

Aplicação do Mapeamento de Fluxo de Valor para Identificação de Desperdícios Operacionais em uma Empresa de Software de Grande Porte

Felipe B. Ribeiro¹, Pedro Henrique de A. Machado¹, Jackson Gaspar Schimit,
Luana Belusso¹, Rafael A. Paes de Oliveira¹

¹Coordenadoria do Curso de Bacharelado em Engenharia de Software - Universidade Tecnológica Federal do Paraná (UTFPR)
Estr. p/ Boa Esperança, S/n - Zona Rural - 85.660-000 - Dois Vizinhos - PR - Brasil

felipebrenaribeiro@gmail.com, pedrohmachado@utfpr.edu.br

jacksonschimit@gmail.com, luanabelusso@alunos.utfpr.edu.br

raoliveira@utfpr.edu.br

Abstract. *Through the application of value stream mapping in a real large-scale software maintenance environment, information will be picked from the requests that the clients open with this company, and the teams and directions will be analyzed to identify if there is any operational waste in the resolution of the requests. After the analysis the results will be showed putting the operational wastes in evidence.*

Resumo. *Através da aplicação do mapeamento de fluxo de valor em um ambiente real de manutenção de um software de grande porte, será colhido informações das solicitações que os clientes abrem com a empresa, e analisado as equipes e direcionamentos para identificar se há algum desperdício operacional na resolução das solicitações. Após a análise os resultados serão apresentados evidenciando os desperdícios operacionais encontrados.*

1. Introdução

O Processo de Desenvolvimento de Software é uma área da Tecnologia da Informação (TI) que ao longo dos anos foi e continua se desenvolvendo de forma benéfica para as empresas, fornecendo regras, padrões e diretrizes para facilitar o desenvolvimento do software e todos os aspectos a ele relacionado. Hoje as empresas de software, chamadas de *Software Houses* (SH), em sua maior parte atuam no mercado com um software robusto e pronto para atender as necessidades dos clientes, mas também estão abertos a receber solicitações de melhoria do mesmo, com o objetivo de aperfeiçoá-lo continuamente. Caso haja algum processo interno dos clientes que o sistema não consegue atender, é solicitado a empresa do sistema que faça uma melhoria ou uma customização para que se adapte e tenha essa funcionalidade, beneficiando a todos. Esse processo de customização/melhoria recebe o nome de manutenção de software.

Ao aplicar um processo de desenvolvimento em uma fábrica de software, é comum que dificuldades e falhas sejam encontradas até que formas sejam levantadas para corrigi-las. Com a sociedade se tornando cada dia mais imediatista e carente de

softwares de qualidade, faz-se necessário técnicas para que todo tempo desperdiçado seja extinto do processo de produção para um maior destaque no mercado de trabalho. [Machado and Tretin 2017]

O Mapeamento de fluxo de valor (MFV) é um método de modelagem baseado na observação do fluxo de processos, materiais e equipes dentro da empresa, que são transformados em um mapa. Esse mapa sintetiza os fluxos atuais dos processos. Por fluxo de valor, entende-se o conjunto de todas as atividades que ocorrem desde a obtenção da matéria prima até a entrega do produto ao consumidor final. Assim sendo, surge a problemática deste artigo: Identificar as falhas e desperdícios no processo de manutenção de um software de grande porte por meio da aplicação de MFV.

2. Fundamentação Teórica

2.1. Processo de Desenvolvimento de Software

No contexto da Crise do Software de 1970 devido a falta de mão de obra qualificada, surge um conceito de padronização do desenvolvimento de software, os chamados Processos de Desenvolvimento de software, que tornam o desenvolvimento mais compreensível. A padronização de modelos de processo começou com os modelos genéricos, que compreendem 5 atividades: Comunicação, Planejamento, Modelagem, Construção e Entrega. Com esses passos estruturados, é possível orientar ao time de desenvolvimento a situação atual do projeto e qual direção deve ser seguida [Pressman 2011].

