

DOI:10.5748/9788599693131-14CONTECSI/RF-4856

THE INFLUENCE OF LEAN SOFTWARE DEVELOPMENT ON SOFTWARE REQUIREMENTS ENGINEERING

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

The study aims to show how the principles of Lean Software Development (LSD) relate to the area of requirements engineering and help to achieve better results. To measure these benefits, a research was carried out in order to determine the professionals of the area The applicability of LSD principles in the requirements engineering process. The initial results pointed to a positive perception of the applicability of the principles by the professionals.

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

A INFLUÊNCIA DO LEAN SOFTWARE DEVELOPMENT NA ENGENHARIA DE REQUISITOS DE SOFTWARE

O estudo tem por objetivo mostrar como os princípios do Lean Software Development (LSD) se relacionam com a área de engenharia de requisitos e a auxiliam a atingir melhores resultados. Para mensurar esses benefícios foi realizada uma pesquisa com o intuito de apurar dos profissionais da área de requisitos a aplicabilidade dos princípios do LSD no processo de engenharia de requisitos. Os resultados iniciais apontaram para uma percepção positiva da aplicabilidade dos princípios por parte dos profissionais.

Palavras-chave: Lean Thinking; Software; Engenharia de Requisitos; 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.

Entretanto, existem momentos em que é necessária a automatização das atividades relacionadas aos negócios para a agilidade na tomada de decisão. Para que essa automatização ocorra de maneira assertiva é necessário entender a real necessidade do cliente, quais são seus anseios e expectativas.

A engenharia de requisitos é a área responsável por compreender as necessidades e expectativas do cliente e traduzir essa resposta em especificações de sistemas e softwares que auxiliem os clientes em suas atividades.

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, com o intuito de promover mais agilidade e assertividade nas atividades relacionadas.

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.

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.

Segundo Kotonoya e Sommerville (1997), o termo engenharia de requisitos refere-se 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”.

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.

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.

Alguns fatores 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.

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 detêm 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).

2.3 LeanThinking

O *LeanThinking* (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 (Sellito; 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 (Sellito; 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 *LeanThinking* 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 e que de fato agrega valor ao seu negócio.

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.

Ao utilizar estes princípios durante o período de desenvolvimento de software é possível notar o quanto de tempo e recursos são gastos no desenvolvimento de funcionalidades que não agregam valor de fato ao cliente.

3. Método de Pesquisa

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 para buscar uma validação do cenário proposto no questionamento; pode ser aplicado tanto em pesquisas qualitativas ou quantitativas. (Castro; Rezende, 2009).

A abordagem quantitativa tem por objetivo determinar as características de uma determinada população frente a um produto ou serviço, ou seja, a abordagem quantitativa deve ser utilizada em situações onde é possível quantificar os dados. (Moresi, 2003).

Diferente da abordagem quantitativa, a qualitativa não segue um procedimento rígido para a avaliação dos dados. Entre suas principais características estão: a análise de experiências de indivíduos ou grupos, procurando entender quais são suas impressões a respeito de um determinado tema ou de um determinado contexto; o estudo das formas de interações e comunicações entre os indivíduos envolvidos e através da investigação de documentações e relatos sobre um determinado assunto. (Flick, 2009).

De acordo com Appolinário (2012), a pesquisa qualitativa trata da coleta de dados por meio das interações entre o pesquisador e o seu objeto de pesquisa, para posterior tratamento e análise, e busca, ainda, descrever comportamentos e situações variáveis e a pesquisa quantitativa envolve controle estatístico e controle da amostra.

A natureza da pesquisa realizada neste estudo é qualitativa-quantitativa, pois, envolve aspectos tanto da abordagem qualitativa quanto da abordagem quantitativa.

Para Creswell e Clark (2006), as vantagens da utilização da abordagem combinada de pesquisa (qualitativa e quantitativa) são:

- Ajuda a responder questões que não podem ser respondidas separadamente.
- Compensa a lacuna de ambas as abordagens.
- Traz evidências mais abrangentes para o estudo de um problema de pesquisa do que cada abordagem separadamente.

Por conta desses pontos, a abordagem combinada foi o tipo de abordagem escolhida para a condução deste estudo. 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 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* os cenários foram apresentados questionando aos profissionais sobre a utilização dos princípios em determinadas situações.

