



UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

MONOGRAFIA DE GRADUAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Análise sobre a Implantação do Scrum de Acordo com o Perfil Organizacional

KESIA OLÍMPIO BRAGA

NOVA IGUAÇU-RJ
2017

Análise sobre a Implantação do Scrum de Acordo com o Perfil Organizacional

Monografia apresentada à Coordenação do curso de Ciência da Computação do Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro *Campus* Nova Iguaçu, como requisito parcial para obtenção do título de **Bacharel em Ciência da Computação**.

Este exemplar corresponde à redação final da monografia intitulada ANÁLISE SOBRE A IMPLANTAÇÃO DO SCRUM DE ACORDO COM O PERFIL ORGANIZACIONAL devidamente corrigida e defendida por **Kesia Olimpio Braga**, matrícula 201178010-1 e aprovada pela Banca Examinadora.

Nova Iguaçu, dia 21 de Fevereiro de 2017.

BANCA EXAMINADORA:

Prof. D. sc. Natália Chaves Lessa Schots
Orientador

Prof. D. sc. Lúgia Maria Soares Passos

Prof. D. sc. Luís Fernando Orleans

Nova Iguaçu-RJ
2017

“Ora, aquele que é poderoso para fazer tudo muito mais abundantemente além daquilo que pedimos ou pensamos, segundo o poder que em nós opera.”

Efésios 3:02

Dedicatória

Dedico esta, bem como todas as minhas demais conquistas, ao meu Deus, aos meus pais e ao meu esposo Cláudio Henrique.

Agradecimentos

Quero agradecer, primeiramente, ao meu bondoso Deus, pela graça e força concedidas durante toda esta longa caminhada. Por ter me sustentado e permitido conquistar mais esta vitória.

Sou grata também aos meus pais Edson e Regina, e ao meu esposo Cláudio Henrique por todo apoio, paciência e compreensão. Agradeço pelas palavras de ânimo e todo esforço realizado para me ajudar.

Agradeço a todos os meus amigos pela paciência e auxílio nos momentos difíceis. Todas as horas que passamos juntos estudando ou nos divertindo, ficará para sempre em minha memória. Tenho todos guardados em meu coração.

A todos os professores que me acompanharam durante a graduação, fornecendo um ensino de qualidade e contribuindo para minha formação acadêmica. Em especial, quero agradecer a Prof^a Natália Chaves Lessa Schots por toda dedicação e ajuda fornecida para a realização deste trabalho. E também aos professores Ronaldo Goldschmidt, Ligia Maria e Isabel Fernandes pela orientação na monitoria. Não menos importante, agradeço a Valéria, secretária do departamento, por todas as vezes que me ajudou em momentos de urgência.

Resumo

Dado o aumento da globalização e da competitividade das organizações, no contexto de desenvolvimento de *software*, tem ocorrido uma busca por métodos que sejam eficientes e que atendam rapidamente às necessidades dos clientes. A princípio eram utilizados apenas os métodos tradicionais, porém em meados dos anos 90, surgiram como opção os métodos ágeis.

Atualmente, há no mercado vários métodos ágeis, porém o foco deste trabalho é o *framework* Scrum, voltado para a gerência de projetos, com conceitos embasados no Manifesto Ágil. O Scrum apresenta uma abordagem que possibilita a otimização e melhoria de desenvolvimento de *software*, fornecendo benefícios às organizações que obtêm sucesso em sua implantação. Porém, a questão é se este *framework* é adequado para um determinado perfil organizacional.

Sendo assim, a proposta inicial deste trabalho foi identificar, se possível, um perfil organizacional adequado ao Scrum, auxiliando organizações que pretendem adotar o *framework*. A fim de alcançar este objetivo foi realizado um estudo teórico a partir de publicações de organizações de desenvolvimento de *software* que implantaram o Scrum. Além disto, foi aplicado um questionário e realizada uma entrevista com profissionais da área.

Com base no estudo realizado, não foi possível determinar um perfil organizacional que seja mais apropriado ao Scrum. Portanto, infere-se que o *framework* pode ser implantado por qualquer organização e projeto, desde que o contexto seja propício à agilidade. Porém, a partir dos resultados deste trabalho foi possível avaliar elementos estruturais, analisando sua relevância para a implantação do Scrum.

PALAVRAS-CHAVE: Scrum, *Framework*, Desenvolvimento de *Software*, Perfil Organizacional.

Abstract

Due to the increasing of globalization and competitiveness of organizations, in the software development context, there has been a search for methods that are efficient and that quickly find the customers needs. In the beginning, only traditional methods were used, but in the mid-1990s agile methods emerged as an option.

There are currently on the market various agile methods, but the focus of this work is the Scrum framework, focused on project management, with concepts based on the Agile Manifesto. Scrum presents an approach that enables the optimization and improvement of software development, providing benefits for the organizations that are successful in their deployment. However, the question is whether this framework is appropriate for a particular organizational profile.

In this sense, an initial proposal of this work was to identify, if possible, an organizational profile appropriate to Scrum, assisting organizations that intend to adopt this framework. In order to achieve this goal, a theoretical study was carried out from publications of software development organizations that implemented Scrum. Furthermore, was applied a questionnaire and was conducted an interview with professionals in the area.

Based on the study it was not possible to determine an organizational profile that is more appropriate to Scrum. Therefore, it is possible to infer that the framework can be deployed by any organization and project, as long as the context is propitious to agility. However, from the results of this work it was possible to assess structural elements, analyzing their relevance for the implementation of Scrum.

KEYWORDS: Scrum, Framework, Software Development, Organizational Profile.

RESUMO	vi
ABSTRACT	vii
SUMÁRIO	viii
LISTA DE FIGURAS	xii
LISTA DE TABELAS	xiii
LISTA DE ABREVIATURAS	xiv
1 INTRODUÇÃO	1
1.1 Motivação	1
1.2 Objetivos	3
1.3 Estrutura do texto	3
2 REVISÃO LITERÁRIA	5
2.1 Introdução	5
2.2 Desenvolvimento ágil de software	6
2.2.1 Desenvolvimento ágil x desenvolvimento tradicional	6
2.2.2 Método ágil: práticas e características	7
2.2.3 Principais métodos ágeis	11
2.3 Scrum	15
2.3.1 Pilares do Scrum	16
2.3.2 Papéis do Scrum	17
2.3.3 Eventos do Scrum	18
2.3.4 Artefatos do Scrum	21
2.4 Processo de Implantação do Scrum	22

2.4.1	Principais dificuldades encontradas	24
2.4.2	Principais benefícios alcançados	26
2.5	Considerações finais	27
3	ELEMENTOS ESTRUTURAIS DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	28
3.1	Introdução	28
3.2	Ambiente Organizacional	29
3.2.1	Tamanho	29
3.2.2	Nível de investimento em tecnologia	30
3.2.3	Nível de maturidade	30
3.2.4	Interação com o cliente	31
3.2.5	Estrutura organizacional	31
3.2.6	Porte de projeto	32
3.3	Equipe de desenvolvimento	33
3.3.1	Nível de especialização	33
3.3.2	Tamanho da equipe	33
3.3.3	Grau de autonomia da equipe	34
3.3.4	Localização da equipe	34
3.3.5	Experiência da equipe	34
3.3.6	Comunicação entre os membros da equipe	35
3.4	Produto de software	35
3.4.1	Nível de documentação exigida	35
3.4.2	Mudança de escopo	36
3.5	Aplicação no Scrum	36
3.6	Considerações finais	39
4	ANÁLISE TEÓRICA: SCRUM NAS ORGANIZAÇÕES	40
4.1	Introdução	40
4.2	Mapeamento sistemático	41
4.2.1	Definição de escopo e análise preliminar	42
4.2.2	Determinação do protocolo	42
4.2.3	Execução da pesquisa	44
4.3	Análise dos resultados	45
4.3.1	Ambiente organizacional	50
4.3.2	Equipe de desenvolvimento	52
4.3.3	Produto de <i>software</i>	54
4.3.4	Comparação	55
4.4	Considerações finais	58

5	ENTREVISTA: SCRUM NAS ORGANIZAÇÕES	59
5.1	Introdução	59
5.2	Planejamento	60
5.2.1	Questionário	61
5.2.2	Entrevista	61
5.3	Execução	62
5.4	Análise do questionário	63
5.5	Análise da entrevista	67
5.5.1	Considerações relacionadas ao perfil organizacional e aos elementos estruturais estudados	68
5.5.2	Considerações em relação ao desenvolvimento de <i>software</i> em geral	71
5.6	Considerações finais	72
6	CONCLUSÃO	75
6.1	Considerações finais	75
6.2	Contribuições	76
6.3	Limitações	77
6.4	Trabalhos futuros	77
	REFERÊNCIAS BIBLIOGRÁFICAS	78
A	QUESTÕES DO QUESTIONÁRIO	86
B	QUESTÕES DA ENTREVISTA	89
C	RESUMO DO QUESTIONÁRIO	91

LISTA DE FIGURAS

2.1	Ciclo de vida do Scrum (Adaptado) [PEREIRA et al. 2013]	16
2.2	<i>Planning poker</i> [SAMI 2014]	22
2.3	Gráfico <i>Burndown</i> [SOUZA 2013]	22
3.1	Grau de maturidade da organização de acordo com os níveis de maturidade do modelo PMMM [SILVEIRA et al. 2013, VERGOPIA 2008]	30
3.2	Característica do projeto de acordo com a estrutura organizacional [PMBOK 2013]	32
4.1	Legenda da tabela de análise	50
4.2	Tamanho	51
4.3	Nível de Investimento em TI	51
4.4	Nível de maturidade	51
4.5	Interação com o cliente	51
4.6	Estrutura organizacional	52
4.7	Porte de projeto	52
4.8	Nível de especialização	53
4.9	Tamanho da equipe	53
4.10	Grau de autonomia da equipe	53
4.11	Localização da equipe	53
4.12	Experiência da Equipe	54
4.13	Nível de comunicação entre a equipe	54
4.14	Nível de documentação exigida	55
4.15	Mudança de escopo	55
5.1	Tamanho da organização	64
5.2	Nível de investimento em TI	64
5.3	Nível de maturidade	64

5.4	Interação com o cliente	64
5.5	Estrutura organizacional	65
5.6	Porte do projeto	65
5.7	Nível de especialização da equipe	65
5.8	Experiência da equipe	65
5.9	Tamanho da equipe	66
5.10	Grau de autonomia da equipe	66
5.11	Comunicação	66
5.12	Localização da equipe	66
5.13	Nível de documentação exigida	67
5.14	Nível de mudança de escopo	67
5.15	Nível de impacto dos elementos estruturais: ambiente organizacional	68
5.16	Nível de impacto dos elementos estruturais: equipe de desenvolvimento	69
5.17	Nível de impacto dos elementos estruturais: produto de <i>software</i>	71
C.1	Impacto dos elementos na opinião dos participantes	92

LISTA DE TABELAS

2.1	Práticas de agilidade	8
2.2	Características de agilidade	10
2.3	Divisão por cores do método Crystal [SOUZA 2013]	14
2.4	Dificuldades na implantação das práticas gerenciais do Scrum [VASCONCELOS e PEREIRA 2012]	25
2.5	Vantagens na implantação das práticas gerenciais do Scrum [VASCONCELOS e PEREIRA 2012]	26
3.1	Classificação da organização de acordo com a receita operacional bruta anual [BNDES 2011]	29
3.2	Classificação do porte da empresa segundo o SEBRAE [SEBRAE 2013].	30
3.3	Perfil organizacional adequado ao Scrum de acordo com [BARBOSA e GREGÓRIO 2014]	37
4.1	Resultado quantitativo das publicações obtidas via pesquisa informal	42
4.2	Referência dos Relatos	45
4.3	Elementos estruturais por publicação	48
4.4	Tabela comparativa entre o resultado do mapeamento sistemático e [BARBOSA e GREGÓRIO 2014]	56
5.1	Resumo geral da análise	73

LISTA DE ABREVIATURAS

DSDM	<i>Dynamic System Development Methodology</i>
FSMA	Faculdade Salesiana Maria Auxiliadora
PMMM	<i>Project Management Maturity Model</i>
RUP	<i>Rational Unified Process</i>
TI	Tecnologia da Informação
XP	<i>Extreme Programming</i>

CAPÍTULO 1

INTRODUÇÃO

Este capítulo apresenta o contexto que motivou este trabalho, assim como os objetivos que se pretende alcançar com sua realização. Além disto, traz o modo como está estruturado.

1.1 Motivação

Em consequência da globalização e da competitividade entre as organizações, tem ocorrido um aumento da procura por *softwares* de qualidade com o objetivo de melhorar os negócios e auxiliar na tomada de decisões. Neste contexto, houve a necessidade de definir métodos padronizados, organizados e estratégicos de desenvolvimento de *software* [JUNIOR e SANTOS 2010], e há anos desenvolvedores têm modificado e adaptado métodos de desenvolvimento a fim de encontrar o método ideal [SOMMERVILLE 2003]. Estes métodos devem ser menos custosos e mais rápidos com relação aos métodos tradicionais, e devem ser capazes de reduzir o índice de riscos e falhas durante o processo de desenvolvimento, assim como aumentar e valorizar a satisfação dos clientes com a garantia de entrega de *software* de qualidade.

Segundo Sommerville (2013), um método de desenvolvimento de *software* descreve de modo mais simples a perspectiva expressa pelo processo de desenvolvimento de *software*¹. Os métodos comumente possuem fases e etapas pré-estabelecidos, e apresentam um conjunto de atividades associadas ao produto de *software*, assim como ao processo de desenvolvimento e às pessoas envolvidas [SOMMERVILLE 2003].

Grande parte dos métodos de desenvolvimento de *software* são embasados em um dos três seguintes paradigmas² de desenvolvimento: modelo cascata; desenvolvimento ite-

¹Conjunto de atividades e resultados relacionados às mesmas, que tem como intuito produzir um produto de *software* [SOMMERVILLE 2003]

²<https://www.dicio.com.br/paradigma/>

rativo; e desenvolvimento baseado em componentes [SOMMERVILLE 2003]. Exemplos de métodos derivados de alguns destes paradigmas são: Prototipação; Desenvolvimento em espiral; Entrega incremental; entre outros.

Estes métodos, principalmente o modelo em cascata e os derivados deste, são caracterizados por sua natureza inflexível, burocrática, sequencial e orientada a dados e processos [JUNIOR e SANTOS 2010]. Comumente chamados de métodos tradicionais, são considerados demasiadamente “pesados”, pois de acordo com suas normas o sucesso do projeto só pode ser obtido caso todo o processo de desenvolvimento de *software* ocorra como planejado e o produto final seja entregue [JUNIOR e SANTOS 2010].

Por consequência da rigidez dos métodos tradicionais e também devido à necessidade de métodos adaptativos que fossem capazes de suportar as constantes mudanças de requisitos, surgiram, em meados dos anos 90, os métodos ágeis [SOUZA 2013]. Estes são mais flexíveis e focados nas pessoas, e realizam entregas por incrementos. Apesar do modelo iterativo também fazer uso de incrementos em suas entregas, nos métodos ágeis o processo realizado para construção de cada incremento é diferente [SOMMERVILLE 2003] [SOUZA 2013].