Em 1970, Winston W. Royce propôs o modelo de desenvolvimento cascata (Waterfall) para que os processos passassem a ser bem definidos. Esse modelo de desenvolvimento é rígido em relação aos seus processos, ou seja, só é possível avançar para o próximo passo do projeto se os anteriores foram concluídos com sucesso. Para a época foi um modelo que revolucionou a forma de desenvolvimento desde os seus passos iniciais até a entrega do software. As fases são: Engenharia de sistemas, análise, projeto, codificação, teste, implementação e manutenção [Hirama 2011].

Porém, como os requisitos da construção de um software são voláteis, o modelo cascata começou se tornar rígido demais para o resultado que os clientes desejavam. Logo, um novo modelo de processo de desenvolvimento foi apresentado ao mercado produtor de software, o modelo Espiral. Em 1988 Barry Boehm apresentou o modelo espiral, que consiste em um modelo mais flexível em relação as fases do projeto e que permitiam iteração para melhorias ou consertos de erros no escopo [Pressman 2011].

Mesmo considerando os benefícios que os modelos sequenciais e evolutivos trouxeram com o objetivo de amenizar o caos do desenvolvimento de software, eles ainda não focavam na flexibilidade de serem adaptados a todos os processos e formas de desenvolvimento que as empresas aplicam, e com essas lacunas e brechas nos processos anteriores, surgiram as metodologias de desenvolvimento ágil de software.

2.2. Desenvolvimento Ágil

O desenvolvimento ágil surgiu da necessidade de adaptar os modelos sequenciais ao processo de desenvolvimento de softwares complexos, que precisavam de modelos mais rápidos, que tivessem toda a documentação associada e que as entregas aos clientes fossem feitas de forma consistente. Assim, surgiram os modelos de desenvolvimento ágil, e os mais famosos foram o *Extreme Programming* e o *Scrum* [Beck and Gamma 2000].

Os processos ágeis ganharam espaço e credibilidade por revolucionar o foco da equipe de desenvolvimento, passando a focar muito mais na satisfação do cliente e no bem-estar da equipe ao invés de produção demasiada. O Manifesto Ágil que foi publicado em 2001 mostrou quatro pilares da forma ágil de desenvolvimento: Indivíduos e interações em vez de processos e ferramentas; Software executando em vez de documentação; Colaboração do cliente em vez de negociação de contratos; Respostas rápidas a mudanças em vez de seguir planos. [Poppendieck and Poppendieck 2003]

Embora ótimos resultados tenham sido obtidos a partir da aplicação dos modelos ágeis de desenvolvimento nos projetos de pequeno e médio porte, há um grande desafio ao aplicar essa metodologia para projetos de grande porte, pois há uma série de fatores que se alteram ao se tratar de um software maior. Dentre eles, podemos citar: Desafios para manter o teste contínuo; A manutenção do sistema aumentará, pois, várias versões serão disponibilizadas ao fim das *sprints*; Sobrecarga nas gerências que precisam coordenar várias equipes e suas comunicações ao mesmo tempo; Priorização de demanda é difícil de ser criada e mantida; Tempos ociosos no processo de desenvolvimento. Tais problemas tendem a ser minimizados a partir da aplicação de práticas enxutas de desenvolvimento de software.

O *Lean* surgiu a partir das premissas do *Toyota Production System* (TPS), idealizado durante a Segunda Guerra Mundial, em 1940, com foco em reduzir desperdícios nas linhas de produção da montadora que sofria grandes impactos financeiros por baixa demanda de venda dos veículos, que eram produzidos demasiadamente. Taiichi Ohno foi o idealizador do TPS, pois ele viu que o modelo tradicional não estava gerando lucros para a empresa, devido ao alto preço dos carros comparado com o que a população ganhava de dinheiro. Sendo assim, Shigeo Shingo, outro idealizador do TPS definiu os sete desperdícios da manufatura, que são: Inventário, super processamento, superprodução, transporte, espera, movimentação e defeitos.

