

Sistemas Produtivos e Desenvolvimento Profissional: Desafios e Perspectivas

O Impacto do *Lean Software Development* na Engenharia de Requisitos de Software

Eliana Santos de Oliveira

Centro Estadual de Educação Tecnológica Paula Souza – Unidade de Pós-Graduação, Extensão e Pesquisa – São Paulo - Brasil

santos.elianasantos@gmail.com

Marília Macorin de Azevedo

Centro Estadual de Educação Tecnológica Paula Souza – Unidade de Pós-Graduação, Extensão e Pesquisa – São Paulo - Brasil

marilia.azevedo@fatec.sp.gov.br

Antônio Cesar Galhardi

Centro Estadual de Educação Tecnológica Paula Souza – Unidade de Pós-Graduação, Extensão e Pesquisa – São Paulo - Brasil

galhardi.antoniocesar@gmail.com

Resumo: O estudo tem por objetivo demonstrar os resultados atingidos por meio de pesquisa realizada junto aos profissionais da área de engenharia de software em especial os profissionais da área de engenharia de requisitos. Durante a pesquisa foram apresentadas aos respondentes questões relacionadas aos princípios do *Lean Software Development* combinadas com as ferramentas da engenharia de requisitos com o intuito de apurar junto aos profissionais, qual a aplicabilidade dos princípios do *Lean Software Development* na engenharia de requisitos de software. Na fundamentação teórica são apresentados conceitos sobre engenharia de requisitos, *Lean Thinking* e *Lean Software Development*.

Palavras-chave: *Lean Thinking*; Software; Engenharia de Requisitos; *Lean Software Development*

Abstract : The study aims to demonstrate the results achieved through research conducted with the professional software engineering especially the professional requirements engineering. During the research were presented to respondents issues related to the principles of Lean Software Development combined with engineering tools requirements in order to ascertain with professionals , which the applicability of the principles of Lean Software Development in engineering software requirements. In theoretical foundation are presented concepts on requirements engineering, Lean Thinking and Lean Software Development .

Keywords: Lean Thinking; Software; Requirements Engineering; Lean Software Development

1. Introdução

O mercado atualmente apresenta um cenário que exige agilidade, rapidez e assertividade no processo de tomada de decisão, na operacionalização dos processos, bem como no processamento das informações.

Para auxiliar no gerenciamento desse cenário as organizações contam com sistemas informatizados, desenvolvidos para atender às necessidades e expectativas dos usuários e das organizações.

Porém, a partir do momento em que se detecta a necessidade de criação e desenvolvimento de um software até a etapa em que o software é de fato “entregue” para uso, existem diversas fases que compõem o seu processo de desenvolvimento. Dentre estas fases tem-se a engenharia de requisitos.

A engenharia de requisitos é a área responsável por compreender as necessidades e expectativas do cliente e repassar essas necessidades para a equipe de desenvolvimento.

Para auxiliar a engenharia de requisitos a realizar o seu papel de compreensão e tradução das expectativas do cliente, esse estudo propõe a aplicação dos princípios do *Lean Software Development* em seus processos.

O *Lean Software Development* é uma metodologia criada por Mary e Tom Poppendieck em 2003 e tem como objetivo orientar o processo de desenvolvimento de software de forma a ter o mínimo de desperdício e o máximo de valor agregado para o cliente além de contribuir para a melhora na entrega dos projetos de software.

O presente estudo demonstra os resultados de uma pesquisa realizada junto aos profissionais da área de engenharia de requisitos a respeito da utilização e aplicabilidade dos princípios do *Lean Software Development* na área de engenharia de requisitos.

2. Referencial Teórico

Nessa seção do estudo são apresentados os principais conceitos que foram utilizados como base para a construção das questões que compõem o questionário e utilizadas também para embasamento técnico na análise dos resultados obtidos.

2.1. Engenharia de Requisitos de Software

A Engenharia de Requisitos de Software está inserida no contexto da engenharia de software, e é responsável pelo levantamento, entendimento e documentação dos requisitos junto ao usuário.