A tabela 1 mostra a correção que foi realizada entre os princípios e as principais ferramentas utilizadas pela engenharia de requisitos e mapeadas nesse estudo. O critério utilizado foi a aderência entre a ferramenta, por exemplo: Para o princípio eliminar desperdício a correlação foi realizada com a ferramenta *Brainstorming*, uma vez que eliminar desperdício implica primeiro no conhecimento de todo o processo e o *Brainstorming* é uma ferramenta interessante para promover o conhecimento do processo e dessa forma identificar possíveis desperdícios e atuar na sua eliminação.

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 pelas autoras

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 2 exhibe o percentual de percepção dos profissionais na utilização dos princípios durante a execução das suas atividades relacionadas à engenharia de requisitos:

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 pelas autoras

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 verde apresentam um nível de utilização satisfatório.

O princípio 2 – Construir certo da primeira vez 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 a acesso às informações importantes relacionadas ao projeto, além de criar uma cultura de gestão do conhecimento onde todos podem ter acesso ao conhecimento adquirido ao longo do projeto, bem como as lições aprendidas.

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

O princípio 5 – Entregas Rápidas (45%) com um uso considerado satisfatório e com uma utilização razoável, pois está ligado diretamente a um dos princípios do *Lean* que é o de agregar valor ao cliente em um curto espaço de tempo. Esse princípio se refere a realizar entregas de funcionalidades para os clientes.

O princípio 6 – Respeitar Pessoas (45%) também apresenta um nível satisfatório e está ligado ao fato de, na equipe de desenvolvimento, delegar uma determinada atividade para um profissional que é visto como um especialista naquele assunto para o qual foi designado.

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.

Ainda, com relação aos resultados apresentados, eles respondem ao objetivo deste estudo que é: analisar a percepção dos profissionais da área de engenharia de requisitos com relação aos princípios do *Lean* e os dados apontam que essa percepção é positiva.

Dos sete princípios apresentados, cinco atingiram níveis considerados satisfatórios de utilização com destaque para os princípios: Construir certo da primeira vez; Criar conhecimento e Adiar comprometerimentos, que apresentaram um alto nível de utilização. A percepção positiva desses princípios pode estar relacionada com a utilização das metodologias ágeis que são usadas nos projetos de desenvolvimento de software, que carregam esses princípios e que influenciam de maneira positiva a área de engenharia de requisitos.

Em suma, o estudo alcançou o seu objetivo de demonstrar que os profissionais possuem uma percepção positiva sobre a utilização dos princípios do *Lean Software Development* na engenharia de requisitos e que pode trazer benefícios para esta área do conhecimento.

REFERÊNCIAS

- ARRUDA, Natasha de Souza. Engenharia de Requisitos: Como Prevenir e Reduzir Riscos. 2009. Disponível em: <<http://www.aedb.br/seget/arquivos/artigos11/30114261.pdf>> Acesso em: 02 mar 2014
- COSTA, Ricardo Sarmiento; JARDIM, Eduardo. G.M. - Os Cinco Passos do Pensamento Enxuto (*LeanThinking*). 2010. Disponível em: <<http://www.trilhaprojetos.com.br>> Acesso em: 01 mar 2014.
- CRESWELL, John W.; CLARK, Vicki L. P. Designing and conducting mixed methods research. EUA, SagePublications, 1ª. Edição, 2007.
- 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.
- FLICK, Uwe. Desenho da pesquisa qualitativa. Coleção Pesquisa Qualitativa. 1ª ed. São Paulo: Bookman, 2009.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS* – IEEE Std. 830. **IEEE guide to software requirements specification**. The Institute of Electrical and Electronics Engineers. New York, EUA. 1984.
- _____. **IEEE International Symposium on Requirements Engineering**. IEEE Computer Society Press. Los Alamitos, Ca, USA, 1997, p. 44-53.
- KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering - Processes and Techniques**. John Willy & Sons, 1997.
- MORESI, Eduardo. Metodologia da Pesquisa. 2003. Disponível em: <http://ftp.unisc.br/portal/upload/com_arquivo/1370886616.pdf> Acesso em: 04 nov. 2013.
- 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.

ZAVE, Pamela. **Classification of Research Efforts in Requirements Engineering**. ACM Computer Surveys, vol 29, n°4, 1997.