

Luiz Cezer Marrone Filho

**Scrum Distribuído - Práticas e
ferramentas de apoio para equipes
distribuídas**

Monografia apresentada no curso de Pós-Graduação do Centro Universitário Católica de Santa Catarina como requisito parcial para obtenção do certificado do curso.

Joinville
2015

Luiz Cezer Marrone Filho

Scrum Distribuído - Práticas e ferramentas de apoio para equipes distribuídas

Monografia apresentada no curso de Pós-Graduação do Centro Universitário Católica de Santa Catarina como requisito parcial para obtenção do certificado do curso.

Área de Concentração: Pós Graduação em Engenharia de Software

Orientador: Maurício Henning

Joinville
2015

Filho, Luiz Cezer Marrone

Scrum Distribuído - Práticas e ferramentas de apoio para equipes distribuídas. Joinville, 2015.

Monografia - Centro Universitário Católica de Santa Catarina.

1. Desenvolvimento de Software 2. Equipes distribuídas 3. Scrum I. Centro Universitário Católica de Santa Catarina. Curso de Bacharelado em Sistemas de Informação. Curso de Pós-Graduação em Engenharia de Software.

Sumário

Sumário	i
Resumo	ii
Abstract	iii
 Capítulo 1	
Introdução	1
1.1 Objetivo Geral	2
1.2 Objetivos Específicos	2
1.3 Justificativa do trabalho	2
 Capítulo 2	
Referencial Teórico	3
2.1 Metodologias Ágeis	3
2.2 Scrum	5
2.3 Trabalho distribuído	5
Referências Bibliográficas	6

Resumo

O tema deste trabalho é expor a forma como equipes de desenvolvimento de software geograficamente distribuídas podem usufruir de métodos ágeis, como por exemplo o *Scrum*, abordando quais seus pontos fortes e desafios a serem superados. Também tem como objetivo mostrar opções de ferramentas que podem auxiliar na colaboração e comunicação da equipe durante o projeto.

Palavras-chave: Desenvolvimento de Software, Equipes distribuídas, Métodos Ágeis, Scrum

Abstract

The purpose of this coursework is to expose the way that remote development teams can use agile methods, like *Scrum*, with focus in advantages, challenges to overcome. Also show which are the support tools to improve colaboration and communication between members during the software project.

Keywords: Software Development, Distributed teams, Agile, Scrum

Capítulo 1

Introdução

Nos dias atuais, empresas de desenvolvimento de software precisam de cada vez mais agilidade na hora de desenvolver e entregar seus projetos. Além de agilidade é preciso garantir consistência e um ritmo sustentável de trabalho, sem deixar de lado o cliente, garantindo a ele entregas periódicas do projeto, tudo isso sem esquecer da garantia de qualidade do que foi e está sendo entregue e que os requisitos podem mudar a qualquer momento. Para isso, empresas fazem o uso de métodos ágeis de gestão, afim de entregar software de forma incremental, mais rapidamente e com uma acertividade maior (ÁGIL, 2001).

Um exemplo desses métodos ágeis de gestão é o *Scrum*. O *Scrum* é um *framework* de gerenciamento de projetos, que possibilita um desenvolvimento iterativo e incremental do projeto, visando uma maior comunicação entre todos os envolvidos no projeto (ÁGIL, 2015).

Para que empresas possam evoluir e responder as demandas do mercado, precisam de cada vez mais profissionais melhores e mais capacitados, porém nem sempre é possível obter esses recursos humanos no local onde a empresa se encontra fisicamente, além disso o custo de manutenção de escritório, deslocamento dos funcionários e outros fatores podem atrapalhar na decisão de expandir a equipe com novos funcionários.

Na tentativa de reduzir esses impactos e ainda sim conseguir expandir a equipe, muitas empresas adotam a contratação de funcionários para trabalho remoto, o que faz com que a equipe se torne distribuída em partes ou em sua totalidade (ECÔNOMIA, 2015).

Porém mesmo que uma equipe seja parcialmente ou totalmente remota, ela ainda precisa ser organizada, comunicativa e focada na entrega, e o *Scrum* é uma das metodologias que podem ajudar equipes que trabalham nesse modelo.

Este trabalho tem como objetivo apresentar o *Scrum* voltado a equipes distribuídas, descrevendo suas práticas, diferenças encontradas entre equipes presenciais e distribuídas e quais ferramentas e práticas ser adotadas para minimizar esse problemas.

1.1 Objetivo Geral

Realizar um estudo sobre as principais práticas, desafios, problemas e ferramentas utilizadas por equipes remotas que utilizam a metodologia *Scrum*.

1.2 Objetivos Específicos

1. contextualização por meio de referencial teórico sobre métodos ágeis e *Scrum*;
2. identificar quais melhores práticas se aplicam em equipes remotas;
3. identificar quais as dificuldades equipes remotas enfrentam com mais frequência;
4. propor melhores alternativas para facilitar o trabalho da equipe;
5. propor ferramentas de apoio;
6. coleta de dados com profissionais da área sobre práticas de equipes remotas.