Segundo Kotonoya e Sommerville (1997), o termo engenharia de requisitos refere-se ainda a todas as atividades realizadas para a descoberta, documentação e manutenção de um conjunto de requisitos para o desenvolvimento de um determinado sistema. Pode-se dizer ainda que a engenharia de requisitos de software é a disciplina que atua na determinação dos objetivos e funções dos requisitos de software.

Segundo Pressman (2011), engenharia de requisitos pode ser definida como: “a área de engenharia de software responsável pelo entendimento dos

requisitos; inicia-se na comunicação dos requisitos e vai até a etapa de modelagem”.

A engenharia de requisitos tem um papel importante dentro do projeto de desenvolvimento de software, pois é a responsável por definir qual é o escopo das alterações, o que será de fato desenvolvido e qual a prioridade de cada item. Se nessa etapa o diagnóstico do projeto for realizado de maneira incorreta as chances de fracasso e retrabalho no projeto aumentam de maneira significativa.

É importante ressaltar que o processo de desenvolvimento de software por si só é uma tarefa complexa, independente da complexidade do software a ser desenvolvido e por uma série de variáveis: usuários, cenário organizacional e cultura da empresa. Enfim todos esses fatores contribuem para que a tarefa de desenvolvimento de software seja complexa.

Arruda (2009) cita como sendo os principais problemas da área de engenharia de requisitos:

a) Falta de definição clara dos requisitos: na maioria dos projetos os *stakeholders* não sabem exatamente o que o sistema deve fazer e isso gera impactos no momento da construção do sistema.

b) Estrutura da organização: em organizações com uma estrutura rígida, há certa dificuldade em acessar os usuários que detém as informações necessárias para o desenvolvimento do sistema.

c) Resistência por parte dos usuários: geralmente os usuários apresentam um pouco de resistência ao processo de levantamento de requisitos, especialmente quando as alterações no sistema ou a implantação de um sistema já existente implica em mudança na sua rotina.

d) Falta de tempo hábil para o processo de requisitos: em alguns projetos o tempo destinado ao desenvolvimento dos requisitos é reduzido e em alguns casos até mesmo subtraído para que possa ser atendido o prazo de desenvolvimento e entrega (ARRUDA, 2009).

Parte dos problemas que impactam a engenharia de requisitos é de origem externa e social, e exige muito mais do que a correta utilização das inúmeras ferramentas existentes no mercado; exige, sim, habilidades sociais e de relacionamento por parte do analista.

Outros fatores também contribuem para a ocorrência desses problemas nos processos de engenharia de requisitos entre eles: cultura organizacional; estrutura de gerenciamento de projetos dentro da organização; a falta de conhecimento dos usuários no projeto como um todo e por conta dessa falta de conhecimento os usuários ficam resistentes em fornecer as informações necessárias para o bom andamento do projeto.

2.3 Lean Thinking

O *Lean Thinking* (Pensamento Enxuto) é uma forma de especificação de valor onde é realizada a compreensão do funcionamento de todo o processo e em seguida são definidos os seguintes itens: definição da melhor sequência para a realização das atividades; execução das tarefas de uma única vez e sem

interrupções, sempre com o intuito de gerar valor para o cliente (WOMACK; JONES, 2004).

A ideia da realização de um determinado trabalho com o mínimo de desperdício e o máximo de eficiência é uma ideia enxuta (WOMACK; JONES, 2004).

A principal característica do pensamento enxuto é identificar e especificar o que é de fato valor percebido pelo cliente. Para isso, são propostos cinco princípios que podem auxiliar os gestores na aplicação do pensamento enxuto nas organizações (COSTA; JARDIM, 2010).

Identifique o que é valor para o cliente: Esse princípio parte do pressuposto de que valor é tudo aquilo que gera benefícios para um prazo e custo específicos, ao identificar o valor para o cliente deve-se levar em consideração o que o cliente realmente necessita e a partir desse conceito procura-se atender as expectativas (WOMACK; JONES, 2004).

Mapeie o fluxo de valor e identifique o desperdício: Refere-se ao entendimento de todas as etapas e passos necessários para que o processo (serviço ou produção de bens) gere o valor esperado pelo cliente; identificar e eliminar toda e qualquer forma de desperdício (WOMACK; JONES, 2004).