De acordo com Mainart e Santos (2010), os métodos ágeis auxiliam no aprendizado do cliente, que aprende sobre suas necessidades e como priorizar o que deve ser desenvolvido, para que seja produzido primeiro o que possui mais valor para os negócios.

No mercado de desenvolvimento de *software*, há vários métodos ágeis disponíveis, como por exemplo: Scrum, DSDM *Dynamic Systems Development Method*, XP *Extreme Programming*, Crystal, entre outros. Neste trabalho será abordado o *framework* Scrum, voltado para a gerência de projetos e desenvolvido em meados do anos 90 por Ken Schwaber, Jeff Sutherland e Mike Beedle.

O Scrum baseia-se no Manifesto Ágil ³, seguindo seus princípios e valores. É definido como um *framework* - e não um método - pois não fornece uma fórmula perfeita capaz de solucionar todos os problemas, mas serve como um guia possibilitando que estes problemas sejam visualizados e solucionados com mais facilidade e rapidez [VARASCHIM 2009].

O *framework* Scrum tem o objetivo de otimizar o processo de desenvolvimento de produtos de *softwares* complexos, auxiliando na prevenção de riscos e falhas. Para tal, engloba pilares, papéis, eventos, artefatos e regras bem definidos no seu guia [SCHWABER e SUTHERLAND 2013].

O uso do Scrum, assim como os demais métodos, pode trazer diversos benefícios, mas não é ideal para todos os tipos de projetos de desenvolvimento de *software*. Antes de se adotar um novo método é preciso: 1) analisar todo o contexto no qual o projeto está inserido; 2) verificar o impacto da implantação do método na organização e se esta está apta para uma mudança cultural organizacional; e 3) verificar se seus colaboradores

³<http://www.manifestoagil.com.br/>

estão dispostos a aceitar e aprender novos conceitos [VARASCHIM 2009]. Além disso, é preciso analisar cuidadosamente se o método escolhido é o que realmente atenderá as necessidades do projeto e da organização, respondendo a todas as expectativas.

Neste contexto, é importante auxiliar as organizações de desenvolvimento de *software* no processo de implantação do Scrum. Se possível, a definição de um perfil organizacional que seja mais apropriado ao *framework*, pode servir como base para organizações que se englobam neste perfil, fornecendo-lhes mais segurança para a implantação.

1.2 Objetivos

Este trabalho analisa relatos de experiência de organizações de desenvolvimento de *software* com a implantação bem sucedida do *framework* Scrum a fim de identificar, se possível, um perfil organizacional que seja mais adequado à adoção do *framework* Scrum. O perfil é analisado a partir dos elementos estruturais do processo de desenvolvimento de *software* propostos por Barbosa e Gregório (2014) divididos em três visões: Ambiente Organizacional, Equipe de Desenvolvimento e Projeto de *Software*.

Especificamente, tem-se os seguintes objetivos:

- realizar um estudo teórico de publicações que descrevam a experiência de organizações de desenvolvimento de produto de *software* com a implantação bem sucedida do *framework* Scrum;
- a partir das publicações identificadas, analisar o perfil organizacional das organizações relatadas antes da implantação do Scrum, com base nos elementos estruturais adotados nesta pesquisa;
- comparar os resultados obtidos no mapeamento sistemático com a classificação fornecida, para o Scrum, pelos autores Barbosa e Gregório (2014);
- obter a opinião de especialistas na área a fim de validar os resultados obtidos e agregar sua experiência.

1.3 Estrutura do texto

Este trabalho está estruturado em 6 capítulos. Este capítulo apresentou a motivação para a execução deste trabalho, assim como o objetivo da pesquisa e a estrutura do texto.

O segundo capítulo apresenta a revisão da literatura, mostrando uma breve comparação dos métodos ágeis com os métodos tradicionais. Além disso, alguns dos principais métodos ágeis, suas características e práticas serão apresentados, em especial o *framework*,

Scrum, escolhido para ser abordado no presente trabalho.

O terceiro capítulo traz alguns elementos que compõem a estrutura geral do processo de desenvolvimento, que serão utilizados como base para a identificação do perfil organizacional. Além disto, apresenta a visão de Barbosa e Gregório (2014) quanto à adequação destes elementos ao Scrum.

O quarto capítulo apresenta o estudo realizado a fim de alcançar o objetivo deste trabalho, que pretende identificar um perfil organizacional adequado ao Scrum. O capítulo apresenta o mapeamento sistemático realizado bem como a análise dos resultados obtidos.

O quinto capítulo apresenta o questionário aplicado e a entrevista semi-estruturada realizada com especialistas da área. Além de apresentar o planejamento da entrevista, bem como a execução e a análise dos resultados obtidos.

O sexto capítulo expõe as conclusões obtidas a partir do desenvolvimento do presente trabalho, assim como as limitações encontradas, suas contribuições e os trabalhos futuros pretendidos.

CAPÍTULO 2

REVISÃO LITERÁRIA

Este capítulo apresenta sucintamente como surgiu o desenvolvimento ágil de *software* e alguns fatores que o distingue do desenvolvimento tradicional. Além disso, são descritos alguns dos principais métodos ágeis, assim como, algumas das suas práticas e características. Também é abordado com mais detalhes o *framework* Scrum, selecionado como objeto de estudo neste trabalho.

2.1 Introdução

O *software* tem se tornado cada vez mais indispensável, estando inserido em praticamente todas áreas de nossas vidas e impulsionando os negócios organizacionais [LEIDEMER 2014]. De acordo com [SOMMERVILLE 2003], a Engenharia de *Software* é uma área da engenharia de sistemas¹ que lida com todos os aspectos relacionados à construção do *software*, da concepção até a manutenção, após este ser posto em uso [LEIDEMER 2014].

A Engenharia de *Software* controla e organiza os processos de desenvolvimento de produto de *software* a fim de evitar o fracasso do projeto. Um método moderno da Engenharia de *Software* precisa ser ágil, de modo que sejam apenas realizadas atividades e gerados artefatos necessários, evitando assim o excesso e perda de tempo com a construção de artefatos que não serão utilizados [SOUZA 2013].

A procura por aumento de percentual de produtos de *software* desenvolvidos com sucesso estimulou engenheiros de *software* a encontrarem métodos de desenvolvimento melhores e mais eficazes [SOUZA 2013]. Esta procura levou ao surgimento dos métodos ágeis, que são considerados mais “leves” e menos burocráticos que os métodos tradicio-

¹A engenharia de sistema relaciona-se com todos os fatores associados ao desenvolvimento e evolução de sistemas. Esta engloba: desenvolvimento de *hardware*, engenharia de *software*, entre outras áreas [SOMMERVILLE 2003].

nais [SOUZA 2013].

A seção 2.2 apresenta mais detalhadamente alguns aspectos dos métodos ágeis. Já a seção 2.3 traz o *framework* que será o foco deste trabalho, o Scrum. Na seção 2.4 será apresentado sucintamente um processo para implantação do *framework*, assim como, alguns benefícios e dificuldades encontradas na adoção.

2.2 Desenvolvimento ágil de software

“À medida que a importância do software cresceu, a comunidade de programadores tem continuamente tentado desenvolver tecnologias que tornem mais fácil, mais rápido e menos dispendioso construir e manter programas de computador de alta qualidade [SOMMERVILLE 2003].”

2.2.1 Desenvolvimento ágil x desenvolvimento tradicional

Os métodos de desenvolvimento de produto de *software* se caracterizam por seu processo dividido em etapas e fases, que comumente compreendem atividades tais como Análise, Modelagem, Desenvolvimento e Testes [UTIDA 2009].

Os métodos tradicionais, também conhecidos como “pesados”, são considerados demasiadamente burocráticos e lentos, pois definem que as etapas e fases do processo devem seguir rigorosamente as normas estabelecidas pelo método e focam em gerar documentação sobre o projeto [VASCONCELOS e PEREIRA 2012, MASSIMUS 2016]. Sendo assim, são muito utilizados em cenários onde os requisitos do sistema são estáveis e os requisitos futuros são previsíveis [PRIKLADNICKI, WILLI e MILANI 2014, UTIDA 2009]. Além disto, nestes métodos, considera-se o sucesso do projeto somente no final, quando o produto de *software* é entregue por completo; isto ocorre mesmo nos métodos iterativos onde ocorrem entregas parciais e são aprovadas alterações no escopo [MASSIMUS 2016]. Alguns exemplos de métodos tradicionais são: Modelo Cascata, Modelo Espiral, Modelo Incremental, entre outros [UTIDA 2009].

Em meados dos anos 90, em resposta à rigidez dos métodos tradicionais, surgiram os métodos ágeis que são mais adaptativos e flexíveis em relação aos métodos tradicionais, concentrando-se na frequente entrega de produto e nas interações com os indivíduos e não no processo [VASCONCELOS e PEREIRA 2012]. No mercado de desenvolvimento de *software* atual, é difícil definir os requisitos de *software* antes do início do projeto e prever como o mesmo evoluirá com o tempo, dada as constantes mudanças a fim de suprir o aumento da concorrência e a necessidade dos usuários finais [SOMMERVILLE 2003]. Nestes casos, o mais indicado é o uso dos métodos ágeis que oferecem respostas rápidas às constantes mudanças de requisitos e garantem entregas frequentes de *software* de alta qualidade [VASCONCELOS e PEREIRA 2012]. Nos métodos ágeis considera-se cada en-

trega parcial, ou seja, cada incremento como um produto pronto de qualidade e eficiente; sendo assim, pode-se verificar o sucesso gradual no projeto e não apenas no final [RUOSO 2009].

Em 2001, formalizou-se a criação dos métodos ágeis em um encontro que reuniu especialistas com objetivo de discutir experiências, práticas e técnicas concernentes ao desenvolvimento de *software*, a fim de melhorar o desempenho de seus projetos [LEIDEMER 2014]. Como resultado publicou-se o “Manifesto Ágil”, que descreve os princípios básicos e valores dos métodos ágeis utilizados para o sucesso em projetos [LEIDEMER 2014]. São quatro os valores fundamentais descritos no “Manifesto Ágil” e que também diferenciam os métodos ágeis dos métodos tradicionais [BECK et al. 2001]:

- indivíduos e interação são mais importantes do que processos e ferramentas;
- *software* em funcionamento é mais importante do que uma documentação abrangente;
- colaboração com o cliente é mais importante do que negociação de contratos;
- responder a mudanças é mais importante do que seguir um plano.

2.2.2 Método ágil: práticas e características

“É importante ressaltar que o desenvolvimento ágil prega o simples, entretanto, ser simples implica em grande carga de disciplina e organização [VARASCHIM 2009]”.

No sentido literal, ser ágil consiste em ser veloz e agilidade significa agir com velocidade. Porém, de acordo com [MENDES 2011], estes conceitos aplicados no ambiente de Tecnologia da Informação possuem a seguinte definição: ser ágil é executar tarefas de forma eficiente em um intervalo específico, baseando-se em padrões e princípios predefinidos; e agilidade é o ato de gerar ou adaptar de modo hábil e criativo meios que asseguram a execução ágil de produtos ou serviços.

Há um conjunto de práticas e características que permitem identificar se um processo de desenvolvimento de *software* é ágil ou não, e que auxiliam em caso de interesse de implantação de agilidade em processos de *software*. As práticas de agilidade, são atividades ou hábitos que executam os princípios adotados pelos métodos ágeis [ABRANTES e TRAVASSOS 2013]; enquanto as características são particularidades ou atributos importantes que um método pode apresentar para que se diferencie de outros métodos, levando em consideração uma perspectiva ágil [ABRANTES e TRAVASSOS 2013]. Nas Tabelas 2.1 e 2.2 são apresentadas as práticas realizadas e as características comumente encontradas nos métodos ágeis, respectivamente.

Tabela 2.1: Práticas de agilidade

Práticas	Descrição
Relacionadas com os requisitos do produto	
<i>Backlog</i> do produto	Lista com as funcionalidades que precisam ser desenvolvidas; vale ressaltar que esta lista está em constante mudança, seja por inserção de novos requisitos, manutenção ou refatoração [MELLO, SILVA e TRAVASSOS 2014].
<i>Design</i> simples	O <i>design</i> do <i>software</i> deve ser o mais simples possível, de modo que seja priorizado o que deve ser feito na iteração corrente; e o que pode ser útil no futuro deve ser desenvolvido no futuro [TELES 2006].
Relacionadas com a fase de construção do produto	
Programação em par	Todo e qualquer código é implementado por dois desenvolvedores que trabalham juntos revezando o mesmo computador [TELES 2006].
Código coletivo	Todos da equipe de desenvolvimento possuem autorização para editar qualquer parte do código do <i>software</i> a qualquer momento [TELES 2006].
Integração contínua	Consiste em integrar diversas vezes ao dia o trabalho realizado pela equipe para a elaboração do <i>software</i> , assegurando que a base de código permaneça consistente ao final de cada integração [TELES 2006].
Refatoração de código	A estrutura do <i>software</i> tende a se degradar com o passar do tempo, de acordo com a inserção de novas funcionalidades, alterações e correções de erros. A fim de evitar que o sistema se torne difícil de manter, a equipe aplica a refatoração que consiste em alterar constantemente pequenas partes do código sempre que surgir a oportunidade de melhorá-lo [TELES 2006].
Metáfora do sistema	São usadas por desenvolvedores ao criar elementos dentro do computador para simular outros elementos que existem no mundo físico [TELES 2006].
Pequenas liberações	Durante o desenvolvimento do produto devem ser realizadas pequenas entregas com partes funcionais do produto [AMÉRICO 2004].

Tabela 2.1 – Práticas de agilidade (continuação)

Práticas	Descrição
Relacionadas com a fase de construção do produto	
Padrões de codificação	Toda a equipe de desenvolvimento tem acesso ao código podendo alterar e refatorar a qualquer momento, sendo assim é importante que seja mantido um padrão de codificação, isto é, todos programem da mesma forma, a fim de facilitar o entendimento e alteração do código [AMÉRICO 2004].
<i>Test Driven Development</i>	O projeto o desenvolvimento do <i>software</i> é baseado em testes. Os testes são automatizados e rodam repetidamente, de modo que seus resultados são utilizados para analisar o desenvolvimento [BORGES 2010].
Relacionadas com a organização e ambiente de trabalho	
Interação com o cliente	Manter uma relação próxima com o cliente é importante, pois auxilia na redução de falhas e no aumento do grau de satisfação do mesmo [MELLO, SILVA e TRAVASSOS 2014].
Ritmo sustentável	Visa garantir que a equipe não trabalhe mais e sim trabalhe de forma mais inteligente, produtiva e sustentável para que assim não sofra desgaste, fadiga e déficit de atenção [TELES 2006].
Time multidisciplinar	O time deve possuir todas as habilidades necessárias para que seja criado um incremento do produto [SCHWABER e SUTHERLAND 2013].
Equipe completa	Para que o projeto tenha sucesso é imprescindível a disponibilidade e presença do cliente e quaisquer outras pessoas que precisam ser ouvidas no desenvolvimento para que possam auxiliar a equipe nas dúvidas que surgirem [TELES 2006].
Relacionadas com o gerenciamento do projeto	
Visibilidade do projeto	Deve ser possível acompanhar o andamento do projeto, avaliando a qualidade do produto e se os requisitos estão bem definidos [FARIA 2011].

