablers for managing engineering programs**. Cambridge, MA: MIT-PMI-INCOSE. 2012. Disponível em: <<http://hdl.handle.net/1721.1/70495>>. Acesso em: 03 mar. 2017.

OEHMEN, Josefet *al.* Implemetation of Lean Engineering Practices in projects and programs through simulation based training. **PM World Journal**, v. 3, n. 3, mar. 2014. Disponível em: <[http://orbit.dtu.dk/fedora/objects/orbit:129577/datastreams/file\\_91e076c6-4a62-4adf-bc6e-818a71fbda61/content](http://orbit.dtu.dk/fedora/objects/orbit:129577/datastreams/file_91e076c6-4a62-4adf-bc6e-818a71fbda61/content)>. Acesso em: 10 jan. 2017.

OHNO, Taiichi. **O sistema Toyota de produção**: além da produção em larga escala. Porto Alegre: Bookman, 1997.

POPPENDIECK, Mary; POPPENDIECK, Tom. **Implementando o desenvolvimento Lean de software do conceito ao dinheiro**. Porto Alegre: Bookman, 2011.

SHINGO, Shigeo. **O sistema toyota de produção**: do ponto de vista da engenharia de produção. 2. ed. Porto Alegre: Bookman, 1996.

SLACK, Nigel; JONES, Alister Brandon; JOHNSTON, Robert. **Princípios de administração da produção**. São Paulo: Atlas, 2013.

SCHWAB, Klaus. **The fourth industrial revolution**. New York: Crown Business, 2017.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide**. 2016. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf>>. Acesso em: 25 mar. 2017.

TUBINO, Dalvio Ferrari. **Manufatura Enxuta como Estratégia de Produção**: A Chave para a Produtividade Industrial. Atlas, 2015.

VERSIONONE. 11th annualstateofagile report. Disponível em <<https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>>. Acesso em: 05 mai. 2017.

WOMACK, James P; JONES, Daniel T. **A mentalidade enxuta nas empresas**: elimine o desperdício e crie riqueza. 4.ed. Rio de Janeiro: Campus, 1998.

WOILER, Samsão; MATHIAS, Washington Franco. 2. ed. **Projetos**: planejamento, elaboração e análise. São Paulo: Atlas, 2008.

YIN, Robert K. **Estudo de caso**: planejamento e métodos. 3. ed. Porto Alegre: Bookman, 2005.