Ludmila Varela Arruda[1]

[1]Aluna do Curso de Engenharia de Telecomunicações – IFCE. e-mail: ludmila_varela@ifce.edu.br

DESENVOLVIMENTO ÁGIL DE SOFTWARE: UMA ANÁLISE SINTÉTICA A PARTIR DA METODOLOGIA KANBAN

Resumo: Diante da real complexidade encontrada no processo de desenvolvimento de um software, diversas metodologias foram criadas a fim de solucionar tal problema. As que mais tem se destacado são as ágeis, que contemplam ideias mais próximas à realidade dos clientes e empresas e que, portanto, tem adquirido um forte espaço no mercado. A meta desse artigo é situar os leitores no contexto dos processos de software e seus métodos, apresentando as sugestões oferecidas por mais um desses procedimentos, que é o Kanban, com o propósito de difundi-lo. Quanto maior o conhecimento sobre as metodologias de elaboração e acompanhamento de um sistema, melhor será a análise a fim de encontrar uma que seja a mais apropriada para o ambiente do projeto que se deseja desenvolver. O kanban, por sua vez, engloba características e estratégias simples que tem atendido de forma satisfatória muitas organizações, na atualidade.

Abstract: Given the complexity found in real development process of software, various methodologies have been created to solve this problem. Those most outstanding are the agile ideas that come closer to the reality of companies and customers and, therefore, has acquired a strong market share. The goal of this paper is to situate the reader in the context of software processes and methods, with the suggestions offered by most of these procedures, which is the Kanban, in order to spread it. The greater the knowledge about the methods of preparing and monitoring a system, becomes most easy to analise in order to find one that is most appropriate for the project environment you want to develop. The kanban, in turn, includes features and simple strategies that satisfy many companies today.

Palavras-chave: agilidade, desenvolvimento de software, kanban, metodologias, sistema Toyota de produção

1. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Construir um software que seja usável e que atenda às necessidades de um determinado cliente não é uma tarefa simples. A maioria dos sistemas desenvolvidos hoje, no final do processo, não tem todas as suas funcionalidades em uso ou sequer entram em produção.

Ao longo dos anos, os estudiosos da área de Engenharia de Software tentam aperfeiçoar esse processo, de maneira que o software possa colaborar concretamente com o usuário final. Esse é o objetivo da automatização das atividades.

Cada fase desse processo possui suas particularidades, bem como as suas dificuldades de execução. Estas estão descritas abaixo:

a) Análise e Levantamento de Requisitos: Essa etapa define quais são os problemas enfrentados pelos usuários do futuro sistema e quais as soluções e desejos que os mesmos pretendem adquirir. É importante que todo o ambiente do problema seja analisado, a fim de estudar quais são as reais necessidades. Normalmente, os requisitos, se não forem detalhados com clareza, trazem complicações durante todas as outras fases e comprometem os resultados. É também considerada a etapa mais relevante.

b) Projeto: Após a fase de análise, os requisitos precisam ser traduzidos para a linguagem

ISBN 978-85-62830-10-5

computacional, seja qual for a tecnologia utilizada. Assim, os dados devem ser estruturados e a arquitetura do sistema deve ser montada. Ou seja, algumas diretrizes são cumpridas, como a definição de hardware básico, sistema operacional, interfaces, linguagem de programação etc. Entretanto, essa representação é abstrata, por enquanto. "O processo de projeto pode envolver o desenvolvimento de vários modelos do sistema em diferentes níveis de abstração." (SOMMERVILLE, 2010, 8 ed, p. 50).

c) Implementação ou Codificação: É nesta etapa onde a especificação será convertida em um sistema executável (torna-se possível a compilação e geração de código). Aqui se concentra o objetivo da Engenharia de Software, o que não quer dizer que detenha a maior importância. Seja de forma interativa ou incremental, o código vai apresentando mudanças e evoluções ao longo do tempo, o que concede forma ao sistema como um todo.

d) Teste e Validação: Depois que o software é codificado, é necessário que sejam realizados diversos testes (funcionais e não funcionais) para certificar-se que os requisitos especificados pelo cliente foram atendidos. Essa fase deve ser contemplada durante todo o processo de desenvolvimento, visto que, quanto mais cedo os erros são descobertos, mais facilmente estes são corrigidos. O grande número de falhas encontradas em um sistema tem sido uma das maiores dificuldades das equipes que constroem software atualmente. Logo, essa etapa caracteriza um dos principais alvos da Engenharia de software, que é a qualidade.

e) Manutenção: Entregar o produto devidamente testado não significa a conclusão do trabalho no processo. O código deve ser implementado visando mudanças posteriores, portanto, ele deve ser manutenível. Mesmo depois de "pronto", novos requisitos podem surgir e o sistema deve estar preparado para recebê-los.