Tabela 2.1 – Práticas de agilidade (continuação)

Práticas	Descrição
Relacionadas com o gerenciamento do projeto	
Retrospectivas	São eventos cujo objetivo é permitir que os membros do projeto possam se inspecionar e também ao projeto, observando o que se pode melhorar [SCHWABER e SUTHERLAND 2013].
Reuniões diárias	Visam apresentar as tarefas desempenhadas no dia anterior pelos membros da equipe, assim como expor as dificuldades encontradas ao executá-las e também tem o objetivo de evidenciar as tarefas que devem ser priorizadas durante o dia. Essas reuniões retratam o atual estado do projeto [MELLO, SILVA e TRAVASSOS 2014].
Relacionadas com a fase de teste	
Testes automatizados	O nível dos testes efetuados durante o desenvolvimento do <i>software</i> é essencial para a qualidade do mesmo. Automatizar os testes diminui o trabalho do analista e aumenta a qualidade dos testes [BARBOSA e GREGÓRIO 2014].

Tabela 2.2: Características de agilidade

Características	Descrição
Testes constantes	Os testes são uma maneira confiável e ágil de se obter o <i>feedback</i> do produto, dessa maneira devem ser realizados constantemente [AMÉRICO 2004].
Equipes pequenas	Pequenas equipes são mais fáceis de gerenciar; tamanho ideal e suficiente é aquele onde a equipe consegue ser ágil e completar o trabalho dentro do limite definido [AMÉRICO 2004].
Adaptabilidade	Se for detectado algum tipo de variação no aspecto no processo de desenvolvimento, deve ocorrer a adaptação, isto é, o processo deve ser ajustado o mais breve possível, a fim de evitar danos ao projeto [SCHWABER e SUTHERLAND 2013].

Tabela 2.2 – Características de agilidade (continuação)

Características	Descrição
<i>Time-boxing</i>	Tem como objetivo realizar o desenvolvimento dentro do tempo e orçamento previstos. Divide o projeto em pequenas partes com um orçamento fixo e uma data estabelecida [TEIXEIRA et al. 2005].
Auto-organização	A equipe deve ser capaz de organizar-se sozinha, definindo a melhor maneira de desempenhar suas tarefas [MELLO, SILVA e TRAVASSOS 2014].
Colaboração	Deve haver colaboração tanto entre os membros da equipe do projeto quanto entre a equipe e o cliente [UTIDA 2009].
<i>Feedback</i>	Para que seja obtido um produto de qualidade e que corresponda a real necessidade do cliente, o mesmo deve passar constantemente por <i>feedback</i> , visando avaliar sua evolução e analisar a necessidade de melhorias [UTIDA 2009].
Incremento	É a soma de todos os itens do <i>backlog</i> do produto desenvolvidos na <i>sprint</i> atual com os itens desenvolvidos nas <i>sprints</i> anteriores [SCHWABER e SUTHERLAND 2013].
Orientação a pessoas	As pessoas são priorizadas nos métodos ágeis, de modo que, é essencial que os profissionais sejam capacitados e desenvolvam suas tarefas com qualidade [UTIDA 2009].
Reflexão e introspecção	A equipe deve ser prudente, de modo que, cada membro da equipe reflita sobre os assuntos concernentes ao projeto e o seu posicionamento em relação ao mesmo [DICIONÁRIO 2009-2016].

2.2.3 Principais métodos ágeis

As organizações de desenvolvimento de *software* possuem à sua disposição vários métodos ágeis que podem ser utilizados de acordo com suas necessidades. Os principais métodos são apresentados brevemente nas subseções a seguir. O *framework* Scrum, que é o foco deste trabalho, será apresentado na seção 2.3 com mais detalhes.

2.2.3.1 XP - Extreme Programming

O *Extreme Programming* ou XP foi criado em 1999. Surgiu como resultado de uma experiência na organização Crysler e possui como principais colaboradores Kent Beck, Ron Jeffries e Ward Cunningham [MARQUES 2012]. Várias características presentes no método foram herdadas da comunidade que seus colaboradores participavam, a comunidade *SmallTalk*², tais como: adaptação a mudanças, desenvolvimento iterativo, integração contínua e ênfase nos testes [MARQUES 2012].

O termo *Extreme Programming* vem do fato deste método reunir um conjunto de práticas já realizadas no desenvolvimento de *software*, consideradas boas práticas, e utilizá-las o máximo que puder, levando-as ao extremo [MARQUES 2012, BONATO 2002]. É um método ágil que pode ser aplicado em projetos onde as equipes são de pequeno ou médio porte, e os requisitos mudam constantemente [BONATO 2002]. Se diferencia de outros métodos por fatores como: dependência de testes automatizados produzidos pelo cliente e por desenvolvedores; dependência da colaboração entre os membros da equipe; dependência de práticas que funcionam tanto em projetos de curto prazo quanto em projetos de longo prazo; entre outros fatores [BONATO 2002].

O XP apresenta as seguintes características, apresentadas em termos de seus valores [MARQUES 2012, BONATO 2002]:

- **Simplicidade** - Manter o produto dentro dos parâmetros impostos pelo cliente, ou seja, não deve ser feito nada do que não tenha sido requisitado. Mas o mais importante é não tentar solucionar problemas futuros, mantendo o *software* mais simples possível.
- **Comunicação** - Os membros da equipe devem manter uma comunicação constante, a fim de evitar erros no projeto.
- **Retorno** - Entregas frequentes de versões funcionais, onde os clientes podem dar um retorno ajudando a evitar falhas no projeto.
- **Respeito** - Respeito mútuo entre os membros da equipe, e também entre cliente e desenvolvedores.
- **Coragem** - Manter uma transparência em relação ao projeto, e em caso de necessidade de mudanças, é imprescindível realizá-las sem receio.

Os papéis sugeridos pelo XP são [MARQUES 2012, BONATO 2002]:

²linguagem de programação orientada a objetos criada em meados dos anos 1960.

- **Cliente** - Responsável pelo controle das “histórias”³, definindo prioridades, adicionando ou removendo, de modo a acompanhar o prazo definido e valorizar as “histórias”.
- **Desenvolvedor** - Estima os custos e os recursos necessários para cada “historia”, e é responsável pela criação das tarefas e seu desenvolvimento.
- **Técnico (*Coach*)** - Monitora o desenvolvimento do projeto, ajudando o time a manter o foco no que realmente é necessário.
- **Rastreador** - Responsável por analisar o progresso do time e monitorar o plano de liberação (histórias de usuário), o plano de iteração (tarefas) e os testes de aceitação.

2.2.3.2 Crystal

Desenvolvida em 1991 por Alistair Cockburn, o Crystal não é apenas um método, mas sim uma família de métodos; isto é, há vários métodos Crystal para diferentes tipos de projetos [SINÉSIO et al. 2013]. Estes métodos possuem um “código genético” em comum, ou seja, internamente possuem os mesmos princípios, prioridades, além de compartilhar propriedades como: entrega frequente, comunicação, interatividade, foco por parte da equipe, entre outras [SINÉSIO et al. 2013].

A família Crystal prioriza a segurança no resultado final do projeto, de modo que seja garantida a entrega do *software*. As práticas e convenções adotadas no processo de desenvolvimento devem ser aceitas pela equipe de modo que o processo de desenvolvimento seja realizado de maneira eficiente [SINÉSIO et al. 2013].

Há um método específico para cada tipo de projeto, sendo assim ele pode ser adaptado de acordo com a necessidade e complexidade do projeto, sendo adotadas certas políticas e convenções em cada situação [SINÉSIO et al. 2013].

Projetos menores, com menos desenvolvedores e impasses, tendem a ter um nível crítico menor e causam menos prejuízo em caso de mal funcionamento, já os projetos maiores com uma equipe mais robusta tendem a ter um prejuízo maior, exigindo um aumento da coordenação e integração da equipe [LEIDEMER 2014] [SINÉSIO et al. 2013]. Há um método Crystal apropriado para cada projeto de acordo com sua criticidade e tamanho. Para definir o método adequado são utilizadas cores [LEIDEMER 2014]. A Tabela 2.3 apresenta a divisão por cores da família Crystal [LEIDEMER 2014].

³Histórias - são os acordos dos requisitos entre os clientes e os desenvolvedores.

Tabela 2.3: Divisão por cores do método Crystal [SOUZA 2013]

Cor	Número de desenvolvedores	Prejuízo em caso de falha
Clear	1-6	Perde dinheiro, porém a recuperação é rápida.
Yellow	7-20	Perde dinheiro discretamente.
Orange	21-4	Perde dinheiro substancialmente.
Red	41-100	Perde dinheiro substancialmente e até consequências mais drásticas, quanto a vidas.

O “código genético” da família Crystal também engloba um item definido como “Modelo de jogo econômico-cooperativo”, que define o desenvolvimento de *software* como uma sequência de “jogos”, cujos movimentos básicos são invenção e comunicação [SINÉSIO et al. 2013].

A invenção no sentido que cada jogo é específico, de modo que sua estratégia é ligeiramente diferente da aplicada nos jogos anteriores; isto é, jogos não se repetem. E a comunicação que deve ser constante entre os membros da equipe, a fim de manter a troca de informações [SINÉSIO et al. 2013].

Este “jogo” permite que as pessoas pensem no trabalho de modo mais focado, específico e eficaz [SINÉSIO et al. 2013]. No modelo de jogo econômico-cooperativo há dois objetivos a serem alcançados [MARQUES 2012]: 1) Entrega de software; e 2) Preparo para a próxima etapa do jogo.

2.2.3.3 DSDM (Dynamic System Development Methodology)

“O princípio fundamental por trás deste método está no fato de que, em vez de fixar as funcionalidades do produto e somente depois ajustar o tempo e os recursos, é melhor fixar o tempo e os recursos para depois ajustar o número de funcionalidades adequadamente [TEIXEIRA et al. 2005].”

A Metodologia de Desenvolvimento Dinâmico foi desenvolvida nos anos 90 por um consórcio de vendedores e peritos em Sistemas de Informação e formulada originalmente do método RAD (*Rapid Application Development*) [TEIXEIRA et al. 2005]. A DSDM prioriza a interação entre as partes envolvidas, buscando desenvolver um *software* de qualidade que se aproxima da real e atual necessidade do negócio, sem ultrapassar os limites de tempo e custo estimados para o projeto. Desse modo, sua principal aplicação

é em projetos que possuem cronograma e orçamento apertados [TEIXEIRA et al. 2005].

Antes de implantar o método DSDM o projeto deve possuir alguns pré-requisitos, tais como: deve haver comunicação entre os membros da equipe, clientes e usuários finais; o projeto deve ser decomposto com facilidade em partes menores; e os requisitos devem ser definidos com clareza [TEIXEIRA et al. 2005]. O método DSDM não é apropriado para projetos que não possuam estes pré-requisitos, pois estes podem não se adaptar às exigências do método [TEIXEIRA et al. 2005].

A DSDM apresenta como principais conceitos:

- *Timeboxing* - que consiste em uma técnica que tem como objetivo realizar o desenvolvimento dentro do tempo e orçamento previstos, dividindo o projeto em pequenas partes com um orçamento fixo e uma data estabelecida;
- MosCoW - modo de representar as prioridades das tarefas realizadas; e
- *Workshop* - técnica para discutir as funcionalidades do projeto; além dos conceitos de prototipagem e testes [TEIXEIRA et al. 2005].

O método DSDM possui três fases, sendo elas: pré-projeto, projeto e pós-projeto. Na fase de pré-projeto é feita a identificação do projeto no qual será realizado o método, logo após é realizado um plano de financiamento e é assegurado um compromisso de realização [TEIXEIRA et al. 2005]. A fase de projeto contém um ciclo de vida de cinco níveis que deverão ser percorridos pela equipe de desenvolvimento a fim de criar o sistema. Os cinco níveis são: estudo de viabilidade, estudo de negócio, análise funcional, desenho e implementação. Os níveis Estudo de Viabilidade e Estudo de Negócio são sequenciais e complementares entre si, já nos níveis Análise Funcional, Desenho e Implementação o sistema é desenvolvido de maneira iterativa e incremental [TEIXEIRA et al. 2005]. Na última fase, pós-projeto, assegura-se que seja entregue um sistema eficiente [TEIXEIRA et al. 2005].

2.3 Scrum

O Scrum é um método de desenvolvimento ágil, voltado para a gerência de projetos que se baseia nos princípios do Manifesto Ágil e foi desenvolvido em meados dos anos 90 por Ken Schwaber, Jeff Sutherland e Mike Beedle [VASCONCELOS e PEREIRA 2012]. Porém, o termo Scrum surgiu pela primeira vez em 1986, através da publicação do artigo chamado “*The new new Product Development Game*” [TAKEUCHI e NONAKA 1986] por Ikujiro Nonaka e Hirotaka Takeuchi. Neste artigo, os autores associaram o método à estratégia Scrum do jogo de Rugby [SOUZA 2013], que consiste em reiniciar o jogo, seja após um incidente ou depois que a bola sai de jogo. O propósito é a equipe manter o jogo ativo, todos trabalhando juntos por um objetivo em comum [CARVALHO et al. 2009].

Segundo Ken Schwaber, o Scrum não é exatamente um método e sim um *framework*, pois não é previsível de modo que oferecerá uma “receita” pronta a fim de solucionar todos os problemas, mas auxiliará de modo que estes problemas sejam solucionados mais rápido, com mais facilidade e de maneira mais eficiente [CARVALHO et al. 2009]. Por ser um *framework*, o Scrum serve de guia de boas práticas para alcançar o objetivo desejado, sendo possível acrescentar várias técnicas e processos necessários ao desenvolvimento. O Scrum adota uma abordagem de trabalho iterativo e incremental [MARQUES 2012], possibilitando a realimentação, auxiliando na otimização e melhora do desenvolvimento de produtos complexos e na prevenção riscos [MARQUES 2012]. Além disso, o Scrum baseia-se no empirismo, ou seja, considera que o conhecimento é adquirido através da experiência e da tomada de decisões levando em consideração o que já se conhece [SCHWABER e SUTHERLAND 2013].

A definição de Scrum integra pilares, papéis, eventos, artefatos e regras (inseridas dos demais conceitos) que se agregam a fim de melhorar o desenvolvimento de produtos. Estes conceitos são apresentados nas subseções a seguir.

A Figura 2.1 apresenta o ciclo de vida do *framework* Scrum, no qual os principais conceitos do *framework* são apresentados.

