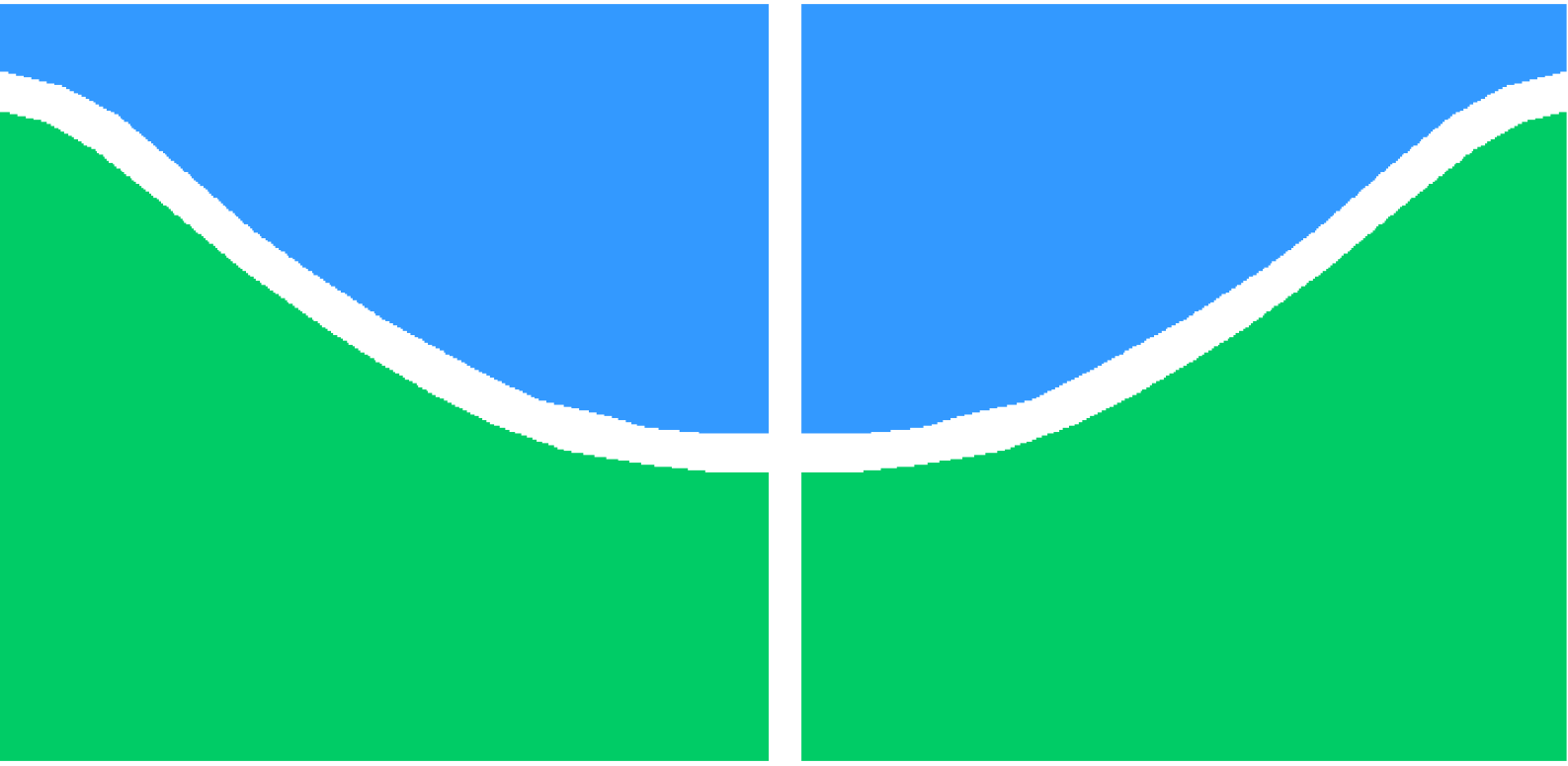

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Requisitos de Software - 201308

Justificativa de Abordagem

Grupo 01
Autores: Dylan, Eduardo, Wilton, Pedro
Orientador: George Marsicano Corrêa, MSc

Brasília, DF
2015

Dylan, Eduardo, Wilton, Pedro

Justificativa de Abordagem

Justificativa de escolha de abordagem referente à disciplina de Requisitos de Software, do curso de Engenharia de Software da Universidade de Brasília.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: George Marsicano Corrêa, MSc

Brasília, DF

2015

1 Escolha da Abordagem

No desenvolvimento de um software é imprescindível a utilização de alguma abordagem que conduza seu processo produtivo.