Esses sete desperdícios da manufatura foram convertidos para a área de software, que se tornaram: Trabalho parcialmente feito, processos extras, funcionalidades extras, mudança de tarefas, espera, movimentação e defeitos [Poppendieck and Poppendieck 2003].

Uma das formas de sintetizar o processo de desenvolvimento de software de uma empresa e constatar se há desperdícios operacionais é aplicando o mapeamento de fluxo de valor.

2.3. Mapeamento de Fluxo de Valor

O Mapeamento de Fluxo de Valor é uma ferramenta desenvolvida pelo *Operations Management Consulting Division* (OMCD), e tem um papel importante para sintetizar os fluxos atuais dos processos. Suas características o tornam um diagrama amplamente utilizado para sintetização dos processos, pois ele fornece uma linguagem comum, visual e simbólica, tem fácil visualização e compreensão, ajuda a visualizar os processos individuais e os departamentais, auxilia na melhoria como um todo e não apenas de algumas equipes, agrega técnicas e conceitos da manufatura enxuta [Rother and Shook 2003].

Após o MFV ser difundido dentro da filosofia *Lean*, ele se tornou um método adequado para empresas que desejam implementar o conceito *Lean*, pois ele indica quais as ações a serem tomadas após a identificação dos desperdícios praticados durante o de-

envolvimento de software. O MFV possibilita medir e a avaliar os processos a baixo custo, já que é uma simples visualização dos processos realizados, e favorece a mudança cultural da empresa, já que fornece a compreensão dos benefícios obtidos pela melhoria dos processos [Machado and Tretin 2017].

3. Metodologia

O Fluxograma da Figura 1 representa o fluxo utilizado para a coleta dos dados, análise dos dados até o encontro de evidências de desperdícios operacionais.

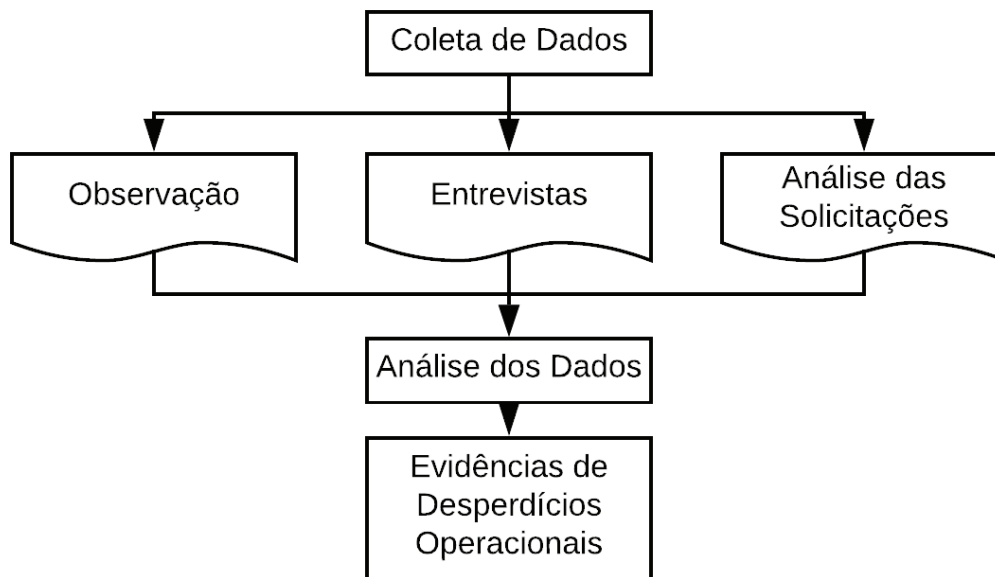


Figura 1. Fluxograma da Metodologia a ser Utilizada
Fonte: Dados da Pesquisa

3.1. Coleta de Dados

Para a coleta de dados, objetivando a identificação e redução dos desperdícios operacionais em uma fábrica de produção de software, foi utilizado a abordagem triangular, envolvendo três métodos de coleta dos dados, sendo eles: Observação, entrevistas e análise das solicitações. Essas técnicas de coleta de dados foram escolhidas devido a sua compatibilidade com o estudo, possibilitando uma abrangência maior e uma coleta melhor dos dados do objeto de estudo.