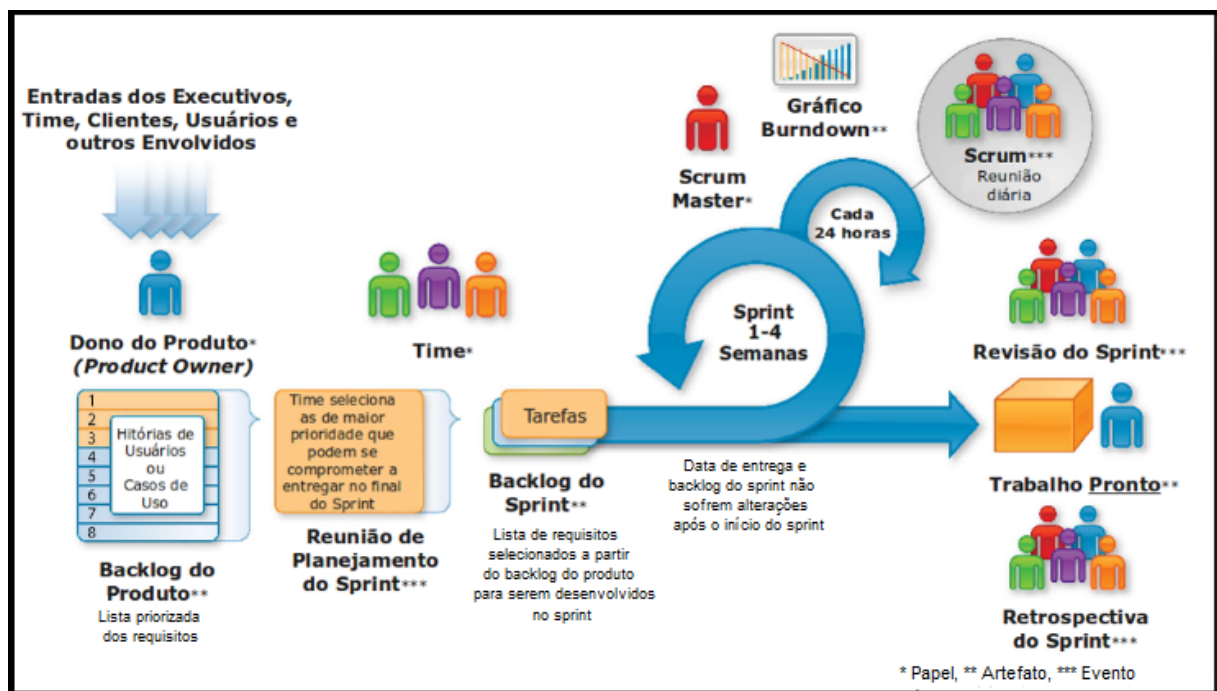


Figura 2.1: Ciclo de vida do Scrum (Adaptado) [PEREIRA et al. 2013]

2.3.1 Pilares do Scrum

O Scrum baseia-se no empirismo, afirmando que o conhecimento é obtido como resultado da experiência [SCHWABER e SUTHERLAND 2013]. Sendo assim, o *framework*

realiza um controle de processo fundamentado na observação. A implementação deste controle se apoia em três pilares: transparência, inspeção e adaptação [SCHWABER e SUTHERLAND 2013].

A transparência visa assegurar a visibilidade de aspectos relevantes do processo aos responsáveis pelos resultados, de modo que tais aspectos sejam padronizados e claros, possibilitando o mesmo entendimento por parte dos interessados [MARQUES 2012] [SCHWABER e SUTHERLAND 2013].

Durante o processo de desenvolvimento devem ocorrer frequentes inspeções aos mecanismos do Scrum e ao progresso do projeto, com o objetivo de detectar quaisquer variações inaceitáveis [MARQUES 2012]. A inspeção deve ser realizada com frequência adequada, e seus resultados devem ser analisados por pessoas especializadas para realizar tal tarefa, a fim de alcançar melhores resultados e de não impactar o fluxo de trabalho [SCHWABER e SUTHERLAND 2013].

A adaptação tem como objetivo ajustar o mais rápido possível qualquer aspecto do processo de desenvolvimento que, por meio da inspeção, tenha sido determinado inaceitável, assim evitando ou minimizando desvios posteriores [SCHWABER e SUTHERLAND 2013].

A inspeção e adaptação são realizadas no Scrum a partir de quatro eventos (a serem descritos na seção 2.3.3): reunião diária, reunião de revisão da *sprint*, reunião de planejamento da *sprint* e a reunião retrospectiva da *sprint* [SCHWABER e SUTHERLAND 2013].

2.3.2 Papéis do Scrum

A equipe do Scrum é projetada para otimizar a versatilidade, o rendimento e a criatividade durante o processo de desenvolvimento [PRIKLADNICKI, WILLI e MILANI 2014]. A equipe deve ser multifuncional e auto-organizável, sendo capaz de desempenhar todas as atividades essenciais do projeto e escolher a melhor forma de realizar seu trabalho não dependendo de pessoas que não estejam envolvidas no projeto [MARQUES 2012] [CARVALHO et al. 2009]. A equipe do Scrum é composta por poucos papéis, são eles: *Scrum Master*, *Product Owner* e o Time de Desenvolvimento [PRIKLADNICKI, WILLI e MILANI 2014].

O papel de *Scrum Master* é desempenhado por apenas um membro da equipe, que a auxilia e a organiza na adoção do Scrum [SOUZA 2013]. O *Scrum Master* é responsável por assegurar que as regras, teorias e práticas do Scrum sejam entendidas e aplicadas corretamente [CARVALHO et al. 2009]. Sendo um facilitador que auxilia na remoção de qualquer impedimento enfrentado pelo Time de Desenvolvimento [PRIKLADNICKI, WILLI e MILANI 2014], o *Scrum Master* contribui para a comunicação da equipe com o *Product Owner* e busca ferramentas e técnicas que otimizem o trabalho [SOUZA 2013].

O papel de *Product Owner* (ou dono do produto) pode ser desempenhado por apenas uma pessoa que pode ser um membro da equipe ou um cliente/usuário [SOUZA 2013]. É o responsável pela valorização do produto e do trabalho realizado pela equipe para o desenvolvimento do produto [PRIKLADNICKI, WILLI e MILANI 2014] [MARQUES 2012]. O *Product Owner* também é o responsável por tomar decisões concernentes ao produto, gerenciando o *backlog* do produto⁴. Este gerenciamento inclui ordenar e detalhar de forma clara e objetiva os itens do *backlog* do produto os tornando visíveis e transparentes, facilitando o entendimento no nível necessário por parte do Time de Desenvolvimento [SOUZA 2013] [CARVALHO et al. 2009].

O Time de Desenvolvimento tem como principal objetivo transformar o *backlog* do produto em incrementos potencialmente entregáveis ao final de cada *sprint* [MARQUES 2012]. O Time deve ser auto-organizável, isto é, deve ser capaz de organizar-se sozinho descobrindo por si só a melhor maneira de criar o incremento a partir do *backlog* do produto, facilitando que cada membro aplique suas especialidades [CARVALHO et al. 2009]. Além disso, deve ser multifuncional, ou seja, os membros do Time devem possuir todas as habilidades necessárias para criar o incremento do produto e alcançar o objetivo da *sprint* [CARVALHO et al. 2009].

2.3.3 Eventos do Scrum

O Scrum possui alguns eventos de duração fixa (*time-boxed*) realizados em intervalos regulares, a fim de criar uma rotina para o Time de Desenvolvimento [PRIKLADNICKI, WILLI e MILANI 2014]. Tais eventos visam possibilitar a inspeção e adaptação durante o processo de desenvolvimento, sendo uma forma de regularizar o processo e minimizar a necessidade de reuniões extras que não estão dentro das regras do Scrum [CARVALHO et al. 2009]. Os eventos do Scrum são: *sprint*; Reunião de planejamento da *sprint*; Reunião diária; Reunião de revisão da *sprint*; Reunião de retrospectiva da *sprint* [SOUZA 2013].

A *sprint* é o principal evento do Scrum e possui um período de tempo de duração fixa de um mês ou menos, pois se a *sprint* for muito longa pode aumentar a complexidade e risco do trabalho devido à possibilidade de mudança na descrição do que deve ser realizado [SOUZA 2013]. Neste período de tempo, um incremento do produto é criado de modo que ao final da *sprint* deve estar pronto, ou seja, potencialmente entregue ao cliente e utilizável [CARVALHO et al. 2009].

Durante uma *sprint*, os itens do *backlog* do produto selecionados e transferidos para o *backlog* da *sprint* podem ser detalhados e revisados entre o Time de Desenvolvimento e o *Product Owner*, ou até mesmo podem ser alterados desde que o objetivo da *sprint* não seja afetado [SOUZA 2013]. Cada *sprint* possui a definição do que deve ser

⁴Lista ordenada composta por todas as funcionalidades desejadas no produto [CARVALHO et al. 2009].

construído, um projeto que guia a construção, o trabalho e o resultado do produto a ser obtido [SOUZA 2013]. As *sprints* permitem a visualização periódica do progresso do projeto, garantindo a inspeção e adaptação; além disso, o risco de custo é limitado ao período estabelecido [CARVALHO et al. 2009].

A reunião de planejamento da *sprint* (ou *sprint planning meeting*) tem como objetivo planejar o trabalho a ser realizado na *sprint*, isto é, os itens que serão desenvolvidos durante a *sprint*. Para uma *sprint* de um mês, é recomendado que a reunião de planejamento da *sprint* tenha oito horas de duração; mas caso a *sprint* seja menor, a duração da reunião deve ser reduzida proporcionalmente [SOUZA 2013] [PRIKLADNICKI, WILLI e MILANI 2014]. A reunião de planejamento possui duas fases que buscam responder [CARVALHO et al. 2009]:

1. O que será entregue como resultado do incremento da próxima *sprint*?
2. Como será realizado o trabalho necessário para que este incremento seja entregue?

A primeira fase visa esclarecer o que será feito na *sprint*. Para isto, são definidos os itens que serão desenvolvidos na *sprint* e estimado o esforço que será gasto em cada item. Além disto, o Time define o objetivo da *sprint* que servirá como orientação durante o desenvolvimento, informando o motivo da criação do incremento [SOUZA 2013]. Os insumos desta reunião são: o *backlog* do produto, o incremento mais recente do produto, a capacidade projetada de produção do time de desenvolvimento e o desempenho do time durante a *sprint* anterior [CARVALHO et al. 2009].

Após selecionar os itens do *backlog* do produto e definir o objetivo da *sprint*, o Time de Desenvolvimento decide como desenvolver os itens do *backlog* do produto nesta *sprint*, detalhando a estimativa destes itens e identificando as tarefas necessárias para transformar o *backlog* do produto em um incremento do produto; estas tarefas formam o *backlog* da *sprint* [SOUZA 2013]. O *Product Owner* pode estar presente na segunda fase da reunião a fim de auxiliar na compreensão e priorização dos itens do *backlog* do produto, e também caso seja necessária a realização de trocas e renegociação, o que pode ocorrer caso sejam selecionados itens em excesso ou em falta para a *sprint* [SOUZA 2013] [MARQUES 2012].

Ao finalizar a reunião de planejamento da *sprint*, o Time de Desenvolvimento deve apresentar ao *Product Owner* e ao *Scrum Master* a forma como se organizará para atingir o objetivo do *sprint* e para criar o incremento previsto [SOUZA 2013].

A reunião diária ou *daily scrum* é uma reunião com aproximadamente 15 minutos de duração, realizada a fim de diminuir possíveis problemas. Esta reunião ocorre, diariamente, sempre no mesmo local e horário e preferencialmente, ao iniciar o expediente [SOUZA 2013]. E permite ao Time de Desenvolvimento sincronizar as atividades desenvolvidas, rever o progresso do dia anterior e planejar as atividades para as próximas 24 horas.

Durante a reunião, cada membro do Time de Desenvolvimento deve esclarecer as seguintes questões:

- O que você realizou desde a última reunião para que o objetivo do *sprint* seja alcançado?
- O que você vai realizar antes da próxima reunião para atender o objetivo da *sprint*?
- Quais obstáculos podem gerar impactos em suas atividades ou nas do time de desenvolvimento?

As reuniões diárias melhoram a comunicação e o conhecimento de todos da equipe em relação ao projeto, eliminam reuniões extras, ajudam a avaliar o progresso em direção ao objetivo da *sprint* (tornando-o mais provável de ser atingido), e auxiliam na identificação e remoção de obstáculos que possam impedir o desenvolvimento do projeto [SOUZA 2013, CARVALHO et al. 2009]. Além disso, estas reuniões destacam e promovem a rápida tomada de decisão quanto às questões envolvidas no projeto [CARVALHO et al. 2009].

Ao final de cada *sprint* é realizada a reunião de revisão (*sprint review meeting*) com participação das partes interessadas e de toda equipe envolvida no projeto. Para *sprints* de um mês, a duração fixa da reunião é de quatro horas, mas se a *sprint* possuir um período de tempo menor a duração da reunião de revisão deve ser ajustada proporcionalmente [SOUZA 2013].

O objetivo desta reunião é inspecionar o incremento entregue e analisar a necessidade de adaptação do *backlog* do produto. É realizada a verificação do que foi feito na *sprint*, quais os obstáculos encontrados e como foram resolvidos, apresenta os pontos positivos e negativos da *sprint*, e exibe os itens do *backlog* que foram concluídos nesta *sprint* [SOUZA 2013]. A reunião também permite a análise de mudança no mercado ou no uso do produto, definindo o próximo passo, além da análise do cronograma e orçamento para o próximo incremento esperado do produto [CARVALHO et al. 2009].

A reunião de revisão possui como resultado o *backlog* do produto revisado por todos os envolvidos no projeto. Além disso, fornece os insumos que servirão como base no planejamento das próximas *sprints* [CARVALHO et al. 2009].

Depois da reunião de revisão e antes da reunião de planejamento da próxima *sprint* deve ocorrer a reunião de retrospectiva do *sprint* (*sprint retrospective*) [SOUZA 2013]. Esta reunião deve ter uma duração fixa de três horas para *sprints* de um mês e deve ser ajustado de modo proporcional para *sprints* com período de tempo menor. A reunião de retrospectiva é uma oportunidade para o Time de Scrum se inspecionar e criar um plano de melhoria a ser executado na próxima *sprint* [CARVALHO et al. 2009].

A reunião de retrospectiva tem a finalidade de [SOUZA 2013]:

1. inspecionar o relacionamento entre as pessoas, os métodos de comunicação, os processos e as ferramentas em relação a última *sprint*;

2. identificar e ordenar os pontos positivos e analisar a necessidade de melhorias no processo de desenvolvimento para a próxima *sprint*;
3. criar um plano, de modo que, estas melhorias sejam implantadas na forma de trabalho do time.

Esta reunião foca na inspeção e adaptação, no entanto, as melhorias podem ser implantadas a qualquer momento durante o desenvolvimento [CARVALHO et al. 2009].

2.3.4 Artefatos do Scrum

Os artefatos do Scrum visam maximizar a transparência das informações e fornecer oportunidades de inspeção e adaptação, o que garante sucesso na entrega de um incremento de produto pronto⁵ [CARVALHO et al. 2009].

O *backlog* do produto é uma lista ordenada das características ou funcionalidades necessárias no produto, possuindo os atributos de descrição, prioridade e estimativas [SOUZA 2013]. O conteúdo do *backlog* do produto é ordenado por prioridade e é dinâmico, isto é, está em constante mudança, a fim de identificar o que o produto necessita para ser apropriado e útil para o cliente [SOUZA 2013] [CARVALHO et al. 2009].

Durante a reunião de planejamento da *sprint*, o Time de Desenvolvimento seleciona os itens do *backlog* do produto para serem desenvolvidos na próxima *sprint* [SOUZA 2013]. Estes itens são transcritos para o *backlog* da *sprint* que possibilita a visualização do trabalho realizado pelo Time de Desenvolvimento ao originar tarefas e transformar os itens do *backlog* do produto em um incremento pronto [SOUZA 2013]. Os itens devem ser decompostos de modo que seja possível desenvolvê-lo em um dia ou menos, e que seja suficiente para que mudanças nestes itens possam ser entendidas na reunião diária [CARVALHO et al. 2009].