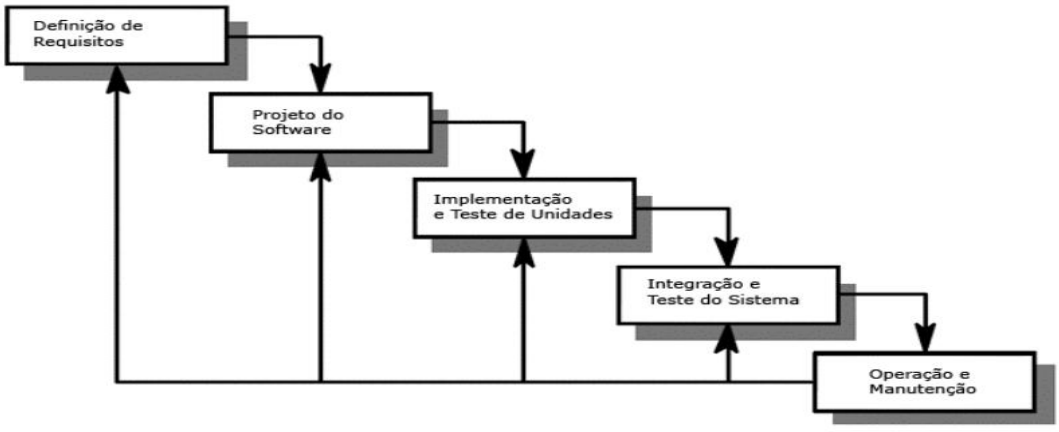


Figura 1: Fases do modelo de Desenvolvimento Clássico
Fonte: Adaptado de Sommerville (2003)

A figura 1 sintetiza o processo de desenvolvimento que foi descrito através da definição de suas fases. Mesmo com os inúmeros estudos realizados nessa área, a implementação de softwares ainda é uma atividade custosa e delicada.

2. METODOLOGIAS DE DESENVOLVIMENTO ÁGIL

ISBN 978-85-62830-10-5

Levando em consideração a dificuldade de desenvolver um sistema, muitas metodologias foram criadas, melhoradas e adaptadas com o objetivo de solucionar esse impasse. O dicionário [WEBSTERS, 1998] define metodologia como um conjunto de métodos, regras e postulados empregados por uma disciplina: um procedimento particular ou conjuntos de procedimentos. Portanto, a maioria das empresas utilizam esses recursos para melhorar seus resultados.

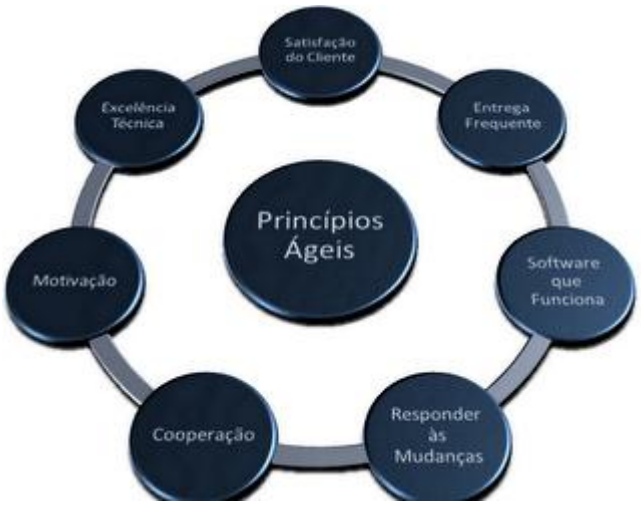
Temos como exemplo algumas das principais metodologias aplicadas atualmente: Scrum, XP (Extreme Programming), RUP (Rational Unified Process), Kanban, FDD (Feature Driven Development), entre outras. Contudo, o conjunto de regras que vem sendo implantado com força maior e que tem evoluído consideravelmente chama-se "Metodologias Ágeis de Software" ou simplesmente "Agilidade", onde está o foco do estudo desse artigo.

Mas, o que vem a ser agilidade? No mundo da Engenharia de Software, este termo nada tem a ver com rapidez na entrega de artefatos, como muitos pensam. Na verdade, trata-se de tentar amenizar os riscos de desenvolvimento através de curtas interações. Interações são intervalos reduzidos de tempo, nos quais determinadas funcionalidades do sistema serão desenvolvidas. Quando conseguimos organizar certo número de atividades em um período de tempo, torna-se mais facilitada a construção do software, além do cliente poder ter acesso livre e rápido ao que está sendo implementado. É como pensarmos em todas as fases descritas no processo de desenvolvimento de software acontecendo em vários ciclos de tempo. Os requisitos são analisados e descritos, a arquitetura é planejada, o código é feito, os testes são realizados e uma parte do sistema é entregue. Depois, novamente os requisitos são analisados, a arquitetura é replanejada e assim por diante.

