**CONFIABILIDADE, DISPONIBILIDADE E MÉTODOS ÁGEIS NA ENGENHARIA SOFTWARE**

Jefferson Oliveira do Nascimento. ¹

Filipe Bruno Moreira Nascimento. ²

Wellington Feitoza Gonçalves ³

**RESUMO:** Este trabalho tem por finalidade relacionar conhecimentos sobre como a engenharia de software contribuiu para a criação e desenvolvimento profissional de software e como poderá contribuir oferecendo confiabilidade e disponibilidade, mesmo tendo que lidar com desafios do tipo pouco tempo para muitas demandas, o que gerou necessidades de mudanças devido a grandes problemas em desenvolvimentos de projetos de software, o artigo será realizado com base nas referencias bibliográficas que comprovam a importância desses conhecimentos, e como a aplicações de métodos ágeis tornaram-se algo imprescindível no desenvolvimento de software confiáveis, na tentativa de incentivar e mostrar que essa engenharia ainda tem muito a contribuir com as organizações e grandes empresas, que estão sempre em busca da qualidade, rapidez e confiança dos novos produtos nessa área.

**Palavras-chave:** Engenharia de software, métodos ágeis, desenvolvimento de software, confiabilidade, disponibilidade.

**ABSTRACT:** This study aims to relate knowledge about software engineering contributed to the creation and development of professional software and how you can contribute by offering reliability and availability even having to deal with challenges of type short time to many demands, which generated changing needs due to major problems in software project development, the article will be based on bibliographic references that prove the importance of this knowledge, and how the application of agile methods have become something essential in the development of reliable software in an attempt encouraging and show that engineering still has much to contribute to organizations and large companies, who are always in search of quality, speed and reliability of new products in this area.

**Keywords:** software engineering, agile methods, software development, reliability, availability.

**INTRODUÇÃO**

Nos últimos anos a tecnologia tem avançado de forma assustadora, por ter que se adaptar ao novo mundo rápido e cheio de exigências e agilidades, na tentativa de fazer o desenvolvimento de forma mais rápida em produção de software, necessários para o funcionamento de grandes e pequenas empresas, e ainda para a transmissão de informações e produção em outras áreas, a engenharia de software que tem por objetivo o desenvolvimento profissional de software, como afirma Sommerville, 2011, vem tentando se adequar as grandes demanda mesmo pouco tempo para entrega, alem de ter outros desafios, como lidar com o aumento de diversidade e desenvolvimento de software confiáveis.

A engenharia de software por ser uma área da engenharia que se preocupa com todos os aspectos de produção dos software, passa a ser indispensável por proporcionar técnicas e métodos de acordo com a necessidade dos usurários e locais onde serão aplicados. O problema é que não existem técnicas e métodos definido que serão aplicados para a produção e desenvolvimento de todos os software na engenharia, cada software irá exigir métodos e técnicas diferentes, ou seja, é necessários um modelo bem definido dessas para cada e nova produção, sempre se preocupando com a realidade onde será aplicado.

Como a engenharia tentou se adequar a rapidez no pouco tempo exigido, projetos de desenvolvimento de software passaram a apresentar defeitos no desempenho, colocando em risco a confiabilidade e disponibilidade dos mesmos, o que fez com que grandes organizações dessa área passassem a produzir os métodos ágeis e ainda o desenvolvimento distribuído de software, na tentativa de obter as mudanças necessárias e cabíveis a nova situação.

Em (PRI 02b) verifica-se que diversos problemas e desafios ainda presentes no processo de desenvolvimento de software. Um dos problemas mais significativo é o planejamento, pois ele é o ponto de partida de qualquer atividade relacionado ao desenvolvimento de software. Considerando os desafios apresentados, um dos mais significativos envolve um processo de desenvolvimento baseados em equipes e ambientes fisicamente distribuídos, ou seja, Desenvolvimento Distribuído de Software (DDS). (prinkladnicki, 2002). Com isso, possibilitam-se os avanços e sofisticação de software que irão beneficiar as organizações que utilizam dessa distribuição de software, com a intenção de expandir sua produtividade com menos custos, garantindo a qualidade do serviço prestado.