Implante o fluxo contínuo: A implantação do fluxo de valor contínuo consiste em fazer com que o valor do processo flua pela cadeia sem interrupções desnecessárias, que ocorra de maneira sincronizada e eficiente (SELLITTO; BORCHARDT; PEREIRA, 2010).

Adoção de um sistema de produção puxada: Um determinado produto ou serviço só pode ser produzido a partir da solicitação do cliente, ou seja, só há trabalho a partir da solicitação de uma demanda; caso não haja demanda, não há trabalho; dessa forma, evita-se o desperdício (SELLITTO; BORCHARDT; PEREIRA, 2010).

Busca pela Perfeição: Esse princípio estabelece que mesmo após a modelagem e implantação do processo *Lean*, é necessário buscar continuamente o aperfeiçoamento e melhoria dos processos (WOMACK; JONES, 2004).

O *Lean Thinking* prega a realização das atividades sempre visando o máximo de retorno ao cliente e o mínimo de desperdício ou seja o foco principal de toda e qualquer atividade é sempre agregar o máximo de valor ao cliente de maneira que suas expectativas e necessidades sejam satisfeitas sempre com o máximo de qualidade e a utilização mínima de recursos.

2.4 Lean Software Development

O *Lean Software Development* foi idealizado com a intenção de eliminar todos os desperdícios do desenvolvimento de software, cortando os excessos e produzindo somente aquilo que é necessário ao cliente.

Para isso, Mary e Tom Poppendieck (2003) desenvolveram sete princípios que norteiam o desenvolvimento de software a partir da perspectiva *Lean*. Os princípios são:

1. **Eliminar Desperdício:** Consiste em eliminar a linha do tempo que existe entre o momento que o cliente realiza a demanda e o momento em que a entrega foi de fato entregue.
2. **Integrar Qualidade:** Significa desde o início de o desenvolvimento evitar o máximo de erros; isso pode ser alcançado através da inspeção do

código. A inspeção pode ser realizada de duas formas: inspeção após a ocorrência do erro e a inspeção para prevenir a ocorrência do erro.

3. Criar Conhecimento: Consiste em entender que o software é um organismo vivo que cresce e evolui de acordo com as tendências de mercado.

4. Adiar Comprometimentos: Significa treinar a equipe para que adie o máximo possível as decisões de forma que só tome as decisões após ter o máximo de informações disponíveis.

5. Entregas Rápidas: Consiste em realizar entregas em pequenos pacotes; dessa forma, os clientes já iniciam a utilização do software além de já usufruir dos benefícios.

6. Respeitar as Pessoas: Significa respeitar o trabalho da equipe e protegê-la de interferências externas além de oferecer condições favoráveis de trabalho.

7. Otimizar o Todo: Consiste em otimizar todo o processo, não apenas o processo de desenvolvimento de software, mas também o processo do cliente.

3. Método

A pesquisa foi de natureza quali-quantitativa. Para o levantamento de dados e a avaliação das práticas propostas, que consistiu no questionamento de um grupo de respondentes especialistas no assunto com o intuito de obter informações acerca do entendimento que os sujeitos de pesquisa possuem a respeito do tema pesquisado, e buscar a validação do cenário proposto no questionamento; pode ser aplicado tanto em pesquisas qualitativas ou quantitativas. (CASTRO; REZENDE, 2009).

As questões inicialmente foram confeccionadas com o intuito de apurar do respondente qual o entendimento que ele possui a respeito da aplicação dos princípios do *Lean Software Development* na engenharia de requisitos de software. Sendo assim, nas questões são apresentados cenários nos quais o respondente é levado a avaliar a aplicabilidade no cenário proposto.

Os dados apresentados nessa pesquisa foram extraídos de um questionário elaborado para esse fim e foram analisados de acordo com o referencial teórico.