Esse termo "Metodologias Ágeis" surgiu a partir da criação do Manifesto Ágil, um documento que tem como conceitos principais:

- Indivíduos e interações ao invés de processos e ferramentas;
- Software executável ao invés de documentação;
- Colaboração do cliente ao invés de negociação de contratos;
- Respostas rápidas a mudanças ao invés de seguir planos.

As ferramentas, documentações, contratos, por exemplo, não foram excluídos do processo, mas já não recebem tanta importância como em outras metodologias. O foco da agilidade está nas pessoas, nas interações, no sistema, na participação do cliente.



ISBN 978-85-62830-10-5

Figura 2: Princípios Ágeis
Fonte: http://ebookbrowse.com/pbjug-tech-day-processos-de-desenvolvimento-de-software-pdf-d296232956

Embora seja notável a difusão dos métodos ágeis, isso não indica que as metodologias tradicionais não funcionem. Porém, a estrutura física e organizacional das empresas atuais tem dado preferência à agilidade, por esta se comportar de forma positiva quanto aos requisitos que estão sempre emergindo, aos projetos com um número de pessoas pequeno, aos ambientes de baixa criticidade etc. Dentro desse contexto, umas das metodologias que ganhou grande destaque nos últimos anos foi o Scrum, que se relaciona com conceitos da Engenharia e não possui regras específicas para a fase de desenvolvimento.

A partir dessa metodologia, outra tem assumido grande valor no mercado, o Kanban, que é o objeto de estudo desse trabalho.

3. DESENVOLVIMENTO ÁGIL COM KANBAN

Primeiramente, o que é kanban? Partindo do princípio que se trata de mais uma metodologia ágil de desenvolvimento, qual o seu diferencial? “Kanban é um método de desenvolvimento de software com fortes bases em práticas Lean, e que tem como objetivo aperfeiçoar o processo de desenvolvimento de software pré-existente. Este método limita o trabalho em progresso, apresentando a evolução de forma visual, tornando os problemas evidentes e cultivando uma cultura de melhoria contínua.” (GOMES, André. Desenvolvimento Ágil com kanban. Java magazine, 84, pg. 1).

As práticas Lean ou Sistema Toyota de produção, como são mais conhecidas, caracterizam a estrutura do kanban. No passado, as indústrias utilizavam o sistema de produção que conhecemos como “produção empurrada”, onde uma atividade era executada, tendo seu resultado “empurrado” para a atividade posterior. Logo, não existia relação entre as atividades produzidas e a demanda do cliente. Com o Sistema Toyota ou produção “puxada”, o processo passou a ser diferente. Cada atividade executada dependia da demanda da atividade executada anteriormente.

Esse sistema tem como princípio eliminar todo e qualquer tipo de desperdício, além de ter como primícia o modelo de Just in time, o qual determina que exista um momento adequado para cada fase do desenvolvimento. Nada deve ser feito sem que esteja no tempo apropriado e que seja necessário.

Trazendo esses conceitos para o processo de desenvolvimento de software, o trabalho de construir uma nova funcionalidade para um sistema só é gerado a partir do momento que uma funcionalidade anterior já tenha sido implementada. O kanban, dentro desse contexto, procura aperfeiçoar os processos, as equipes e projetos. É útil para empresas que estão procurando melhorar constantemente, ao passo que melhoram também sua produtividade e sua relação com os clientes. “Esse método começou a crescer após sua adoção na conferência Agile, em agosto de 2007. Sua primeira aplicação para Engenharia de Software foi na empresa Microsoft em 2004”. (ANDERSON, David. Kanban: Successful Evolutionary Change for Your Technology Business, cap.2).

O principal objetivo dessa metodologia é avaliar o trabalho em progresso, chamado WIP (Work in Progress). Essa avaliação tem a proposta de mostrar quando uma funcionalidade do software pode ser arquitetada, codificada, testada etc. Mas quando se percebe que mais trabalho pode ser “puxado”? Ou seja, quando se percebe que outras atividades já podem ser desenvolvidas

ISBN 978-85-62830-10-5

pela equipe?

A demonstração vem pela visualização. O kanban, como o próprio nome diz, é uma palavra japonesa que significa "cartão sinalizador". Através do quadro de atividades, é possível perceber qual a quantidade de esforço que poderá ser adicionada. É importante lembrar que, para isso, é preciso também levar em consideração a indicação de limite e de capacidade da equipe e do software que está sendo desenvolvido. Algumas das empresas que implantam essa metodologia possuem um quadro com espaços que indicam a capacidade.