3.1.1. Observação

O primeiro método aplicado foi a observação do atual fluxo de trabalho da empresa, acerca dos processos utilizados na solicitação de customização e melhoria que os clientes requerem. Essas solicitações chegam através do centro de suporte da empresa, onde são categorizadas conforme a sua demanda: Customização de Sistema ou Melhoria de Sistema.

- Customização de Sistema: É identificada quando já existe uma funcionalidade no sistema e o cliente deseja alterá-la para melhor se encaixar nos processos operacionais do cliente;
- Melhoria de Sistema: É caracterizada quando o cliente deseja uma nova funcionalidade no sistema.

Depois de receber a solicitação de melhoria ou customização, a equipe de Gestão de Mudanças é acionada e encaminha a solicitação para a estimativa de horas da equipe responsável pela situação. Essa estimativa de horas compreende três aspectos: Análise, Desenvolvimento e Testes.

Após a estimativa da solicitação ser concluída, a solicitação é encaminhada para a equipe dos representantes comerciais, responsáveis de entrar em contato com o cliente, avisando-o a respeito da estimativa de custo que é gerada em decorrência da solicitação, caso ela seja realizada. Com o aceite de investimento por parte do cliente, a solicitação retorna para a equipe responsável para que seja alocada nas iterações de desenvolvimento, onde, inclusive, será estimado a sua data de entrega.

Após a finalização dos processos da fábrica, os analistas de qualidade entram em contato com o cliente para estar reportando que a melhoria/customização solicitada está pronta para ser utilizada.

A observação permitiu analisar o fluxo atual de forma minuciosa, com muitos detalhes nos documentos gerados.

3.1.2. Entrevista

Outra forma de coleta dos dados necessários para este estudo foi o método de entrevistas, com o propósito de validar se o modelo utilizado atualmente pela empresa nos processos operacionais, comerciais e nos processos de fábrica de desenvolvimento, supre as necessidades dos clientes, satisfazendo-os a respeito de suas solicitações. Dois métodos de entrevista foram utilizados: Face a Face e Mediada.

A entrevista Face a Face é realizada pessoalmente com o entrevistado, por outro lado, a mediada utiliza-se de uma ou mais ferramentas tecnológicas para proceder com a entrevista, tais como vídeo conferência ou ligação.

A estrutura da entrevista é semiestruturada, ou seja, foi elaborado um roteiro de questões a serem feitas para o entrevistado para que ele a responda, porém a entrevista não é limitada apenas a essas questões, é permitido que o entrevistado contribua com outras informações relevantes ao projeto.

3.1.3. Análise das Solicitações

A análise das solicitações foi mais um passo crucial ao estudo, e ocorre por meio da análise das conversas que foram realizadas com o cliente por meio da ferramenta de HelpDesk, atualmente utilizada no Centro de Suporte e na Fábrica de Desenvolvimento de Software da empresa.

Essas conversas são armazenadas em Logs que, por sua vez, contém os dados

necessários para o levantamento do MFV atual de cada solicitação, dados como: Data de recebimento, data de encaminhamento, data de contato com o cliente, se o solicitado é o que foi entregue, entre outros.

3.2. Análise dos Dados

A análise dos dados foi feita de forma específica, solicitação por solicitação. Os critérios para a seleção das solicitações foram os seguintes: Já encaminhada para desenvolvimento; Já entregue para o cliente; Com investimento por parte do cliente. Tais critérios foram selecionados com o propósito de ter uma base de solicitações com data de recebimento e data de entrega, e que tenha passado por todos os processos e equipes.

Após ter as solicitações categorizadas por demanda, elas foram submetidas ao MFV para que se possa ter uma visão clara do *lead time* de cada solicitação, gerando dados concretos de qual equipe tem se delongado mais em cumprir seus prazos com as solicitações, evidenciando a problemática levantada por essa proposta de trabalho.