Uma abordagem se refere a uma metodologia que será usada para estruturar, planejar, e controlar o processo de desenvolvimento do sistema (CMS, 2005).

Logo, neste capítulo será justificada a escolha de uma determinada abordagem feita pelo grupo 1 da disciplina Requisitos de Software, responsável por realizar a manutenção de sistemas de software mantidos pelo Ministério das Comunicações. Primeiro será exposto uma visão geral de cada abordagem, explicando suas diferenças e os pontos positivos e negativos presentes em cada uma delas, para que enfim uma justificativa seja apresentada. As abordagens escolhidas para representar cada uma das correntes, tradicional e adaptativa, são o Rational Unified Process (RUP) e o Scaled Agile Framework (SAFe), respectivamente.

1.1 Abordagem Tradicional - RUP

O Processo Unificado surgiu em meados dos anos 90 e tem como proposta reunir diversas práticas e filosofias que se provaram eficientes até então no contexto de desenvolvimento de software (KRUCHTEN, 2003, p. 45).

Valoriza conceitos como arquitetura de software, desenvolvimento iterativo, gerência de requisitos, construção de modelos visuais para descrever o sistema, entre outros (KRUCHTEN, 2003, p. 45). A abordagem é baseada em disciplinas, onde cada disciplina agrupa uma série de atividades relacionadas e cada atividade produz diversos artefatos, que por sua vez contêm informações do processo e são submetidos à controle de versão (são produzidos, modificados e evoluídos durante o ciclo de vida do projeto). As disciplinas representam áreas de interesse presentes no processo de desenvolvimento de software (KRUCHTEN, 2003, p. 45).

O RUP estabelece a divisão do processo em quatro fases, sendo que cada fase possui um marco (objetivo principal) que será almejado durante sua execução. As fases por sua vez são divididas em iterações que consistem em um conjunto de atividades a serem realizadas para se obter um incremento do produto. Argumenta que o planejamento a longo prazo do projeto fornece uma visão à equipe de desenvolvimento do que deve ser feito e quando deve ser feito para que o projeto seja concluído e entregue no prazo. Analisando o processo unificado sob a perspectiva da disciplina de requisitos, é possível observar que o principal recurso para se representar requisitos dentro do processo são

os casos de uso (KRUCHTEN, 2003, p. 45). Um caso de uso descreve uma interação entre uma entidade e o sistema e demonstra uma capacidade do software, assim como as funcionalidades que serão oferecidas ao usuário (KRUCHTEN, 2003, p. 124-125). Para atingir os objetivos da ER no RUP, a disciplina de requisitos descreve como definir uma visão do sistema, traduzir essa visão em um modelo de caso de uso (que, com especificações suplementares, definirá os requisitos detalhados do software), e como usar os atributos dos requisitos para ajudar a gerenciar o escopo do projeto e como mudar os requisitos do sistema (KRUCHTEN, 2003, p. 182-183).

Por fim, o RUP é um framework implementado por grandes empresas, mas determina que sua customização é aceita para que ele se adeque à realidade de organizações de menor porte, sendo possível definir as atividades e práticas do RUP que sejam do interesse da empresa em questão, desde que não sejam feridos os seus principais valores. As grandes empresas que o utilizam não o utilizam sempre da mesma maneira: enquanto algumas o utilizam de maneira formal, outras o customizam completamente (KRUCHTEN, 2003, p. 57).

1.2 Abordagem Adaptativa - SAFe

Bem mais novo, o Scaled Agile Framework (SAFe) traz ideias novas, mas tambem aspectos já consolidados na área de metodologias de software (Scrum, Kanban, etc).

Adaptável em diversos pontos, traz a ideia de dividir o projeto em três níveis: Portfolio, Program e Team. Cada nível produz seus artefatos, dispõe de diferentes papeis, tem responsabilidades, e interagem de maneira diferente (LEFFINGWELL, 2010, p. 124-125).

