

A METODOLOGIA LEAN

Bruna Cardoso¹
Gabriella Aquino²
Henrique Silvestre³
Mateus Augusto⁴

Antônio Junior⁶ – Orientador

¹brunacardosojanuario@gmail.com; ²xavier.gabi@gmail.com;
³gb_henrique@hotmail.com; ⁴mateusaugusto7@gmail.com;
⁵antonio.moura@prof.unibh.br;

Curso de Ciência da Computação – Uni-BH (www.unibh.br)

Resumo - Neste artigo é apresentada metodologia de desenvolvimento Lean, como surgiu, seus princípios, vantagens e desvantagens

Palavras-chave Lean, Desenvolvimento, Software

Abstract –On this article is presented the methodology Lean Software development, how it came, its principles, advantages and disadvantages

Keywords –Lean, Development, Software

1 INTRODUÇÃO

No mercado de TI, no setor de desenvolvimento de softwares, pensa-se como um mercado sempre aquecido, e realmente é, entretanto, diversas organizações veem seus projetos fracassarem antes de sua finalização. Um levantamento realizado pela Standish Group, mostra o percentual de utilização das funcionalidades dos sistemas comerciais nos Estados Unidos da América (EUA), pode-se perceber então que cerca de 80% a 90% dos projetos de software fracassam, pois no fim não

são utilizados, seja por não atender a necessidade do cliente, seja por não ter sido entregue em tempo hábil, conforme mostra o gráfico abaixo na Figura01:

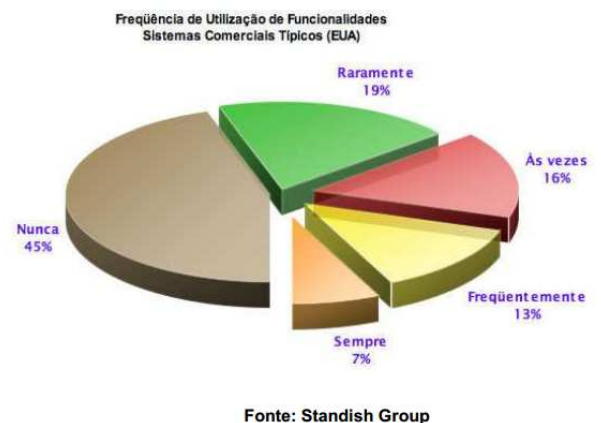


Figura01 – Levantamento Standish Group

Por que estes projetos fracassam? São inúmeras as situações que podem inviabilizar um projeto, abaixo estão citadas algumas delas:

- Sistemas entregues com atrasos ou/ou orçamento estourado
- Não atendem aos requisitos do negócio

- Clientes descontentes(sem confiança nos desenvolvedores)
- Clientes não compreendem claramente o que está sendo desenvolvido
- Cliente não dá suporte adequado ao desenvolvimento
- Clientes não se interessam pelos “problemas”
- Time de desenvolvimento sem disciplina, desmotivados ou com carga de trabalho muito alta
- Desenvolvimento focado na tecnologia sem se preocupar com o negócio

A partir destas questões começou a entrar no mundo de desenvolvimento de software a palavra “Ágil” e criou-se alguns conceitos como “Estamos evidenciando maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através desse trabalho, passamos a valorizar:

- Interação entre pessoas MAIS QUE processos e ferramentas;
- Software em funcionamento MAIS QUE documentação abrangente;
- Responder a mudanças MAIS QUE seguir um plano
- Colaboração com o cliente MAIS QUE negociação de contratos

Então surge os princípios do desenvolvimento ágil:

- Satisfazer o cliente, entregando o software em tempo hábil e continuamente
- Aceitar as mudanças de requisitos, em qualquer fase do projeto
- Entregar software na menor escala de tempo possível
- Equipe de desenvolvimento e cliente são do mesmo time
- Construir projetos com indivíduos motivados e comprometidos com o resultado

Entendido o que são metodologias ágeis, no próximo capítulo será abordada a metodologia Lean, que é o foco deste trabalho.