A pesquisa realizada e apresentada nesse artigo, mostrar como a engenharia de Software através de seus modelos de processos e desenvolvimentos de software, passa a ser cada vez mais, importante na produção de produtos, procurando disponibilizá-los com qualidade e confiança aos interesses dos que vão utilizar essas tecnologias. Com o objetivo de enfatizar as grandes contribuições que a engenharia poderá oferecer, como já ofereceu e oferece nos dias atuais mesmo tento que enfrentar problemas e desafios já citados.

**ENGENHARIA DE SOFTWARE: IMPORTÂNCIA E CONTRIBUIÇÕES**

A engenharia de software tem por objetivo apoiar o desenvolvimento profissional de software mais do que a programação individual. Ela inclui técnicas que apóiam especificação, projeto e evolução de programas, que normalmente não são relevantes para o desenvolvimento de software pessoal. (SOMMERVILLE, 2011).

Engenharia de software é metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas; adequação aos requisitos funcionais do negocio do cliente e seus respectivos procedimentos pertinentes; efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na Tecnologia da Informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas. (REZENDE, 2005).

Com um mundo cada vez mais avançado e exigindo qualidade e rapidez nas entregas de produtos a engenharia como afirma, Sommerville, 2011, tem como principais desafios lidar com o aumento de diversidade, demandas pela diminuição do tempo para entrega e desenvolvimento de software confiáveis. Com isso, houve grandes problemas no desempenho de projetos e desenvolvimento desses produtos, trazendo consigo a falta de confiança e atrasando a disponibilidade desses, gerando prejuízos a reputação de forma direta ou indireta da engenharia envolvida.

Nesse contexto percebe-se cada vez mais a necessidade das organizações desenvolverem software tendo por base um processo bem definido. De acordo com o que foi escrito em (PRI. 02b) um processo de software é representado por um modelo. Existem diversos modelos na área de desenvolvimento de software, sendo assim, as empresas adotam um determinado modelo como referencia, customizando-o e adequando este modelo de acordo com a sua realidade. Este modelo deve ser operacionalizado por meio de uma metodologia estabelecendo basicamente a sequencia de atividades existentes e a relação entre elas, incluindo neste contexto os métodos e as ferramentas que dão suporte ao modelo, procurando sempre a qualidade total, tanto do processo como do produto final gerado seguindo-se o processo. (PRIKLADNICKI, 2002).

Na revista engenharia de software, diz que como uma resposta às crescentes pressões por inovação em prazos cada vez mais reduzidos, às necessidades de constantes mudanças de requisitos e ao mau desempenho de grande parte dos projetos de desenvolvimento de software, houve um movimento na comunidade de desenvolvimento de software que deu origem aos Métodos Ágeis*.*

Hoje em dia as grandes organizações estão cada vez mais distribuídos seus processo de desenvolvimento de software ao redor do mundo, visando ganhos de produtividade redução de custos, diluição de riscos e melhorias na qualidade. Ou seja, o software está evoluindo de uma forma, muito rápida e, além disso, se vê a necessidade de distribuir o desenvolvimento, trazendo muitos benefícios para as organizações. (PRIKLADNICKI, 2002).

Percebe-se dessa maneira que a engenharia tem muito a contribuir com os avanços tecnológicos, assim como contribuiu com a internet, essa segundo Sommerville 2011, era basicamente um armazenamento de informações acessível universalmente e tinha pouco efeito nos sistemas de software. Esses sistemas executavam em computadores locais e eram acessíveis apenas dentro de organizações. Por volta do ano 2000, à internet começou a evoluir, e mais e mais recursos passaram a ser adicionados aos navegadores; levando ao desenvolvimento de novos produtos que oferecem serviços inovadores e que eram acessados através da internet.

**ENGENHARIA DE SOFTWARE: MÉTODOS ÁGEIS E DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE**

Métodos ágeis e desenvolvimento distribuído de

software são duas realidades presentes no mundo altamente competitivo e globalizado vivenciado

atualmente nas empresas. (OLIVEIRA e LIMA, 2011).

Segundo PRESSMAN (2006) as idéias de um desenvolvimento ágil sempre estiveram junto aos engenheiros de software, mas somente na década de 1990 que estas idéias foram transformadas em um movimento. O desenvolvimento ágil possui um esforço grande para vencer as fraquezas percebidas e reais da engenharia de software convencional, ele possui muitos benefícios, mas não é aplicado a todos os tipos de projetos, pessoas e situações. Em 2001 um grupo de notáveis desenvolvedores assinou um manifesto que expressa toda idéia do desenvolvimento ágil, abaixo temos um trecho do mesmo.