Para cada tarefa que será realizada são usadas estimativas. A estimativa inicial de tempo e velocidade são dadas pelo *Product Owner*, o qual a realiza levando em consideração cada tarefa necessária com sua respectiva combinação de complexidade, esforço e risco. O Time de Desenvolvimento pode contestar e renegociar a estimativa [CARVALHO et al. 2009].

Algumas equipes utilizam a ideia de *planning poker*, onde cada membro do Time de Desenvolvimento utiliza um cartão numerado seguindo a sequência de Fibonacci (1, 2, 3, 5, 8, 13, ...) para pontuar a complexidade de cada item do *backlog* do produto. Todas as cartas (Figura 2.2) são apresentadas ao mesmo tempo, de modo que, nenhum membro seja influenciado pela escolha dos demais, e a partir dos resultados a equipe debate, a

⁵Um incremento é tido como pronto quando está completo. A definição de pronto para uma equipe Scrum deve possuir o mesmo significado para todos os membros do time, para que todos saibam o que precisa ser feito em um incremento [CARVALHO et al. 2009].

fim de entrar em um consenso sobre o melhor valor a ser utilizado para realizar a estimativa [PORTO e SILVA 2014].

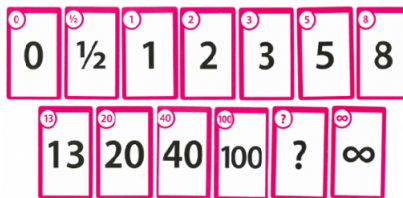


Figura 2.2: *Planning poker* [SAMI 2014]

Durante o processo de desenvolvimento deve ser possível monitorar o andamento do trabalho realizado e o que falta realizar para alcançar o objetivo da *sprint* [SOUZA 2013]. Para monitorar, controlar e prever o progresso, pode ser usado o gráfico *burndown* apresentado na Figura 2.3. Este gráfico permite a identificação da quantidade de trabalho que ainda precisa ser executado, sendo possível detectar pendências e atrasos [SOUZA 2013]. Através deste monitoramento, a equipe consegue visualizar seu progresso, determinando se o mesmo está adequado ou não para atingir a meta da *sprint* [SOUZA 2013].

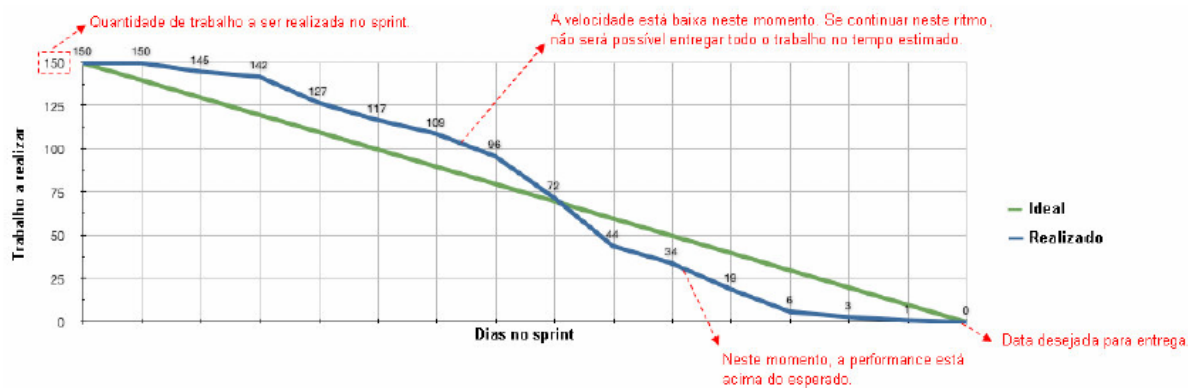


Figura 2.3: Gráfico *Burndown* [SOUZA 2013]

2.4 Processo de Implantação do Scrum

De acordo com Junior (2013), o Scrum não é previsível e não descreve com detalhes o que deve ser feito. O *framework* mantém o gerenciamento do projeto visível expondo problemas e servindo como guia para solucioná-los. Mas o percurso e métodos a serem utilizados para sua implantação são responsabilidade dos que optam por sua implantação [JUNIOR 2013].

Antes de implantar o Scrum, muitas organizações não apresentam um processo de

gerência e de desenvolvimento de *software* explícito e bem estabelecido, o que resulta da falta de uma experiência sólida em processo de desenvolvimento de produto de *software* [LIMA et al. 2012]. A inexistência de um gerenciamento bem definido faz com que as organizações apresentem os seguintes problemas [LIMA et al. 2012]:

- falta de acompanhamento dos projetos;
- falha no padrão de trabalho, onde as responsabilidades dos membros da equipe não são bem definidas;
- inexistência de um manual com políticas organizacionais;
- processos definidos de maneira independente por equipe;
- projetos atrasados;
- descentralização de informação, devido à falta de padronização ao guiar a equipe;
- prazos apertados;
- falta de comunicação e de troca de experiências entre os membros da equipe;
- trabalho após o expediente, a fim de suprir os atrasos das atividades;
- constantes mudanças nos requisitos;
- grande volume de trabalho mal dividido;
- duplicação de tarefas, por falta de detalhamento.

A presença destes problemas alerta a necessidade de adoção de princípios e conceitos de agilidade [LIMA et al. 2012]. O Scrum se destaca pelos seus princípios básicos de objetividade, papéis bem definidos e facilidade de aprendizado.

Ao optar pelo Scrum, é aconselhável que sua implantação seja realizada de maneira gradual em apenas um projeto da organização, a fim de minimizar o impacto de possíveis riscos e, em caso de sucesso, pode ser replicado para os demais. A equipe deve ser submetida a um treinamento, a fim de introduzir o Scrum e todas as suas práticas. Este treinamento deve abordar: os valores e pilares do *framework*, a forma de se relacionar com o cliente, a divisão em papéis, a criação dos artefatos, o uso de estimativas, os eventos e todas as demais práticas e características próprias do Scrum [JUNIOR 2013, CARVALHO et al. 2009].

Após toda equipe possuir pleno conhecimento sobre o Scrum, deve ser acordado como implementá-lo. Deve ser realizada a definição dos papéis, o que demanda verificar as características desejáveis para cada pessoa. Por exemplo, é preferível que o *Scrum Master* possua prévio conhecimento sobre o *framework* Scrum e experiência em gerenciamento de

projetos, enquanto o *Product Owner* possua um alto grau de conhecimento das regras de negócio [CARVALHO et al. 2009]. Além de definir a equipe, é necessário definir como serão as *sprints* e os demais eventos do Scrum, esclarecendo os conceitos do Scrum que serão implantados e como serão implantados [JUNIOR 2013, CARVALHO et al. 2009].

As práticas do método são introduzidas aos poucos com o objetivo de adaptar a equipe; nem sempre é possível introduzir todos os conceitos a partir da primeira iteração, mas no decorrer do desenvolvimento deve ser feito o possível para implantar totalmente as práticas [JUNIOR 2013, CARVALHO et al. 2009].

Durante o processo de implantação do Scrum a organização pode enfrentar várias dificuldades; algumas delas são apresentadas na Subseção 2.4.1. A Subseção 2.4.2 tem como objetivo mostrar alguns dos principais benefícios alcançados pela organização ao realizar o processo de implantação do *framework* corretamente.

2.4.1 Principais dificuldades encontradas

O grau de dificuldade na implantação varia muito de uma organização para outra. Desta forma, antes de realizar a implantação do Scrum é necessário realizar um estudo e fazer um levantamento sobre as possíveis dificuldades que serão encontradas durante a implantação, a fim de diminuí-las. A implantação de um novo método exige desprender de valores, premissas e comportamentos antigos antes que novos conceitos sejam introduzidos [JUNIOR 2013]. Além disso, a mudança cultural da organização e da gestão de projetos deve ser aceita por todos os envolvidos, pois tal mudança é considerada um dos maiores bloqueios ao se implantar um novo método [PORTO e SILVA 2014]. As dificuldades provavelmente serão muitas ao se implantar o Scrum, portanto a melhoria e adaptação devem ser constantes [FREIRE 2009].

Geralmente, as organizações enfrentam as seguintes dificuldades [JUNIOR 2013]:

- gerentes tradicionais apresentam medo de mudar, perder o controle e autoridade;
- programadores possuem medo de quebrar a rotina e de novas responsabilidades;
- testadores ficam com medo de perder o emprego;
- treinar a equipe quanto aos conceitos (papéis, eventos, etc) do Scrum;
- definir as responsabilidades do *ProductOwner* e do *Scrum Master*;
- priorizar e estimar os itens do *backlog* do produto;
- integrar a equipe;
- falta de pessoas com habilidades certas;
- pouco tempo disponível para realização da transição;

- gerenciar e integrar os vários artefatos;
- baixo conhecimento técnico por parte da equipe;
- fazer com que a equipe permaneça comprometida em continuar aplicando os conceitos do Scrum;
- gerenciar as reuniões, pois muitas vezes o tempo estimado é ultrapassado sem que o objetivo da reunião seja alcançado.

Segundo a experiência de Vasconcelos e Pereira (2012), levando em consideração conceitos específicos do Scrum, foram identificadas as dificuldades apresentadas na Tabela 2.4.

Tabela 2.4: Dificuldades na implantação das práticas gerenciais do Scrum [VASCONCELOS e PEREIRA 2012]

Item	Dificuldades
<i>Backlog</i> do produto	Dificuldade para elaborar a primeira versão que geralmente é muito extensa.
Reunião diária	Falhas de periodicidade devido a desencontros de horário dos membros da equipe.
<i>Sprint</i>	Devido à inexperiência da equipe, algumas <i>sprints</i> podem ter duração muito longa.
Planejamento da <i>sprint</i>	Reuniões pouco produtivas devido à inexperiência da equipe.
<i>Backlog</i> da <i>sprint</i>	Seu superdimensionamento pode causar atrasos nas <i>sprints</i> .
Revisão da <i>sprint</i>	Tende a não ser executado pela equipe quando o time está ansioso para o planejamento da próxima <i>sprint</i> .
Gráfico <i>burndown</i>	Tende a ser abandonado, pois, à primeira vista, parece informação redundante.
<i>Product owner</i>	É difícil encontrar uma pessoa que entenda realmente o negócio do cliente a ponto de fazer bem este papel.
<i>Scrum master</i>	Muitas vezes o time o vê como um gerente do projeto, deixando para ele atribuições que deveriam ser do próprio time.

2.4.2 Principais benefícios alcançados

Os benefícios que uma organização pode alcançar com a implantação do Scrum dependem do seu estado anterior e se a implementação do método foi realizada da maneira correta, mas geralmente os seguintes benefícios são alcançados [VASCONCELOS e PEREIRA 2012]:

- melhora na comunicação e colaboração entre os envolvidos;
- maior controle, histórico e planejamento;
- equipe autogerenciável;
- troca de conhecimento entre os membros da equipe;
- aumento da motivação;
- independência e transparência na tomada de decisões;
- otimização e homogeneidade do tempo de desenvolvimento;
- entrega de um produto de qualidade em tempo hábil;
- aumento da satisfação do cliente.

De acordo a experiência relatada em [VASCONCELOS e PEREIRA 2012], ao levar em consideração conceitos específicos do Scrum, foram identificadas as seguintes vantagens e facilidades, apresentadas na Tabela 2.5.

Tabela 2.5: Vantagens na implantação das práticas gerenciais do Scrum [VASCONCELOS e PEREIRA 2012]

Item	Vantagem/Facilidades
<i>Backlog</i> do produto	Sua adoção melhorou o planejamento do time.
Reunião diária	Sua utilização aumentou muito o controle do projeto. Os riscos foram minimizados.
<i>Sprint</i>	A divisão do projeto em <i>sprints</i> aumentou a motivação do time.
Planejamento da <i>sprint</i>	Sua realização melhorou o planejamento da <i>sprint</i> .
<i>Backlog</i> da <i>sprint</i>	Elaborá-lo corretamente é importante para o sucesso da <i>sprint</i> .

Tabela 2.5 – Vantagens (continuação)

Item	Vantagem/Facilidades
Gráfico <i>burndown</i>	Utilizado pelas equipes Scrum para representar o progresso do trabalho em desenvolvimento na <i>sprint</i> , permite visualizar o trabalho realizado e verificar se o mesmo está de acordo com o planejado.
<i>Product owner</i>	Facilitador para o time, que passa a ter um representante do cliente dentro da própria organização.
<i>Scrum master</i>	É imprescindível para tirar dúvidas quanto ao processo de desenvolvimento e liderar as cerimônias do Scrum.
Revisão da <i>sprint</i>	Importante reunião para o recolhimento das lições aprendidas.

2.5 Considerações finais

Os métodos ágeis surgiram como uma opção aos métodos tradicionais, devido à necessidade de desenvolver *softwares* de qualidade que possam ser produzidos mais rapidamente, e de processos que comportem alterações nos requisitos sem causar riscos ao projeto. Os métodos ágeis possuem uma visão diferente dos métodos tradicionais, priorizando as pessoas, a interação com o cliente e produzindo documentação e artefatos que sejam realmente necessários. Vale ressaltar que a escolha por métodos tradicionais ou métodos ágeis depende do contexto em que se encontra a organização e o projeto; sendo assim, os métodos ágeis não surgiram com o intuito de substituir os tradicionais, mas são uma opção melhor em determinados casos.

O Scrum, objeto de estudo deste trabalho, é um *framework* voltado para o gerenciamento de projetos. Seus conceitos são simples e fáceis de se compreender, este *framework* permite a visibilidade dos problemas organizacionais, possibilitando a solução de maneira mais rápida e eficiente.

O *framework* Scrum fornece vários benefícios para as organizações que optam por sua implantação e se adaptam às suas práticas e conceitos. Porém, há a necessidade de analisar o perfil da organização a fim de verificar a possibilidade de implantação de um novo método. O próximo capítulo apresenta alguns dos principais elementos que compõem o perfil de uma organização.

CAPÍTULO 3

ELEMENTOS ESTRUTURAIS DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Este capítulo apresenta os principais elementos que compõem a estrutura geral do processo de desenvolvimento de *software* utilizados neste trabalho para compor o perfil de uma organização. Estes elementos foram propostos por [\[BARBOSA e GREGÓRIO 2014\]](#) e são decompostos em três visões: Ambiente Organizacional, Equipe de Desenvolvimento e Produto de *software*. Além disso, neste capítulo apresenta-se a perspectiva dos autores Barbosa e Gregório (2014) quanto a um perfil organizacional - embasado nos elementos estruturais - adequado ao Scrum.

3.1 Introdução

O processo de desenvolvimento de *software* possui vários fatores que influenciam e são cruciais para o resultado do projeto, determinando se o mesmo será bem-sucedido ou não. Nem sempre esses fatores são internos e fazem parte de uma etapa do desenvolvimento do *software*; às vezes, esses fatores são externos, porém afetando de igual forma ou até mais o processo de desenvolvimento [\[BARBOSA e GREGÓRIO 2014\]](#). Estes elementos compõem a estrutura geral do processo de desenvolvimento de *software*, funcionando como uma base comum sobre a qual projetos diferentes são planejados e independe do método de desenvolvimento escolhido [\[BARBOSA e GREGÓRIO 2014\]](#). Esses elementos devem ser analisados adequadamente a fim de viabilizar a implantação do melhor método de desenvolvimento de *software*, de modo que cada elemento deve ser verificado de acordo com situações específicas.