4.1 Caracterização dos Respondentes

O público da pesquisa é composto por especialistas na área de engenharia de requisitos de software (analista de sistemas, analista de requisitos e analista de negócios), segmentados por tempo de experiência, tipo de sistema em que atua e área de atuação da empresa. A seleção da amostra foi realizada por conveniência por parte da autora deste estudo e contou com 50 profissionais da área de software que possuem o seguinte perfil:

- 60% atuam em grandes empresas (acima de 99 funcionários).
- 40% possuem mais de 10 anos de experiência e 27% possuem entre 5 e 10 anos de experiência; 20% possuem entre 3 e 5 anos de experiência e 13% possuem de 1 a 3 anos de experiência.
- 86% atuam em empresas do setor de serviços.

As questões inicialmente foram confeccionadas com o intuito de apurar do respondente qual o entendimento que ele possui a respeito da aplicação dos princípios do *Lean Software Development* na engenharia de requisitos de software. Para tanto, elaborou-se uma associação entre as principais ferramentas da engenharia de requisitos e os 7 princípios do *Lean Software Development*. Nas questões são apresentados cenários nos quais o respondente é levado a avaliar a aplicabilidade no cenário proposto.

As questões foram construídas a partir da associação entre as ferramentas e os princípios do *Lean Software Development* e os cenários foram apresentados questionando aos profissionais sobre a utilização dos princípios em determinadas situações:

Tabela 1: Correlação entre os princípios e ferramentas

Princípios do <i>Lean Software Development</i>	Ferramentas da Engenharia de Requisitos
Eliminar Desperdícios	<i>Brainstorming</i>
Construir Certo da Primeira Vez	Prototipação
Criar Conhecimento	JAD
Adiar Comprometimento	Entrevista (Validação)
Promover Entregas Rápidas	Prototipação
Respeitar Pessoas	Etonografia
Otimizar o Todo	<i>Brainstorming</i>

Fonte: Elaborado pelo autora

4. Resultados e Discussão

Nesta sessão serão apresentados os percentuais de utilização de cada um dos princípios do *Lean Software Development* em cada uma das ferramentas a tabela abaixo exibe o percentual de utilização dos princípios:

Tabela 2: Resultados do questionário

Princípio \ Ferramenta	Percentual de Uso
Princípio 2: Construir certo da primeira vez	58% utilizam entre 80% e 100% do tempo
Princípio 5: Entrega Rápida	45% utilizam entre 80% e 100% do tempo
Princípio 1: Eliminar Desperdício e <i>Brainstorming</i>	47% utilizam entre 20% do tempo ou nunca
Princípio 7: Otimizar Recursos e <i>Brainstorming</i>	44% utilizam entre 40% e 100% do tempo
Princípio 4: Adiar Comprometimentos	49% utilizam entre 80% e 100% do tempo
Princípio 6: Respeitar as pessoas	45% utilizam entre 80% e 100% do tempo
Princípio 3: Criar Conhecimento	53% utilizam entre 60% e 100% do tempo

Fonte: Elaborado pelo autora

A tabela 2 mostra os níveis de utilização de cada um dos princípios na engenharia de requisitos os princípios grifados em vermelho são os princípios que apresentam um baixo índice de utilização e os princípios grifados em verdes apresentam um nível de utilização satisfatório.

O princípio 2 – Construir certo da primeira vez (58%) apresenta um nível de utilização de 58% o que indica a preocupação dos profissionais com esse ponto, entretanto é importante frisar que construir certo da primeira vez não implica em realizar a entrega com todos os requisitos atendidos logo na primeira vez e sim, realizar a entrega o mais aderente possível com a necessidade do cliente e com uma quantidade mínima de falhas, se houver.

O princípio 3 – Criar Conhecimento (53%) consiste em disseminar o conhecimento adquirido com toda a equipe de forma que todos tenham acesso as informações importantes relacionadas ao projeto.

O princípio 4 – Adiar comprometerimentos (49%) consiste em adiar o máximo possível decisões importantes relacionadas a arquitetura e funcionalidades, ao adiar uma decisão pois nesse intervalo de tempo as ideias são amadurecidas e o projeto avança mostrando novas possibilidades e alternativas ao adiar uma tomada de decisão têm se a oportunidade de se reunir o máximo de informações e tomar a decisão de maneira acertada.