Uma das características mais importantes em relação ao UP, é que, enquanto o UP é dirigido a UC, possuindo assim somente ele como descritor de comportamentos desejáveis para o sistema, no SAFe, existem três níveis de abstração para descrever tais comportamentos: o Epic, a Feature e a User Story (LEFFINGWELL, 2010, p. 182-183). O Epic é bem mais alto nível, a Feature mais baixo nível que o Epic, e a User Story mais baixo nível que ambos (LEFFINGWELL, 2010, p. 182-183).

1.3 Escolha da Abordagem e Justificativa

Considerando os conceitos levantados à respeito das duas abordagens, é possível inferir que ambas fornecem diversas práticas e filosofias que podem agregar valor ao nosso processo e podem nos conduzir ao nosso objetivo: realizar a manutenção de sistemas para o Ministério das Comunicações. Desde os primórdios do desenvolvimento de software, uma das principais causas responsável pelo fracasso de projetos é a utilização de uma

abordagem linear proveniente das engenharias tradicionais, tendo em vista que no contexto da engenharia de software na maioria das vezes não é possível determinar completamente os requisitos de um sistema que será desenvolvido, o que inviabiliza a implantação de um processo sequencial.

Ambas metodologias citadas neste documento como exemplo procuram contornar este problema através da implementação de diversos conceitos que são comprovadamente eficientes no desenvolvimento de software, tais como gerência de requisitos e desenvolvimento iterativo e incremental. As propostas de ambas metodologias são embasadas em uma série de argumentos consistentes e temos certeza que apesar de suas diferenças sob o ponto de vista da adaptabilidade e previsibilidade, ambas propõe a aplicação de diversas das melhores práticas existentes no contexto de processos de desenvolvimento de software, o que nos leva a concluir que independente da abordagem escolhida, estaríamos bem equipados para desenvolver o projeto.

Finalmente, a equipe optou pela abordagem adaptativa pelos seguintes pontos:

- Todos os integrantes apreciam e valorizam os aspectos levantados pelo manifesto ágil
- O desenvolvimento de software iterativo (no contexto adaptativo, baseado em sprints) nos permitirá entregar incrementos constantes do sistema, permitindo demonstrar a evolução contínua do software ao cliente
- A representação dos requisitos do sistema em diferentes níveis de abstração (features e estórias) nos auxiliará a compreender as necessidades do cliente, tendo em vista que é possível avaliar os requisitos tanto sobre um ponto de vista macroscópico, onde é possível avaliar as principais capacidades do produto, quanto sobre um ponto de vista microscópico, onde é possível definir requisitos que podem ser entregues em uma única iteração
- A divisão interna do nível de time é considerada interessante pela equipe, levando em conta a importância de um ScrumMaster em qualquer organização e os benefícios trazidos pela proximidade entre uma pessoa que tenha autoridade para representar os clientes (Product Owner) e a equipe de desenvolvimento. A equipe considera essa relação e interação crucial para a execução do processo.

Apesar da experiência da equipe em projetos que utilizam uma abordagem adaptativa, ela admite que os principais frameworks de metodologias ágeis são aplicáveis à equipes de pequeno e médio porte. A promessa do SAFe de estabelecer uma abordagem adaptativa para organizações de grande porte estimulou nossa curiosidade e levando em consideração o primeiro ponto apresentado, nós valorizamos a coragem para implementar um framework (SAFe) nunca antes utilizado por nenhum integrante da equipe.

É importante ressaltar que valorizamos ambas as abordagens e reconhecemos os benefícios trazidos por elas. Portanto, procuramos enaltecer os pontos positivos da abordagem escolhida, e ressaltamos que os pontos levantados não são uma resposta direta à abordagem tradicional (não é porque um ponto está presente na abordagem adaptativa que ele não é proposto pela abordagem tradicional). Preferimos construir nossa justificativa em cima dos aspectos considerados beneficentes à nós, no lugar de construí-la em cima dos aspectos negativos da abordagem tradicional.

Referências

CENTERS FOR MEDICARE & MEDICAID SERVICES. *Selecting a Development Approach*. [S.l.], 2005. Disponível em: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>. Citado na página 3.

KRUCHTEN, P. *The Rational Unified Process: An Introduction*. [S.l.], 2003. Citado 2 vezes nas páginas 3 e 4.

LEFFINGWELL, D. *Agile Software Requirements*. [S.l.], 2010. Citado na página 4.