Segundo Barbosa e Gregório (2014), esses elementos estão presentes em três visões diferentes que são: Ambiente Organizacional (seção 3.2), Equipe de Desenvolvimento (seção 3.3) e Produto de *software* (seção 3.4). Os autores também apresentam, considerando

estas visões, um perfil organizacional (seção 3.5) que, segundo eles, mais se adequa ao Scrum.

3.2 Ambiente Organizacional

O ambiente organizacional está relacionado a cada organização especificamente, sendo constituído pelas características da mesma e tudo o que preza e emprega a fim de alcançar seus objetivos [BARBOSA e GREGÓRIO 2014]. Cada organização possui o conjunto de itens: tamanho, nível de investimento em tecnologia, nível de maturidade, interação com o cliente e estrutura organizacional. Estes elementos são apresentados nas subseções a seguir.

3.2.1 Tamanho

O tamanho da organização possibilita a percepção do grau de investimento necessário, a formalidade, os gastos e também o porte que os projetos devem ter para que possam manter a organização [BARBOSA e GREGÓRIO 2014]. O tamanho da organização pode ser determinado utilizando vários critérios, tais como: número de empregados, número de clientes, faturamento, setor de atividade, lucro, entre outros [MARTINS 2014].

Segundo BNDES (2011), há uma classificação aplicável a todos os setores que leva em consideração a receita bruta mensal da empresa, tal classificação pode ser vista na Tabela 3.1. Já na Tabela 3.2 é possível observar a classificação utilizada pelo Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE), que determina o tamanho da organização usando como critérios o número de empregados e o setor de atividade econômica exercida pela organização [SEBRAE 2013]. Neste trabalho será considerada a classificação por setor de comércio e prestação de serviço, pois este inclui a produção de *software*.

Tabela 3.1: Classificação da organização de acordo com a receita operacional bruta anual [BNDES 2011]

Classificação	Receita Operacional Bruta Anual
Microempresa	Menor ou igual a R\$ 2, 4 milhões
Pequena empresa	Maior que R\$ 2, 4 milhões e menor ou igual a R\$ 16 milhões
Média empresa	Maior que R\$ 16 milhões e menor ou igual a R\$ 90 milhões
Média-grande empresa	Maior que R\$ 90 milhões e menor ou igual a R\$ 300 milhões
Grande empresa	Maior que R\$ 300 milhões

Tabela 3.2: Classificação do porte da empresa segundo o SEBRAE [SEBRAE 2013].

Porte da empresa	Setor	
	Indústria ¹	Comércio ² e Prestação de Serviço ³
Microempresa	Até 19 pessoas	Até 9 pessoas
Pequena empresa	De 20 a 99 pessoas	De 10 a 49 pessoas
Média empresa	De 100 a 499 pessoas	De 50 a 99 pessoas
Grande empresa	De 500 ou mais	100 ou mais

3.2.2 Nível de investimento em tecnologia

Segundo Sanchez e Albertin (2009), a combinação adequada da tecnologia com outros recursos internos organizacionais pode propiciar o sucesso e particularidades difíceis de serem copiadas por concorrentes.

Cada organização define um orçamento para seus projetos internos, tal orçamento deve ser conhecido a fim de definir como os processos serão configurados e suas necessidades serão apoiadas [BARBOSA e GREGÓRIO 2014].

De acordo com Barbosa e Gregório (2014), levando em consideração o faturamento bruto em TI, os níveis de investimento são deliberados da seguinte forma: investimento inferior a 1% é classificado como baixo, entre 1% e 2% é classificado como mediano e acima de 2% é classificado como alto.

3.2.3 Nível de maturidade

Grau de Maturidade Da organização	Níveis PMMM		
	Nível 1 Processo Inicial	Nível 2 Processos Estruturados e Padrões	
Baixa Maturidade	Práticas ou normas não estabelecidas. A produção da documentação do projeto e a coleta das métricas são realizadas informalmente.	Métricas básicas e documentação do projeto estão presentes, mas nenhum padrão organizacional é definido	
Alta Maturidade	Nível 3 Padrões Organizacionais	Nível 4 Processos Gerenciados	Nível 5 Processos de Otimização
	Todos os projetos utilizam padrões formais organizacionalmente institucionalizados.	As métricas são usadas para gerenciar os projetos, e integradas em outros sistemas corporativos para maximizar o desempenho global da organização.	As lições aprendidas são estudadas rotineiramente para melhorar os processos PM.

Figura 3.1: Grau de maturidade da organização de acordo com os níveis de maturidade do modelo PMMM [SILVEIRA et al. 2013, VERGOPIA 2008]

Um gerenciamento de projeto eficaz é um fator determinante para a subsistência da organização. Sendo assim, para obter um melhor desempenho, as organizações necessitam de processos de gerenciamento de projetos organizados e maduros [SILVEIRA et al. 2013]. Segundo o estudo realizado por [SILVEIRA et al. 2013], é possível classificar os níveis de maturidade em Baixa Maturidade e Alta Maturidade. O modelo de maturidade para gestão de projeto utilizado na classificação foi o PMMM (*Project Management Maturity*

Model). Este modelo avalia a organização utilizando as nove áreas de conhecimento especificadas no PMBOK: escopo, integração, tempo, custo, qualidade, recursos humanos, comunicações, risco e aquisição; e segue o padrão do CMM (Modelo de Maturidade da Competência) [VERGOPIA 2008]. O PMMM possui cinco níveis de maturidade que são apresentados na Figura 3.1 [SILVEIRA et al. 2013, VERGOPIA 2008].

3.2.4 Interação com o cliente

A interação com o cliente durante o desenvolvimento e a entrega do *software* é de extrema importância para o sucesso do projeto, sendo essencial para definir o modo que o mesmo será conduzido [BARBOSA e GREGÓRIO 2014]. Estabelecer as necessidades do produto não é uma tarefa trivial, e nem sempre o cliente consegue esclarecer suas necessidades no levantamento de requisitos. Sendo assim, sua presença no decorrer do projeto é de grande relevância [BARBOSA e GREGÓRIO 2014]. Segundo Barbosa e Gregório (2014) há uma classificação que advém do grau de disponibilidade e envolvimento do cliente com o projeto, e que impõem normas específicas no processo de desenvolvimento, conforme definido a seguir:

- Baixa – O envolvimento com o cliente ocorre somente no início do projeto (ao definir como será o projeto no levantamento dos requisitos) e no fim do projeto (ao entregar e aprovar o produto).
- Média – Além do envolvimento com o cliente no início e no fim do projeto, o cliente possui alguma participação durante o desenvolvimento do produto, quando esclarece alguma dúvida quanto ao produto ou solicita novos requisitos.
- Alta – O cliente possui participação durante todo o processo de desenvolvimento, revisando os requisitos, realizando avaliações de entregas parciais do produto e auxiliando na minimização de riscos relacionados ao levantamento de requisitos e ao projeto como um todo.

3.2.5 Estrutura organizacional

A estrutura organizacional, assim como outros fatores, também influencia no projeto, o que pode ser visto na Figura 3.2. De acordo com [PMBOK 2013], a estrutura organizacional pode ser definida da seguinte forma:

- Organização Funcional Clássica: Cada departamento realiza seu trabalho no projeto, independente dos demais departamentos. Há uma hierarquia, de modo que, cada funcionário possui um superior definido e os funcionários são agrupados por especialidades, que podem ser subdivididas em áreas funcionais distintas.

- Organização Matricial: Esse tipo de organização combina atributos funcionais e projetizados. De acordo com o grau de poder e influência do gerente de funcional e do gerente de projeto, é possível classificar uma organização matricial em: fraca, balanceada e forte.
 - Fraca: Possui mais características da organização funcional e o gerente de projeto posiciona-se mais como um facilitador.
 - Balanceada: O gerente de projetos tem sua presença reconhecida como necessária, porém o mesmo não possui autoridade total sobre o projeto ou sobre o seu financiamento.
 - Forte: apresenta muitas características das organizações projetizadas e o gerente do projeto tem uma autoridade aceitável.
- Organização Projetizada: Geralmente, as organizações projetizadas possuem departamentos que se dirigem diretamente ao gerente de projetos ou que fornecem serviços de suporte aos vários projetos da organização.

Estrutura da organização Características do projeto	Funcional	Matricial			Projetizada
		Matriz fraca	Matriz por matricial	Matriz forte	
Autoridade do gerente de projetos	Pouca ou nenhuma	Baixa	Baixa a moderada	Moderada a alta	Alta a quase total
Disponibilidade de recursos	Pouca ou nenhuma	Baixa	Baixa a moderada	Moderada a alta	Alta a quase total
Quem gerencia o orçamento do projeto	Gerente funcional	Gerente funcional	Misto	Gerente do projeto	Gerente do projeto
Papel do gerente de projetos	Tempo parcial	Tempo parcial	Tempo integral	Tempo integral	Tempo integral
Equipe administrativa de gerenciamento de projetos	Tempo parcial	Tempo parcial	Tempo parcial	Tempo integral	Tempo integral

Figura 3.2: Característica do projeto de acordo com a estrutura organizacional [PMBOK 2013]

3.2.6 Porte de projeto

O porte do projeto pode variar de acordo com certos fatores como: a quantidade de profissionais necessária para desempenhar as atividades; as ferramentas e técnicas necessárias; o grau de dificuldade para implementar o produto solicitado; a duração do projeto, entre outros fatores. Mas de modo geral pode-se simplificar e associar o porte dos projetos

com base nos valores de seu custo, pois um projeto que possui uma equipe pequena, com um grau de dificuldade baixo, cuja duração não seja extensa provavelmente terá um custo menor que um projeto que necessite de muitos profissionais, ferramentas e técnicas para alcançar o seu objetivo, além do produto ser de difícil implementação com alto risco em caso de falha.

Em [BARBOSA e GREGÓRIO 2014] é apresentada a seguinte classificação de porte de projeto baseada nos valores de custo: Pequeno: custo inferior a R\$ 200.000,00; Médio: custo entre R\$ 200.001,00 e R\$ 1.000.000,00; Grande: custo acima de R\$ 1.000.001,00.

3.3 Equipe de desenvolvimento

Segundo Miranda (2011), os recursos humanos são os principais ativos de uma organização. Sendo assim, esta visão envolve os fatores relacionados aos membros que compõem a equipe de desenvolvimento e que afetam o processo de desenvolvimento de *software*.

3.3.1 Nível de especialização

O nível de especialização da equipe de desenvolvimento de *software* é obtido através da verificação se cada membro é especializado em apenas um papel específico dentre os vários papéis disponíveis [BARBOSA e GREGÓRIO 2014]; ou se cada membro possui uma visão global do processo e domínio de trabalho, podendo assim ocupar mais de um papel durante o processo de desenvolvimento [BARBOSA e GREGÓRIO 2014] [MIRANDA 2011]. O nível de especialização da equipe pode ser:

- Específico: Os membros possuem seus papéis definidos, de modo que cada membro só está apto a atuar em um papel específico;
- Generalista: Os membros possuem uma ou mais especialidades, tendo um conhecimento geral do domínio de trabalho, de modo que estão aptos a atuar em mais de um papel no processo.

3.3.2 Tamanho da equipe

Uma equipe pode ser composta de uma quantidade variada de membros, podendo ser classificada em: Pequena, Média ou Grande. De acordo com Barbosa e Gregório (2014) a quantidade é definida da seguinte forma: Pequena – até 4 membros; Média – de 5 a 9 membros; e Grande – acima de 10 membros.

3.3.3 Grau de autonomia da equipe

Desde 1950 tem sido objeto de estudo o autogerenciamento de equipes, com o intuito de aumentar a eficácia do trabalho em equipe, o que pode resultar em redução de custos, melhoria na produtividade e qualidade e também no aumento da satisfação por parte dos integrantes da equipe [MIRANDA 2011].

O grau de autonomia determina o quanto a equipe está capacitada para tomar decisões. Este grau de autonomia está diretamente relacionado com as regras da organização, se sua estrutura possibilita independência por parte da equipe. Além disso, leva em consideração se a equipe realmente está apta a tomar decisões sem que o projeto seja comprometido [BARBOSA e GREGÓRIO 2014]. A fim de analisar a autonomia da equipe tem-se a seguinte classificação [BARBOSA e GREGÓRIO 2014]:

- Baixa – A equipe é totalmente dependente do gestor do projeto, não tomando nenhuma decisão;
- Moderada – Na ausência do gestor, a equipe pode tomar decisões que não sejam consideradas complexas ou críticas;
- Alta – A equipe tem permissão para tomar qualquer tipo de decisão, mesmo na ausência do gestor.

3.3.4 Localização da equipe

Atualmente, seja por indisponibilidade da equipe ou por decisão de negócio [MIRANDA 2011] tem se tornado comum as equipes estarem distribuídas com grandes distâncias geográficas entre os membros. Esta distribuição tem sido possível devido ao grande avanço tecnológico, porém pode causar divergências na comunicação e interação entre os membros da equipe [BARBOSA e GREGÓRIO 2014].

Segundo Barbosa e Gregório (2014), a localização pode ser classificada em: Centralizada – A equipe localiza-se toda no mesmo ambiente; ou Descentralizada – A equipe está distribuída por locais diferentes e distantes um do outro.

3.3.5 Experiência da equipe

Segundo Miranda (2011), a habilidade e a experiência são fatores cruciais para um resultado satisfatório de um projeto de desenvolvimento de *software*. O nível de experiência da equipe como um todo é determinado pela experiência de cada membro em particular, que é dada pelo tempo de prática com as técnicas de desenvolvimento utilizadas pela organização [BARBOSA e GREGÓRIO 2014]. O tempo de prática dos membros da equipe pode ser definido como [BARBOSA e GREGÓRIO 2014]: Júnior – até 4 anos; Pleno – de 5 a 8 anos; e Sênior – de 9 anos para cima.

Ao analisar a experiência de cada membro da equipe, é possível definir o nível de experiência da equipe de modo geral, determinando-o como [BARBOSA e GREGÓRIO 2014]: Baixo – A equipe, em média, é composta por membros com tempo de prática definido como Júnior; Médio – A equipe, em média, é composta por membros com tempo de prática definido como Pleno; e Grande – A equipe, em média, é composta por membros com tempo de prática definido como Sênior.

3.3.6 Comunicação entre os membros da equipe

A equipe deve se comunicar e compartilhar as experiências vivenciadas no dia-a-dia. Mesmo que o processo de desenvolvimento seja o ideal e a equipe composta por profissionais experientes [UTIDA 2009], uma boa comunicação e o entrosamento da equipe é imprescindível para o bom funcionamento do projeto, auxiliando na resolução de problemas e redução de pendências [LEIDEMER 2014, UTIDA 2009]. A comunicação da equipe pode ser classificada em [LEIDEMER 2014, UTIDA 2009]:

- Baixa: A equipe se comunica pouco, todas as informações são passadas através de documentação.
- Média: A equipe preza a comunicação, porém por consequências de fatores como distância, falta de tempo e horários inconsistentes, a comunicação não ocorre de maneira síncrona e, na maioria das vezes, não é realizada pessoalmente.
- Alta: A equipe se comunica constantemente e pessoalmente, colaborando entre si e compartilhando experiências.