Com isso, os métodos ágeis consistem em princípios criados por o manifesto 2011, segundo o qual dar para ser ter noções de como e com que intenções foram criados, esses caracterizam por ter como prioridade a satisfação do cliente desde o inicio por meio de entrega continua de software valioso; os processos ágeis aproveitam as modificações como vantagens para competitividade do cliente; entrega de softwares funcionando frequentemente, a cada duas semanas até dois meses, de preferência no menor espaço de tempo; o pessoal de negocio e os desenvolvedores devem trabalhar juntos diariamente durante todo projeto; construção de projetos em torno de indivíduos motivados. Forneça-lhes o ambiente e apoio que precisam e confie que eles farão o trabalho; o método mais eficiente e efetivo de levar informação para e dentro de uma equipe de desenvolvimento é a conversa face a face; software funcionamento é a melhor medida de progresso; processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante, indefinidamente; atenção continua à excelência técnica e ao bom projeto facilitam a agilidade; simplicidade – a arte de maximizar a quantidade de trabalho não efetuado – é essencial; as melhores arquiteturas, requisitos e projetos surgem de equipes auto-organizadas; em intervalos regulares, a equipe reflete sobre como se tornar mais efetiva, então sintoniza e ajusta adequadamente seu comportamento.

*“[...] os Métodos Ágeis podem ser considerados uma coletânea de diferentes técnicas e métodos, que compartilham os mesmos valores e princípios básicos, alguns dos quais remontam de técnicas introduzidas em meados dos anos 70, como os desenvolvimentos e melhorias iterativos” (COHEN et al, 2003, p.2).(REV. ENGENHARIA DE SOFTWARE, 2014).*

Dessa forma os métodos tradicionais de desenvolvimento ficam cada vez mais esquecidos por consistir uma abordagem mais complexo e demandar grande organização da empresa, já os ágeis passa a ser uma forma mais ágil e ao mesmo tempo confiável para a organização das empresas.

Como afirma Prikladnicki 2002 e 2003, o processo de desenvolvimento de software baseados em equipes e ambientes fisicamente distribuídos, consiste no desenvolvimento distribuído de software, esse por sua vez, passou a ser utilizado por grandes empresas motivam um crescente número de organizações a desenvolverem software de forma distribuída.

Existem várias razões para a crescente popularização da distribuição do desenvolvimento de software: redução dos custos de produção, ganhos de escala, acesso aos recursos especializados, redução do tempo de colocação no mercado, melhoria na qualidade e acesso a novos mercados. Outra grande vantagem é que times distribuídos podem aumentar o aprendizado e a criatividade da equipe, já que os membros têm a chance de interagir com uma grande variedade de culturas, experiências e pontos de vista diversificados. No entanto, lidar com equipes constituídas por pessoas de diferentes localidades, culturas e expectativas para trabalhar em projetos distribuídos pode, muitas vezes, gerar dificuldades. (OLIVEIRA e LIMA, 2011).

**ENGENHARIA DE SOFTWARE NO AVANÇO TECNOLOGICO**

Para Tavares *et al, 2013,* Engenharia de Software assume papel crítico para garantir que tarefas, dados, pessoas e tecnologias estejam apropriadamente alinhadas para produzir um sistema efetivo e eficiente diz ainda que ela propõe estratégias de desenvolvimento, chamadas modelos de ciclo de vida de desenvolvimento de software ou modelos de processo.Estes modelos de ciclo de vida, como o nome diz, ajudam o desenvolvimento do início ao fim do projeto, que abrange as fases: Análise de Engenharia de Sistema; Análise de Requisitos de Software; Projeto; Codificação; Testes e Manutenção.

"O ciclo de vida clássico continua sendo o modelo procedimental

mais amplamente usado pela Engenharia de Software. Embora

tenha fragilidade, ele é significativamente melhor do que uma

abordagem casual ao desenvolvimento de software".

Pressman (1995).

Com tudo, fica claro a importância da engenharia no avanço tecnológicos, econômico e organizacional, das empresas que usufruem de seus produtos, passando a poder proporcionar muito mais vantagens e benefícios futuros.