2 LEAN

O conceito de *Lean Thinking* (Pensamento enxuto), nasceu em meados dos anos 90 criado pela empresa japonesa Toyota para guiar processos industriais de linha de montagem. Ela foca na eliminação de desperdícios, aumento da velocidade de processos e na excelência em qualidade. Os princípios de demanda puxada (*pull systems*), *just-in-time*, qualidade total, teoria das restrições, melhoria contínua e flexibilidade aplicados na indústria japonesa, inspiraram também a indústria de software e fez surgir a abordagem do Lean

Software Development.(Juliana Berossa Steffen, 2011).

Então, o que realmente é o Lean e o que ele representa para o desenvolvimento de software?! De acordo com o Lean Institute Brasil [1], “Lean é uma estratégia de negócios para aumentar a satisfação dos clientes através da melhor utilização dos recursos. A Gestão Lean procura fornecer consistentemente valor aos clientes com os custos mais baixos (Propósito) através da identificação de melhoria dos fluxos de valor primários e de suporte (Processos) por meio do envolvimento das pessoas qualificadas, motivadas e com iniciativa (Pessoas). O foco da implementação deve estar nas reais necessidades dos negócios e não na simples aplicação das ferramentas lean.”.

Segundo Mary Poppendieck [2], expert de Lean voltado a TI, uma das defensoras do agile e uma das autoras do livro: "Lean Software Development - An Agile Toolkit" que aborda como os princípios Lean podem ser aplicados em abordagens de desenvolvimento de software ágil, o desenvolvimento de software Lean é a aplicação dos princípios da *Toyota product development system* para o desenvolvimento de software. Quando ele é aplicado corretamente, o desenvolvimento é de alta qualidade, além de ser realizado rapidamente e possuir um baixo custo. Além disso, o desenvolvimento ágil possui um grande sucesso devido ao Lean.

3 PRINCÍPIOS

Princípios da Lean:

1. Elimine Desperdícios
2. Inclua a Qualidade no Processo
3. Crie Conhecimento
4. Adie Decisões e Comprometimentos
5. Entregue o quanto antes
6. Respeite as Pessoas e "Empower" a equipe
7. Otimize o Todo

1. Elimine Desperdícios

Eliminar tudo que não agrega valor ao cliente e não é percebido pelo mesmo. Por exemplo:

- Funcionalidades incompletas: gera desperdício, porque dependem de esforço para serem iniciadas e não agregam valor ao cliente
- Excesso de processos: gera desperdício porque demanda recursos e aumenta o tempo de conclusão das tarefas.
- Criação de documentos complexos: Gera desperdício,

pois raramente são úteis, difíceis de serem entendidos, demanda muito tempo para serem elaborados e frequentemente se tornam obsoletos, pois não são atualizados a medida que se atualiza o sistema.

Há inúmeros outros exemplos que caracterizam desperdício na visão da filosofia Lean.

2. Inclua a qualidade no processo

Se rotineiramente encontra-se defeitos nos sistemas em um processo de verificação, este processo é defeituoso

Qualidade é essencial a qualquer área, qualquer produto ou serviço, Mary e Tom Poppendieck, também autor do mesmo livro, em seu livro identificaram duas dimensões de integridade: integridade percebida e integridade conceitual. A integridade percebida significa que a completude do produto alcança um equilíbrio entre as funções, usabilidade, confiabilidade, economia e isso ganha o cliente. Entende-se por integridade conceitual os conceitos centrais do sistema de trabalho em conjunto são facilitados e coesos. Essa última é fator essencial de êxito para a integridade percebida. Um produto possui integridade percebida quando o cliente o experimenta e diz: "Era exatamente isso que eu queria!". Software com integridade possui boas

arquiteturas, possuem um alto nível de usabilidade e facilidade de uso, são fáceis de dar manutenção, de adaptar e de estender.