Os princípios: Princípio 1 - Eliminar desperdício (47%) e Princípio 7 - Otimizar recursos (44%) estão entre os princípios com o menor índice de utilização. A pouca utilização do princípio 1 (Eliminar desperdício) pode ser explicada pelo fato de que a identificação de desperdício em software é bastante complexa, uma vez que a maioria dos desperdícios em software é considerada como situações normais no cotidiano desses profissionais. Entre essas situações pode-se citar: funcionalidades extras; códigos inacabados e antecipação de funcionalidades.

Para o princípio 7 - Otimizar recursos, a dificuldade pode residir na padronização dos processos dentro das organizações, especialmente na área de desenvolvimento de software que trabalha por demanda e com forte pressão, o que dificulta a criação e manutenção de processos que permitam otimizar a área como um todo.

5. Considerações Finais

Os resultados, apontam que os profissionais possuem a percepção sobre a utilização dos princípios do *Lean Software Development*, e que essa contribuição é positiva, ou seja, gera ganhos para a engenharia de requisitos.

As vantagens da utilização desses princípios refletem diretamente nas entregas realizadas aos clientes pois a filosofia Lean tem o foco totalmente voltado para o cliente e sua satisfação e isso implica na melhoria das entregas e da satisfação do cliente.

O *Lean Software Development* traz benefícios também para os profissionais da área de engenharia de requisitos uma vez que os auxilia a ter uma visão voltada para aquilo que é essencial ao cliente e a detectar desperdícios tanto no processo atual do cliente como nos processos futuros.

As análises realizadas neste trabalho podem auxiliar na definição de métodos de trabalho para levantamento de requisitos em diferentes tipos de projetos, por exemplo: em projetos com uma grande quantidade de detalhes e de *stakeholders*, irá exigir uma atenção e cuidados redobrados na utilização de algumas práticas aqui propostas.

Importante ressaltar que cada projeto de desenvolvimento de software possui suas peculiaridades e a utilização dessas metodologias devem ser cuidadosamente avaliadas, bem como a utilização de qualquer outra metodologia.

Muito há ainda a ser desenvolvido na relação entre o *Lean Software Development* e a engenharia de requisitos de forma a criar e adaptar mecanismos do *Lean* para os pontos mais específicos da área de engenharia.

REFERÊNCIAS

ARRUDA, Natasha de Souza. Engenharia de Requisitos: Como Prevenir e Reduzir Riscos. 2009. Disponível em: <<http://www.aedb.br/seget/artigos11/30114261.pdf>> Acesso em: 02 mar 2014

COSTA, Ricardo Sarmiento; JARDIM, Eduardo. G.M. - Os Cinco Passos do Pensamento Enxuto (*Lean Thinking*). 2010. Disponível em: <<http://www.trilhaprojetos.com.br>> Acesso em: 01 mar 2014.

FAGUNDES, Priscila Bastos. **Framework Para Comparação e Análise De Métodos Ágeis**. 134 f. Dissertação (Mestrado em Ciências da Computação). Universidade Federal de Santa Catarina, Florianópolis, 2005.

KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering - Processes and Techniques**. John Willy & Sons, 1997.

POPPENDIECK, Mary; POPPENDIECK, Tom. *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional, 2003.

_____. *Implementando o Desenvolvimento Lean de Software – Do Conceito ao Dinheiro*. São Paulo: Bookman, 2011.

PRESSMAN, Roger S. **Engenharia de Software – Uma Abordagem Profissional**: 7ª edição. São Paulo: Editora Mc Graw Hill, 2011.

SELLITO, Miguel Afonso; BORCHARDT Miriam; PEREIRA, Giancarlo Pereira. 2010. Presença dos princípios da mentalidade enxuta e como introduzi-los nas práticas de gestão das empresas de transporte coletivo de Porto Alegre. Disponível em: <http://www.prod.org.br/doi/10.1590/S0103-65132010005000009> Acesso em: 11 fev 2014

WOMACK, J.; JONES, D. **A mentalidade enxuta nas empresas**. Rio de Janeiro: Campus, 2004.