**METODOLOGIA**

O artigo foi realizado com base em pesquisas bibliográficas que ressaltam e comprovam a importância da engenharia de software, e como as aplicações de métodos ágeis e desenvolvimento distribuído de software, tornaram-se imprescindíveis no desenvolvimento e disponibilidade de software confiáveis e adaptados a realidade dos clientes.

Segundo Marconi e Lakatos (1992), a pesquisa bibliográfica é o levantamento de toda a bibliografia já publicada, em forma de livros, revistas, publicações avulsas e imprensa escrita. A sua finalidade é fazer com que o pesquisador entre em contato direto com todo o material escrito sobre um determinado assunto, auxiliando o cientista na análise de suas pesquisas ou na manipulação de suas informações. Ela pode ser considerada como o primeiro passo de toda a pesquisa científica.

As buscas das referencias ocorreram de Agosto a Setembro de 2014, banco de dados utilizados para a busca foram: livros com assuntos abortados, revista engenharia de software, revista unifast, SCIELO, nos idioma inglês e português, sempre relacionando as referencias pré- estabelecidos aos assuntos abordados.

**CONSIDERAÇÕES FINAIS**

Sabendo-se das contribuições da engenharia de software nos avanços tecnológicos, mesmo com grandes desafios e sendo uma disciplina que tem muito a amadurecer, fica claro que esta passa a ser indispensável para a produção e desenvolvimento se software.

Nota-se que ao mesmo tempo em que aparecem problemas ela procura solucioná-los, com a criação e aplicação de programas que determinam mudanças inovadoras, geram assim novos produtos a serem aplicados, proporcionando uma disponibilidade com confiança de software que procura satisfazer, ressarcindo os interesses e realidade dos seus clientes.

**REFERÊNCIAS**

MARCONI, Marina de Andrade. LAKATOS, Eva Maria. Metodologia do Trabalho Cientifico. São Paulo: editora Atlas, 1992. 4° Ed. P. 43 e 44.

**Métodos Ágeis de Desenvolvimento de Software**. Revista Engenharia de Software, 20 edição especial. Disponível em http://www.devmedia.com.br/artigo-engenharia-de-software-20-metodos-ageis-de-desenvolvimento-de-software/15459, acesso em 03 de Setembro de 2014.

OLIVEIRA, Eneida. LIMA, Rosangela. Estado da Arte sobre Uso do Scrum em Ambientes de Desenvolvimento Distribuído de Software. Disponível em <http://www.revista.unifacs.br/index.php/rsc>. Revista de Sistemas e Computação, Salvador, v.1, n.2, p.106-119, Jul./Dez.2011.

PRESSMAN, Roger S. **ENGENHARIA DE SOFTWARE**. São Paulo: Pearson

Makron Books, 1995.

PRESSMAN, Roger S. **ENGENHARIA DE SOFTWARE**. São Paulo: McGraw-Hill, 2006.

PRIKLADNICKI, Rafael. MunDDos: Um Modelo de Referencias para Desenvolvimento Distribuído de Software. Porto Alegre, 2003. Disponível em http://repositorio.pucrs.br/dspace/bitstream/10923/1631/1/000423676-Texto%2BCompleto-0.pdf, acesso em 14 de Agosto de 2014.

PRIKLADNICKI, Rafael. Desenvolvimento Distribuído de Software e Processos de Desenvolvimento de Software. Porto Alegre, 30 de Agosto de 2002. Disponível em http://www.inf.pucrs.br/munddos/docs/TI2.pdf, acesso em 15 de Agosto de 2014.

REZENDE, Denis Alcides. Engenharia de Software e Sistemas de Informação. 3° Ed, Editora Brasport. Rio de Janeiro, 2005.

SOMMERVILLE, Ian. Engenharia de software. 9° Ed, Editora Pearson Prentice Hall, São Paulo, 2011.

Tavares, Ana Lúcia de Oliveira. Eckel, Ana Paula. Scarpa, Cateane. Vendrame, Závia Roselita. **Engenharia de Software: uma visão geral.** Disponível em <http://drc.objectis.net/faculdade-anhanguera-de-anapolis/pos-graduacao/engenharia-de-software/material/07-artigo-eng-sof.pdf>, acesso em 10 de setembro de 2014.