As vantagens encontradas para adotar o kanban são grandes. A equipe pode fazer entregas a qualquer instante para o cliente e o mesmo pode modificar a importância das atividades quando desejar. O desenvolvimento fica mais transparente, já que é possível visualizar o fluxo de trabalho, sem preocupações com as iterações e estimativas, como em outros métodos. Caso isso seja relevante para um projeto, no qual o foco vai estar na implementação do produto ou que tenha a necessidade de ter papéis bem definidos, não é indicada a aplicação do kanban.

A figura a seguir retrata a definição de uma ferramenta que auxilia a implantação desse método. Com um quadro simples e cartões, pode-se criar um controle de fluxo de atividades.

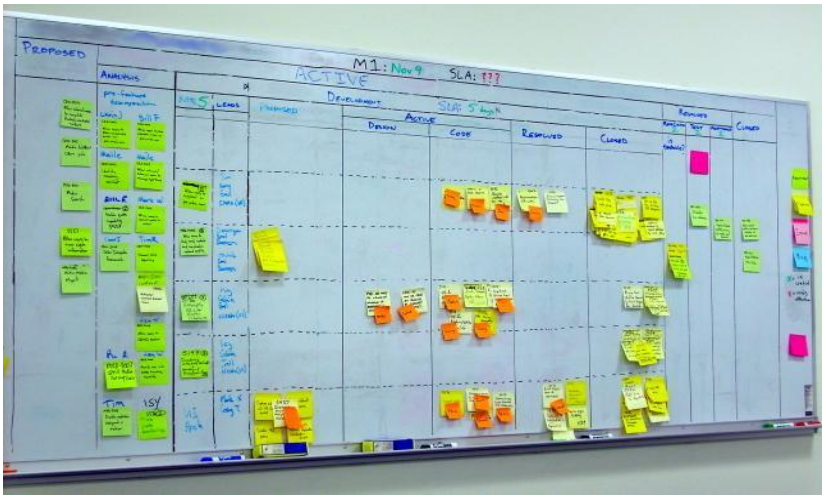


Figura 3: Quadro ilustrativo
Fonte: http://www.devmedia.com.br/desenvolvimento-agil-com-kanban-java-magazine-84/18235

Para começar, é importante mapear o que é mais prioritário dentro do software, estabelecendo pontos estratégicos, definindo o trabalho que deverá estar em progresso e montando a estrutura visual que facilitará o acompanhamento.

4. CONCLUSÕES

A dificuldade de desenvolver um software que seja utilizável tem sido amenizada com os novos conceitos e soluções propostas pelas metodologias ágeis. O kanban, alvo desse estudo, tem esse foco. Quando o cliente pode definir a prioridade das atividades e pode mudá-la a qualquer momento, sua satisfação é maior. Quando a equipe pode acompanhar o fluxo de todas as atividades do processo, podendo também controlar o trabalho que está em progresso, esta trabalha de forma mais produtiva e com qualidade. Objetivos são traçados e a simplicidade torna-se lei no desenvolvimento do projeto, o que traz como retorno grandes facilidades para todos. Além disso, o

ISBN 978-85-62830-10-5

kanban pode ser combinado com outros métodos, bem como pode ser adaptado de acordo com a realidade da empresa. Não existe uma metodologia padrão a ser adotada em um projeto, existe uma que seja mais favorável à sua necessidade.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Rodrigo. Processos de Desenvolvimento de Software. PBJUG, 2012. http://ebookbrowse.com/pbjug-tech-day-processos-de-desenvolvimento-de-software-pdf-d296232956. Data de acesso: 12/06/2012;

BASTOS, Aderson et al. Base de conhecimento em teste de software. 2ª edição. São Paulo: Martins Editora, 2007;

GOMES, André. Desenvolvimento Ágil com kanban. Java magazine, 84. http://www.devmedia.com.br/desenvolvimento-agil-com-kanban-java-magazine-84/18235. Data de acesso: 10/06/2012;

Manifesto Ágil de Software, 2012. http://www.agilealliance.org/. Data de acesso: 20/06/2012.

SOMMERVILLE, Ian. Engenharia de Software. 8ª edição. São Paulo: Pearson, 2010;

ANDERSON, David. Kanban: Successful Evolutionary Change for Your Technology Business. United States, 2010;

LADAS, Corey. Scrumban - Essays on Kanban Systems for Lean Software Development. United States: Modus Cooperandi, 2008.

ISBN 978-85-62830-10-5