3.4 Produto de software

Cada produto exige um processo de desenvolvimento de *software* específico, de modo que o produto seja finalizado e entregue com sucesso, tendo todas as necessidades e requisitos elicitados atendidos. Os elementos estruturais relacionados ao produto de *software* são apresentados nas subseções a seguir.

3.4.1 Nível de documentação exigida

A documentação é um elemento muito importante para o produto, pois permite visualizar o processo de construção pelo qual o produto passou [BARBOSA e GREGÓRIO 2014]. Segundo Ambler (2016), há dois principais motivos para se documentar: a comunicação e manutenção. Em relação à comunicação, a documentação serve como um suporte às reuniões; quanto à manutenção, facilita que a mesma ocorra permitindo a compreensão do funcionamento do *software*, e diminuindo o tempo para entendimento do sistema.

Cada produto exige um nível específico de documentação o que é impactado diretamente pelo perfil do cliente [BARBOSA e GREGÓRIO 2014]. Os níveis de documentação são classificados como [BARBOSA e GREGÓRIO 2014]:

- Baixo - O processo de desenvolvimento do produto gera pouca ou nenhuma documentação;
- Moderado - O processo de desenvolvimento do produto gera uma documentação parcial, porém sem exigência legal;
- Alto - O processo de desenvolvimento do produto é 90% documentado.

3.4.2 Mudança de escopo

Atualmente, o ambiente de negócios tem passado por constantes mudanças o que torna muito complicado obter o conhecimento de todos os requisitos estáveis e indispensáveis ao produto; sendo assim, é praticamente inevitável realizar várias alterações no escopo durante o projeto [SOMMERVILLE 2003, BARBOSA e GREGÓRIO 2014]. Mudanças são custosas e trazem riscos ao projeto caso o mesmo não esteja apto para a ocorrência delas; muitos contratos exigem que o escopo seja especificado antes do início do projeto ou que qualquer mudança necessária só seja realizada após a conclusão da construção do produto [BARBOSA e GREGÓRIO 2014].

A mudança de escopo pode ocorrer em três níveis [BARBOSA e GREGÓRIO 2014]:

- Baixo - Escopo deve ser especificado antes do início do projeto e qualquer mudança necessária só deve ser realizada após a conclusão da construção do produto.
- Médio - Escopo deve ser especificado antes do início do projeto, mas mudanças são aceitas ao entregar um incremento desde que esta mudança não traga riscos ao projeto.
- Alto - Escopo deve ser especificado antes do início do projeto, mas é permitido que o cliente/usuário solicite mudanças durante o projeto.

3.5 Aplicação no Scrum

Esta seção apresenta quais elementos estruturais que compõem o processo de desenvolvimento de *software* melhor se adequam ao Scrum de acordo com o estudo realizado por [BARBOSA e GREGÓRIO 2014].

Vale ressaltar que não é correto afirmar que organizações que não estejam enquadradas na análise apresentada na Tabela 3.3 não podem implantar o Scrum, mas de acordo

com Barbosa e Gregório (2014) organizações que possuem a opção apresentada na análise terão mais facilidade para implantação e responderão melhor ao processo de adaptação.

Barbosa e Gregório (2014) não especificam como definiram os elementos mais adequados ao Scrum; sendo assim, o presente trabalho apresenta na subseção 4.3.3 uma comparação entre os resultados obtidos a partir da pesquisa realizada e o perfil organizacional apresentado pelos autores, de modo a avaliar a análise realizada por eles.

Tabela 3.3: Perfil organizacional adequado ao Scrum de acordo com [BARBOSA e GREGÓRIO 2014]

Elemento	Classificação	Análise
Tamanho da organização	Média	O Scrum é um <i>framework</i> que exige uma certa estruturação por parte da organização, de modo que, a sua equipe seja capaz de suportar as práticas Scrum. Uma organização média é mais adequada.
Nível de investimento de TI	Mediano	O Scrum é um <i>framework</i> inovador que exige profissionais qualificados, sendo assim é necessário um nível médio de investimento em TI para que seja possível implantar as práticas da metodologia
Nível de maturidade	Mediano	A organização deve possuir o entendimento de que saber como conduzir e organizar um projeto é essencial para a qualidade do produto a ser desenvolvido, para isso um nível de maturidade médio na gestão de projetos é suficiente para a implantação do Scrum.
Interação com o cliente	Alta	O Scrum preza pela constante interação com o cliente, a fim de garantir que o produto esteja dentro dos parâmetros desejados por ele e acomodar melhor as mudanças que possam ocorrer no projeto, garantindo também a confiança para ambas as partes.
Tipo de organização	Irrelevante	O Scrum pode ser implantado em qualquer tipo de organização.

Tabela 3.3 – Perfil organizacional adequado ao Scrum (continuação)

Elemento	Classificação	Análise
Porte de projeto	Médio	O Scrum é adequado para projetos de médio porte, para que seja possível se adaptar melhor as mudanças e fornecer a documentação em um nível satisfatório.
Tamanho da equipe	Médio	O tamanho da equipe deve ser médio, pois uma equipe pequena pode apresentar problemas relacionados a interação e a falta de conhecimento necessário para realizar as tarefas da <i>sprint</i> , e uma equipe grande é muito complexa para gerenciar.
Nível de especialização	Generalista	A equipe deve ser multifuncional para que seja capaz de gerar um incremento funcional sem auxílio externo.
Grau de autonomia	Alto	A equipe deve ter um nível alto de autonomia para que assim seja capaz de se auto-organizar.
Localização da equipe	Centralizada	O Scrum prioriza a comunicação entre os membros da equipe, uma maior interação com o cliente e reuniões para avaliar o andamento do projeto, sendo assim esses fatores podem ser obtidos mais facilmente se a equipe for centralizada
Experiência da equipe	Médio	A equipe deve possuir capacidade para produzir incrementos funcionais do produto sem ajuda externa ao projeto, para isso é necessário ao menos possuir um nível médio de experiência.
Nível de documentação exigida	Mediano	Para o Scrum a quantidade de documentação deve ser a mínima necessária para o incremento do produto.
Mudança de escopo	Alta	Por ser iterativo o Scrum possibilita alterações de escopo com baixo custo e risco para o projeto.

3.6 Considerações finais

No contexto em que vivemos atualmente, são desenvolvidos diferentes tipos de *software* para as mais diversas finalidades. Porém, independentemente do tipo de *software*, este necessita ser planejado e passar por um processo de desenvolvimento, e espera-se um produto eficiente e de qualidade.

O processo de desenvolvimento de *software* pode ser influenciado por diversos elementos, sejam internos ou externos. Estes elementos estruturais formam uma base sobre a qual diferentes projetos de desenvolvimento de *software* podem ser projetados. Neste capítulo, foram apresentados apenas os principais elementos estruturais.

Considerando estes elementos estruturais, Barbosa e Gregório (2014) apresentaram quais elementos são mais adequados ao Scrum. O capítulo 4 apresenta o estudo realizado neste trabalho a fim de identificar, se possível, um perfil organizacional mais adequado ao *framework* a partir de estudos de casos reais de implantação do Scrum em organizações de desenvolvimento de *software*.

CAPÍTULO 4

ANÁLISE TEÓRICA: SCRUM NAS ORGANIZAÇÕES

Este capítulo apresenta o estudo realizado a partir de um mapeamento sistemático, com o intuito de identificar se há um perfil organizacional mais adequado à adoção do Scrum, visando auxiliar organizações que pretendem adotar este *framework*.

4.1 Introdução

A atual necessidade das organizações por *softwares* de qualidade desenvolvidos rapidamente aumentou a procura por métodos ágeis. No âmbito da gerência de projetos, o Scrum tem sido escolhido por muitas organizações, pois é um método adaptativo, responde rapidamente às constantes mudanças que podem ocorrer no projeto, possui papéis bem definidos e possui facilidade de aprendizado [JUNIOR 2013].

Uma falha de muitas organizações é que elas testam vários métodos/técnicas antes de pesquisar a que melhor se aplica ao seu contexto, pois não analisam corretamente suas necessidades e nem mesmo identificam se é o momento ideal para uma mudança cultural organizacional [CARVALHO et al. 2009]. Sendo assim, antes de implementar o Scrum ou qualquer outro método ou técnica, é necessária uma análise detalhada da organização, a fim de verificar se o método escolhido é realmente adequado ao seu perfil e contexto; é necessário também expor problemas prioritários e verificar o modo que a organização conduz seus projetos, avaliando se há uma real necessidade de melhorar seu processo. Além disso, é necessário um estudo prévio do impacto da implantação da mudança na organização e se está preparada estruturalmente [CARVALHO et al. 2009].

Com o aumento da procura pelo *framework* Scrum uma questão que surge é se ele pode ser adotado por qualquer tipo de organização e em qualquer tipo de projeto, sendo garantido o sucesso. A fim de responder esta questão, este trabalho se propõe a analisar - tendo como base relatos de experiência disponíveis na literatura de organizações que já implantaram o Scrum - e identificar se há um possível perfil organizacional mais adequado

à implantação do Scrum.

Desta forma, este trabalho busca fornecer uma visão às organizações que querem implantar o *framework* Scrum, permitindo que optem ou não pela sua escolha, baseando-se em seu perfil organizacional, possibilitando evitar os riscos e até mesmo o fracasso do projeto. Pois a incorporação do Scrum depende da forma como a organização está estruturada, o que varia muito de uma organização para outra; cada organização possui suas particularidades, quanto à forma de organizar a equipe e manter a comunicação entre ela, dividir as tarefas, se relacionar com o cliente, documentar o projeto, lidar com mudanças nos requisitos e forma de entrega do produto.

Este estudo foi elaborado a partir dos elementos estruturais apresentados no Capítulo 3 e a partir da execução de um mapeamento sistemático, a ser apresentado na seção 4.2. A seção 4.3 traz os resultados deste estudo, e na seção 4.4 são descritas as considerações finais.

4.2 Mapeamento sistemático

O mapeamento sistemático é um método de busca na literatura que visa obter e analisar dados referentes a um assunto. É constituído por um conjunto de etapas que determinam a finalidade e as fontes da pesquisa, e os critérios de seleção destas fontes [KITCHENHAM e CHARTERS 2007].

A fim de identificar o perfil de organizações que optaram pela implantação do Scrum, realizou-se um mapeamento sistemático visando organizar de maneira mais adequada o processo de busca por estudos de casos reais e obter resultados consistentes. A partir das publicações identificadas buscou-se caracterizar o perfil organizacional, extraindo os elementos estruturais (apresentados no Capítulo 3) apresentados nos relatos.

Para condução do estudo, seguiu-se parcialmente o processo definido em [KITCHENHAM e CHARTERS 2007], composto pelas seguintes práticas:

- Definir escopo e análise preliminar;
- Determinar protocolo;
- Executar pesquisa;
- Analisar resultados.

As subseções seguintes descrevem como foram conduzidas cada uma destas etapas, exceto a etapa “Analisar resultados” que está descrita na seção 4.3.

4.2.1 Definição de escopo e análise preliminar

Esta etapa consistiu em realizar uma pesquisa informal e manual, tendo como fontes de busca: Revista FSMA¹, anais do *Workshop* Anual do MPS (WAMPS)², BDBComp³ e o Google Acadêmico⁴.

O intuito desta pesquisa informal foi assimilar melhor o assunto e o escopo do trabalho, além de obter uma base para estruturar e definir o processo de busca formal. O foco desta pesquisa foi encontrar publicações com relatos de experiências reais da utilização do *framework* Scrum em processos de desenvolvimento de *software*.

A Tabela 4.1 apresenta o resultado quantitativo por fonte das publicações obtidas através da pesquisa informal realizada, contabilizando apenas as publicações consideradas pertinentes ao escopo deste trabalho.

Tabela 4.1: Resultado quantitativo das publicações obtidas via pesquisa informal

Fonte	Quantidade de publicações	Referência
Revista FSMA	1	[LIMA et al. 2012]
WAMPS	0	-
BDBComp	2	[SOARES et al. 2007] [BORGES 2011]
Google Acadêmico	11	[BENEFIELD 2008] [ELWER 2008] [CARVALHO et al. 2009] [VARASCHIM 2009] [MELO e FERREIRA 2010] [UHLE 2010] [CARVALLHO et al. 2011] [JUNIOR 2013] [LEIDEMER 2014] [LEITE e LUCRÉDIO 2014] [MONTEIRO 2015]

Nos anais do WAMPS, foram encontradas publicações relacionadas ao Scrum, porém nenhuma estava dentro do escopo deste trabalho, ou seja, não descreviam relatos de experiência de implantação do *framework* Scrum.

4.2.2 Determinação do protocolo

O protocolo visa orientar o processo de pesquisa e análise. Este protocolo foi constituído das seguintes etapas: definir finalidade da pesquisa; determinar critérios de seleção

¹<http://www.fsma.edu.br/si/sistemas.html>

²<http://www.softex.br/mpsbr/wamps/>

³<http://www.lbd.dcc.ufmg.br/bdbcomp/>

⁴<https://scholar.google.com.br/>

das fontes de publicação; estabelecer *string* de busca; definir um procedimento de seleção das publicações relevantes a este trabalho; e definir procedimento de obtenção de informações.

Como informado anteriormente, a finalidade da pesquisa é, identificar publicações que apresentem casos reais e de sucesso em projetos de desenvolvimento de *software* e, a partir destes, extrair as informações pertinentes ao objetivo desta monografia, sendo possível: identificar os elementos organizacionais descritos no capítulo 3; e verificar se as organizações que optaram pela implantação do Scrum possuem as mesmas características, determinando se há um perfil organizacional adequado ao Scrum, ou se o mesmo pode ser implantado ou adaptado para qualquer situação.

Como critérios de seleção para as fontes da pesquisa foram definidos: acessibilidade via web; disponibilidade gratuita; disponibilidade completa; e disponibilidade nos idiomas português ou inglês.

Dado estes critérios, optou-se pela escolha da biblioteca digital Scopus⁵ que possui uma máquina de busca, possibilitando uma pesquisa completa através da utilização da *string* de busca. A máquina de busca da Scopus foi a única selecionada para este trabalho, pois esta é bem abrangente, contendo em seu catálogo publicações de diversas bibliotecas digitais e de outras máquinas de busca (tais como: IEEE⁶, Springer⁷, ACM Digital Library⁸, entre outras), fornecendo eficientemente os resultados necessários [ISABEL 2011].