4. Resultados

4.1. Evidências de Desperdícios Operacionais

A aplicação dos métodos de coleta de dados foi concluída e os seguintes resultados obtidos: O processo de observação retornou todo o caminho que uma solicitação percorre dentro da empresa, desde a sua solicitação até a entrega ao cliente. Com o método das entrevistas foi possível analisar as ações tomadas com situações de clientes críticos e que precisam de mais atenção para evitar um possível cancelamento, e a análise das solicitações retornou o tempo utilizado em cada equipe para a resolução da solicitação. Ao final, os dados coletados foram agrupados e foram encontradas 14 solicitações de melhoria / customização do software da empresa, cujas solicitações se encaixam nos critérios da 3ª etapa da coleta dos dados. O mapeamento de fluxo de valor foi aplicado nas 14 solicitações previamente selecionadas. A seguir, as Figuras 2 e 3 ilustram exemplos dos mapas que comprovam o tempo de ociosidade nos respectivos processos.

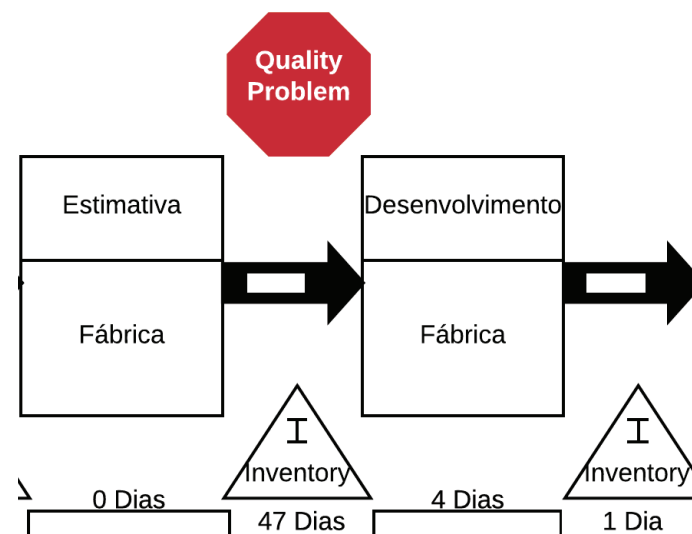


Figura 2. Solicitação 1
Fonte: Dados da Pesquisa

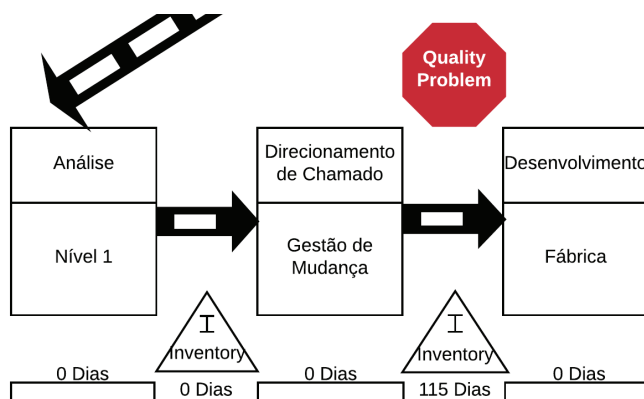


Figura 3. Solicitação 2
Fonte: Dados da Pesquisa

Na linha inferior do mapeamento é mostrado o lead-time da solicitação, ou seja, todo o tempo que ela esteve dentro da empresa. Os dias na parte superior representam o tempo trabalhado e o dias na parte inferior é o tempo de ociosidade da solicitação. As figuras apresentam apenas o problema identificado e não a solicitação completa, e, ao final da linha, é mostrado o *lead-time* total da solicitação. A solicitação da Figura 2 tem um *lead-time* de 78 dias, sendo 15 trabalhados e 63 dias de ociosidade. Na Figura 3 há um *lead-time* de 147 dias, sendo 27 trabalhados e 120 dias de ociosidade.