1.3 Justificativa do trabalho

Com muitas empresas T.I. hoje aderindo a forma de contratação de funcionários remotos, é preciso garantir que a equipe, seja ela em sua totalidade ou parcialidade remota, consiga trabalhar de modo ágil, sustentável, sempre focado na entrega e ainda sim mantendo a sua organização e colaboração ativa. Para isso as empresas podem adotar *Scrum* como metodologia ágil afim de organizar melhor a equipe e focar nos seus resultados.

Capítulo 2

Referencial Teórico

Neste capítulo serão apresentados os principais conceitos que serão abordados durante o trabalho.

2.1 Metodologias Ágeis

Visando minimizar os problemas do desenvolvimento de *software* sem algum planejamento é preciso adotar uma metodologia que ajude no processo de planejamento e na solução a ser desenvolvida.

Segundo (FOWLER, 2003) citado por (PASSUELO, 2005) "Metodologias impõem um processo disciplinado no desenvolvimento de software, com objetivo de torná-lo mais previsível e mais eficiente".

Porém o problema de muitas metodologias é que elas são muito burocráticas, focadas em documentação e gerenciamento rígido do projeto, muitas vezes acabam sendo vistas como um grande atraso para o projeto, já que a grande maioria do tempo do projeto é gasto levantando requisitos e documentando todas as ações do sistema, antes mesmo de ele ser codificado. (FOWLER, 2003)

Então nos anos noventa, surge uma reação a essas metodologias mais rígidas, são as chamadas metodologias ágeis no processo de desenvolvimento de *software*.

Criado por Kent Beck, com uma equipe de outros desenvolvedores, já com experiência na área, o Manifesto Ágil é o documento que reúne todas as práticas e princípios dos métodos ágeis dentro do processo de desenvolvimento (BECK, 2001). O manifesto Ágil possui valores, que são eles:

- **Indivíduos e interações** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação detalhada;
- **Colaboração com cliente** mais que negócios e contratos;
- **Responder a mudança** mais que seguir um plano;

Ou seja, mesmo havendo valor nos itens a direita, valorizamos mais os itens a esquerda. (ÁGIL, 2001)

Além dos valores, o Manifesto Ágil se baseia em alguns princípios (BECK, 2001):

- Satisfazer o cliente por meio de entregas adiantadas e contínuas do *software*;
- Aceitar as mudanças de requisito a qualquer momento dentro do projeto;
- Entrega de software em funcionamento com frequência semanal ou mensal;
- Pessoas que entendem do negócio e desenvolvedores devem estar sempre trabalhando juntas durante o processo de desenvolvimento;
- Trabalhar com pessoas motivadas no projeto, dando a eles um bom ambiente de trabalho e confiar que farão aquele que foi designado a eles;
- A melhor maneira de se transmitir informação é por meio de uma conversa pessoal;
- Software funcional é a medida primária de progresso;
- Processos ágeis promovem um ambiente sustentável;
- Contínua atenção a excelência técnica e bom design;
- Simplicidade;
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis;
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e aperfeiçoam seu comportamento de acordo.

Por meio desses princípios e valores pode-se notar que as metodologias ágeis deixam de lado a documentação pesada, processos de extração e elaboração de requisitos e passam a valorizar mais o software funcionando, liberando pequenas versões à medida que as mesmas ficam prontas, trazem o cliente para dentro do processo de desenvolvimento

fazendo o mesmo participar e acompanhar o crescimento do projeto, enfatiza interação entre pessoas sejam elas do time de desenvolvimento ou pessoas que entendem do negócio e abraçam mudanças no software a qualquer momento (FOWLER, 2003).

2.2 Scrum

2.3 Trabalho distribuído

Referências Bibliográficas

BECK, K. *Manifesto para Desenvolvimento Ágil de Software*. 2001. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em: 17 jun. 2015.

ECÔNOMIA, B. *Trabalho remoto ganha mais espaço nas empresas*. 2015. Disponível em: <<http://brasileconomico.ig.com.br/vida-e-estilo/2015-01-30/trabalho-remoto-ganha-mais-espaco-nas-empresas.html>>. Acesso em: 17 jun. 2015.

FOWLER, M. *The new Methodology*. 2003. Disponível em: <<http://martinfowler.com/articles/newMethodology.html>>. Acesso em: 17 jun. 2015.

PASSUELO, L. *A Nova Metodologia*. 2005. Disponível em: <<http://www.p3software.com.br/home/artigos/6-a-nova-metodologia>>. Acesso em: 17 jun. 2015.

ÁGIL, D. *O que é Scrum ?* 2015. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 15 jun. 2015.

ÁGIL, M. *Manifesto Ágil*. 2001. Disponível em: <<http://www.agilemanifesto.org/iso/ptbr/>>. Acesso em: 15 jun. 2015.