3. Crie conhecimentos

De acordo com Filho [3], lições devem ser extraídas das experiências vividas pela equipe e incorporadas ao processo, fazendo com que as dificuldades passadas sejam fonte de conhecimento e contribuam para o amadurecimento da equipe e do processo. O uso de um processo definido engessa o aprendizado. Desenvolvimento de software é melhor realizado se este integrar um processo de aprendizado parecido ao de criar uma nova receita. A melhor abordagem para melhorar o ambiente de desenvolvimento de software é pela expansão do conhecimento. Práticas sugeridas para promover o conhecimento:

- Ciclos de feedback e inspeções e adaptações;
- Desenvolvimento iterativo;
- Equipes pequenas e cross-functional;
- Treinamentos e Mentoring;
- Criação e utilização de standards, guidelines e qualquer outro artefato;
- Code Reviews;
- Meios de compartilhamento de informações como um Blog ou Wiki;

4. Adie decisões e comprometimentos

Diminuir incertezas retardando as decisões até que possam ser tomadas com base em acontecimentos mais concretos, previsíveis e de total conhecimento. Decisões tardias têm a tendência de serem mais acertadas porque as melhores decisões são feitas baseadas em fatos, e não em suposições ou especulações. Uma estratégia para adiar decisões/ comprometimentos quando desenvolvendo um sistema complexo e com muitas incertezas é usar a capacidade e práticas que permitam realizar as mudanças tardiamente.

5. Entregue o quanto antes

De acordo com Filho [7], com ciclos rápidos o desenvolvimento caminha através de um processo iterativo no qual o cliente aprimora suas necessidades e as obtém implementadas já no próximo ciclo. Iterações curtas trazem mais experiência para o time e aumentam a sua confiança para tomar decisões.

6. Respeite as pessoas e “Empower” a equipe

É extremamente necessário que os desenvolvedores, que são os responsáveis por entregar o software, sejam envolvidos nos detalhes das decisões técnicas, isto é fundamental para o um produto da excelência

Dicas para motivar o time:

- Um ambiente que favoreça o desenvolvimento das pessoas.
- Uma organização que respeita as pessoas, assim as pessoas irão respeitar a organização

O Software desenvolvido é um reflexo da equipe de desenvolvimento. Para que as pessoas possam assumir responsabilidades, se sentir motivadas e atuar como um time elas precisam de confiança e de respeito. Práticas sugeridas para promover o *empowering* do time:

- Trabalho em equipe
- Feedback do trabalho e desempenho

7. Otimize o todo

Otimizar desde o início até o fim, esta é a meta. Para Franco [4], para que haja a obtenção da integridade nos sistemas de grande complexidade é preciso um conhecimento profundo de diversas áreas.

Sugestões de como visualizar o todo

- Utilize Métricas
- Diminua o número de métricas de desempenho individual mas valorize o desempenho da equipe.
- Meça para cima:

- Tempo de ciclo + Mapa de Fluxo de Valor
- ROI + Modelo de Lucros e Perdas
- Satisfação do Cliente + Entendimento das suas necessidades

metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta. Escola Politécnica da Universidade de São Paulo (Dissertação de Mestrado). 2007.

4 CONCLUSÃO

Por fim, entende-se que as metodologias ágeis podem ser implementadas de forma a complementar as metodologias tradicionais ou mesmo substituir. Entende-se que Lean é uma metodologia bem elaborada e pautada em fatos, pois surgiu a partir de uma demanda de mercado e se adaptou as necessidades dos clientes, priorizando a visão do negócio, sendo a tecnologia uma ferramenta para obtenção do valor de negócio do cliente.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] <http://www.lean.org.br/>- Entidade sem fins lucrativos de pesquisa, educação, treinamento e suporte à transformação lean.
- [2] POPPENDIECK, M.: Lean Software Development. 29th International Conference on Software Engineering. IEEE. 2007.
- [3] FILHO, D. L. B.: Experiências com desenvolvimento ágil. Instituto de Matemática e Estatística da Universidade de São Paulo (Dissertação de Mestrado).2008
- [4] FRANCO, E. F.: Um modelo de gerenciamento de projetos baseado nas