Dentro do contexto do objeto de estudo, alguns processos se repetem em cada uma das 14 solicitações. Após a análise dos 14 mapas, alguns destes processos apresentaram um padrão de desperdícios operacionais, que são apresentadas na Figura 4.

	Média de Dias de Ociosidade	Falhas Encontradas
Pendente de Estimativa	78,75	8
Pendente de Comercial	63,88	6
Pendente de Desenvolvimento	45,12	8
Desenvolvimento	31	2
Pendente de Gestão de Mudanças	27	1
Aceite do Cliente	18,12	8
Negociação com o cliente	18	1
Análise do Nível 1	7	1

Figura 4. Média de Dias de Ociosidade
Fonte: Dados da Pesquisa

O desperdício operacional que ficou mais evidenciado foi o Pendente de Estimativa, onde, nas 14 solicitações, ele foi evidenciado em 8 casos, conforme destacado na coluna: "Falhas Encontradas". Para o cálculo da média de dias de ociosidade, foi realizado uma média aritmética dos tempos de ociosidade de tal situação, onde, dos 8 casos de ociosidade, as solicitações ficaram em média 78,75 dias em situação de desperdício. O tempo ocioso que uma solicitação fica "Pendente de Estimativa" é identificado geralmente na transição entre a equipe de Gestão de Mudanças e a equipe de Estimativas. Vários fatores podem estar elevando o tempo de ociosidade, podendo ser eles: Alta demanda de estimativas a serem realizadas; Poucas pessoas aptas a fazer a estimativa de uma solicitação;

Priorização de outras demandas ao invés de seguir o fluxo correto por ordem de chegada e prioridade principal das solicitações. Esses fatores podem estar transformando esse tempo de ociosidade em um desperdício operacional considerável, conforme evidenciado. Outro fator que merece destaque é a situação, destacada na Figura 4 como: "Pendente de Desenvolvimento". Tal circunstância foi evidenciada em 8 casos dos 14 mapas analisados. A média de tempo em ociosidade das falhas encontradas foi de 31 dias. Diferentes fatores podem estar impactando em tais problemáticas, como: Má priorização de demandas, falta de comunicação entre a equipe operacional e a equipe de desenvolvimento, entre outros.

5. Conclusão

A coleta de dados obteve sucesso em seus três métodos, mostrando resultados relevantes para o estudo e evidenciando que o tempo de ociosidade das solicitações está muito alto ao ser comparado com o tempo trabalhado em cada solicitação. Esse desperdício operacional encontrado nesta análise possibilita uma visão clara dos processos que atualmente estão demandando mais tempo para serem executados e conseqüentemente sugere a mudança de culturas e processos visando melhor satisfação dos solicitantes.

A proposta de solução para o problema identificado é a aplicação de melhorias nos processos internos da empresa, tais como: Aumento de equipes; Capacitação das equipes atuais; Melhor categorização das solicitações no centro de suporte. A aplicação destas melhorias e coleta de resultados caracteriza-se como um trabalho futuro, comparando as coletas de dados pré-melhorias e pós-melhorias para certificar-se que estas foram benéficas e aumentaram a satisfação dos solicitantes.

Referências

- Beck, K. and Gamma, E. (2000). *Extreme programming explained: embrace change*. addison-wesley professional.
- Hirama, K. (2011). *Engenharia de Software, Qualidade & Produtividade com Tecnologia*. Elsevier, Rio de Janeiro.
- Machado, P. and Tretin, M. G. (2017). Análise dos impactos na utilização de melhoria contínua em processos Ágeis de desenvolvimento de software: Um estudo de caso.
- Poppendieck, M. and Poppendieck, T. (2003). *Lean software development: an agile toolkit*. Addison-Wesley.
- Pressman, R. S. (2011). *Engenharia de Software: Uma Abordagem Profissional*. AMGH Editora.
- Rother, M. and Shook, J. (2003). *Learning to see: value stream mapping to add value and eliminate muda*. Lean Enterprise Institute.