A pesquisa informal auxiliou a pesquisa formal, pois a partir da leitura das publicações obtidas foi possível estabelecer e aperfeiçoar a *string* de busca a ser utilizada na máquina de busca da SCOPUS, a saber:

(*TITLE-ABS-KEY ((scrum) AND (software OR software development OR software project) AND (organization OR organisation OR enterprise OR company)))*);

A execução da *string* de busca na fonte selecionada não assegura a validade das publicações obtidas. Por conseguinte, além da execução da *string* de busca realizou-se um procedimento de seleção a fim de verificar se todas as publicações estavam dentro do escopo, isto é, eram estudos de caso ou relatavam experiências da implantação do *framework* Scrum em organizações de desenvolvimento de *software*. O procedimento estabelecido possui três etapas que compreendem:

1. Verificação por título;
2. Verificação por resumos (*abstract*); e

⁵<https://www.scopus.com/>

⁶<http://ieeexplore.ieee.org/Xplore/home.jsp>

⁷<https://www.springer.com/br/>

⁸<http://dl.acm.org/>

3. Verificação do texto completo.

Após a realização do procedimento de seleção mencionado, para as publicações pertencentes ao escopo do trabalho, foi realizado um procedimento de obtenção de informações. Este procedimento consistiu na extração dos elementos estruturais apresentados nas seções 3.2, 3.3 e 3.4.

4.2.3 Execução da pesquisa

Após a definição do protocolo de pesquisa, executou-se o estudo, como definido na sequência.

Primeiramente, a *string* de busca (apresentada na subseção 4.2.2) foi executada na máquina de busca da biblioteca digital Scopus. Esta execução foi realizada em Julho/2016. A partir desta execução foram retornadas 240 publicações.

Seguidamente, foram realizadas as etapas do procedimento de seleção. Nas etapas 1 (verificação por título) e 2 (verificação por resumos) foi realizada uma verificação dupla, pela autora deste trabalho em conjunto com a orientadora. A etapa 3 (verificação do texto completo) foi realizada apenas pela autora.

A primeira etapa do procedimento de seleção das publicações foi menos restritiva, visto que nem todos os títulos são explícitos e não poderiam ser descartadas publicações relevantes. Esta foi primeiramente realizada pela autora deste trabalho e após também foi executada pela orientadora. Desta primeira etapa foram selecionadas 71 publicações.

Na segunda etapa, foi feita a leitura do resumo (*abstract*) das 71 publicações obtidas na etapa anterior. Esta seguiu a mesma ordem, sendo primeiramente realizada pela autora e depois pela orientadora. Através do resumo foi possível obter um melhor entendimento do conteúdo das publicações, restando 33 publicações.

Das 33 publicações selecionadas, na segunda etapa do procedimento de seleção, obteve-se acesso a 20. As demais estavam indisponíveis para *download* gratuito, não sendo possível ler o texto completo e realizar a terceira etapa do procedimento de seleção.

Para as 20 publicações obtidas, foi realizada a leitura completa a fim de cumprir a terceira etapa do procedimento. Desta etapa foram excluídas 2 publicações que não atendiam o escopo do trabalho, restando 18.

O procedimento de obtenção de informações foi realizado nas 18 publicações selecionadas formalmente, acrescentando as 14 publicações obtidas através da pesquisa informal. Neste procedimento, foi realizada a análise detalhada das publicações extraíndo os elementos estruturais do processo de desenvolvimento de *software*.

A seção 4.3 apresenta a análise dos resultados por visão: ambiente organizacional (seção 4.3.1), equipe de desenvolvimento (seção 4.3.2) e produto de *software* (seção 4.3.3). Na seção 4.3.4, é realizada uma comparação entre a análise apresentada por [BARBOSA e GREGÓRIO 2014] e a realizada neste trabalho.

4.3 Análise dos resultados

Conforme apresentado anteriormente, a pesquisa e a análise de publicações que relatam experiências da implantação do *framework* Scrum em organização de desenvolvimento de *software* possui o intuito de identificar os elementos estruturais comuns a essas organizações, e se há um determinado perfil organizacional que melhor se adequa ao Scrum ou se, por outro lado, o *framework* é adaptável, podendo ser implantado em qualquer organização e projeto.

A Tabela 4.3 apresenta os elementos estruturais identificados em cada publicação analisada. Estes elementos foram derivados de [BARBOSA e GREGÓRIO 2014], conforme apresentado nas seções 3.2, 3.3 e 3.4, com exceção do elemento “comunicação” que foi analisado devido sua importância para o *framework* Scrum. A Figura 4.1 apresenta a legenda dos elementos estruturais, assim como algumas abreviações utilizadas devido à limitação de espaço. A Tabela 4.2 traz a referência de cada um dos relatos analisados, já a Tabela 4.3 apresenta os elementos estruturais identificados em cada publicação.

Tabela 4.2: Referência dos Relatos

ID	Referência	Título
1	[JUNIOR 2013]	Scrum – Mais um caso de sucesso
2	[CARVALHO et al. 2009]	Aplicação Preliminar do Scrum no Desenvolvimento de Produtos de <i>Software</i> em uma Pequena Empresa de Base Tecnológica
3	[VARASCHIM 2009]	Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet
4	[HAJJDIAB et al. 2012]	<i>An Industrial Case Study for Scrum Adoption</i>
5	[LEIDEMER 2014]	Implantação de Scrum em uma empresa de Desenvolvimento de <i>Software</i>
6	[KROLL e AUDY 2013]	<i>Adopting Agile Methods for Follow-the-Sun Software Development</i>
7	[CARVALLHO et al. 2011]	Métodos Ágeis de Desenvolvimento de <i>Software</i> : Um caso prático de Aplicação do Scrum
8	[LIMA et al. 2012]	Implantação e Adaptação do Scrum em um Laboratório de Pesquisa e Desenvolvimento de Projetos de <i>Software</i>

Tabela 4.2 – Referência dos Relatos (continuação)

ID	Referência	Título
9	[SOARES et al. 2007]	Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de <i>Software</i>
10	[MONTEIRO 2015]	O Scrum como Alternativa ao Modelo em Cascata Aplicado à Zurich Portugal
11	[SCHATZ e ABDELSHAFI 2005]	<i>A Successful Transition to Agile Development</i>
12	[ROLLAND et al. 2016]	<i>Tailoring Agile in the Large: Experience and Reflections from a Large-Scale Agile Software Development Project</i>
13	[RAJPAL 2016]	<i>Lessons Learned from a Attempt at Distributed Agile</i>
14	[BENEFIELD 2008]	<i>Rolling out Agile in a Large Enterprise</i>
15	[KAUTZ et al. 2014]	<i>The Perceived Impact of the Agile Development and Project Management Method Scrum on Information Systems and Software</i>
16	[UHLE 2010]	<i>Agile Software Development in Small Projects</i>
17	[CRISTAL et al. 2008]	<i>Usage of SCRUM Practices within a Global Company</i>
18	[SOUZA et al. 2016]	<i>Using Scrum in Outsourced Government Projects: An Action Research</i>
19	[ELWER 2008]	<i>Agile Project Development at Intel: A Scrum Odyssey by Pat Elwer, Intel Corporation</i>
20	[PIEBER et al. 2016]	<i>University of Vienna's U:SPACE Turning Around a Large Project by Becoming Agile</i>
21	[MELO e FERREIRA 2010]	<i>Adoção de métodos ágeis em uma Instituição Pública de grande porte - um estudo de caso</i>
22	[LEITE e LUCRÉDIO 2014]	Desenvolvimento de <i>Software</i> utilizando o Framework Scrum: um Estudo de Caso
23	[BORGES 2011]	A Utilização de Práticas Scrum no Desenvolvimento de <i>Software</i> com Equipes Grandes e Distribuídas: um Relato de Experiência

Tabela 4.2 – Referência dos Relatos (continuação)

ID	Referência	Título
24	[SANTOS e CANEDO 2014]	<i>Agile methodology in the software development: Case study: Electoral justice of Brazil</i>
25	[TORRECILLA-SALINAS et al. 2013]	<i>Agile in Public Administration: Oxymoron or reality? An experience report</i>
26	[AYED 2014]	<i>Supported approach for agile methods adaptation: An adoption study</i>
27	[MOE et al. 2012]	<i>Challenges of shared decision-making: A multiple case study of agile software development</i>
28	[INAYAT et al. 2012]	<i>Facilitating an Off-Site Customer in Product-Based Agile Software Development: An Industrial Case Study</i>
29	[CABALLERO et al. 2011]	<i>Introducing Scrum in a Very Small Enterprise: A Productivity and Quality Analysis</i>
30	[BATRA et al. 2010]	<i>Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry</i>
31	[PAASIVAARA et al. 2008]	<i>Distributed Agile Development: Using Scrum in a Large Project</i>
32	[DINGSØYR et al. 2006]	<i>Developing Software with Scrum in a Small Cross-Organizational Project</i>

Tabela 4.3: Elementos estruturais por publicação

Ref.	Ambiente Organizacional					Equipe de desenvolvimento					Produto de Software			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Micro	N.D.	Baixa*	Méd.	N.D.	N.D.	N.D.	Méd.	N.D.	Centr.	Méd.	N.D.	Alta*	N.D.
2	Peq.	Méd.	Baixa*	Baixa*	N.D.	N.D.	Alto	Méd.	N.D.	Centr.	N.D.	Méd.	Alta*	Alta
3	Peq.	Alto	N.D.	Méd.*	Func.	N.D.	N.D.	Grande	N.D.	Centr.	N.D.	Baixa*	Alta*	Alta
4	Grande	N.D.	N.D.	Baixa*	Func.	N.D.	N.D.	Méd.	N.D.	Centr.	Alto	N.D.	Alta*	Alta
5	Peq.	N.D.	Baixa*	Baixa	Func.	N.D.	Méd.	Méd.	N.D.	Centr.	N.D.	Baixa	Baixa	Alta
6	Grande	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Méd.	N.D.	Desc.	Méd.	Méd.	N.D.	N.D.
7	N.D.	N.D.	Baixa*	Baixa*	N.D.	N.D.	N.D.	N.D.	N.D.	Centr.	Baixa	N.D.	Alta	Alta
8	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Baixo	Grande	N.D.	Centr.	Méd.	N.D.	N.D.	N.D.
9	Peq.	N.D.	N.D.	Méd.	Func.	N.D.	N.D.	N.D.	N.D.	Desc.	N.D.	Méd.a	Mediana*	N.D.
10	Grande	N.D.	N.D.	Baixa*	N.D.	N.D.	N.D.	Méd.	N.D.	Centr.	N.D.	N.D.	Alta*	Alta
11	Méd.	N.D.	N.D.	Baixa*	Func.	N.D.	N.D.	N.D.	N.D.	Centr.	N.D.	N.D.	Alta*	N.D.
12	Grande	N.D.	N.D.	N.D.	Func.	N.D.	N.D.	Grande	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.
13	Grande	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Desc.	N.D.	Méd.	N.D.	N.D.
14	Grande	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Alta*	Alta
15	Grande	N.D.	N.D.	Baixa*	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Alta*	N.D.
16	N.D.	N.D.	N.D.	Baixa	N.D.	N.D.	N.D.	Méd.	N.D.	N.D.	Baixo	Méd.	N.D.	N.D.
17	N.D.	N.D.	Alta*	Baixa*	N.D.	Méd.	N.D.	Méd.	N.D.	Desc.	N.D.	Baixa	Alta*	N.D.
18	N.D.	N.D.	N.D.	N.D.	N.D.	Peq.	N.D.	Peq.	N.D.	N.D.	Méd.	N.D.	N.D.	N.D.
19	Grande	N.D.	N.D.	Baixa*	Func.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Alta	N.D.
20	Grande	N.D.	N.D.	N.D.	N.D.	Grande	N.D.	N.D.	N.D.	Centr.	N.D.	N.D.	Alta	Alta

Tabela 4.3 – Tabela comparativa (continuação)

Ref.	Ambiente Organizacional						Equipe de desenvolvimento						Produto de Software	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
21	Grande	N.D.	N.D.	Méd.*	N.D.	N.D.	Alta	Méd.	N.D.	Centr.	Baixo	Baixo*	Alta*	N.D.
22	Grande	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Méd.	N.D.	N.D.	Baixo	Baixo	N.D.	N.D.
23	N.D.	N.D.	N.D.	N.D.	N.D.	Peq.	N.D.	Grande	N.D.	Desc.	N.D.	Baixo	N.D.	N.D.
24	N.D.	N.D.	N.D.	Méd.*	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	N.D.	Baixo*	Alta*	N.D.
25	N.D.	N.D.	Alta*	N.D.	Func.*	N.D.	N.D.	Peq.	N.D.	N.D.	Baixa	N.D.	N.D.	Baixa
26	N.D.	N.D.	N.D.	Baixa*	Func.	N.D.	N.D.	Méd.	N.D.	N.D.	Baixa	N.D.	Alta*	N.D.
27	Méd.	N.D.	Alta*	N.D.	N.D.	N.D.	N.D.	Méd.	N.D.	Desc.	N.D.	N.D.	N.D.	N.D.
28	Grande	N.D.	N.D.	Baixa	N.D.	N.D.	N.D.	Méd.	N.D.	N.D.	Alta	N.D.	N.D.	N.D.
29	Peq.	N.D.	Alta*	N.D.	N.D.	N.D.	N.D.	Peq.	N.D.	N.D.	Méd.	Méd.	N.D.	N.D.
30	N.D.	Baixo	Alta*	N.D.	N.D.	Grande	N.D.	N.D.	N.D.	Desc.	N.D.	Méd.	N.D.	Alta
31	Grande	N.D.	N.D.	N.D.	N.D.	Grande	N.D.	Grande	N.D.	Desc.	N.D.	Baixa	N.D.	N.D.
32	N.D.	N.D.	N.D.	Alta	N.D.	Peq.	N.D.	Peq.	N.D.	N.D.	N.D.	N.D.	Baixa	N.D.

Legenda							
Ambiente Organizacional		Equipe de Desenvolvimento		Produto de Software		Outros	
Tamanho	1	Nível de Especialização	7	Nível de Documentação Exigida	13	Não Definido	N.D.
Nível de investimento em tecnologia	2	Tamanho da Equipe	8	Mudança de Escopo	14	Centralizada	Centr.
Nível de Maturidade	3	Grau de Autonomia da Equipe	9			Descentralizada	DesCentr.
Interação com o cliente	4	Localização da Equipe	10			Funcional	Func.
Estrutura Organizacional	5	Experiência da Equipe	11			Pequeno(a)	Peq.
Porte de Projeto	6	Comunicação	12			Médio(a)	Méd.

Figura 4.1: Legenda da tabela de análise

A partir da análise das publicações, foi possível observar a carência de informações relatadas, não sendo possível extrair todos os elementos necessários e relevantes para este trabalho. Alguns elementos (tais como: “Interação com o cliente”, “Nível de comunicação entre a equipe” e “Nível de documentação exigida”) foram inferidos a partir do conhecimento do método utilizado pela organização. Sendo assim, quando o texto relatou que a organização seguia anteriormente o modelo cascata, por exemplo, inferiu-se que: a “Interação com o cliente” era baixa; e o “Nível de documentação exigida” era alto dado as características comuns às organizações que adotam este modelo. Já para organizações que utilizavam o RUP (no qual a comunicação se baseia nos artefatos e não de maneira direta e oral [UTIDA 2009]), inferiu-se que o “Nível de comunicação entre a equipe” era baixo.

Os resultados obtidos para o elemento “Nível de maturidade” também não foram descritos diretamente nas publicações, sendo estes inferidos a partir de trechos dos textos, quando possível. Os elementos que não foram obtidos diretamente no texto, mas sim a partir de inferência da autora desta pesquisa podem ser identificados na Tabela 4.3 através da utilização do caractere “*”. Para o elemento definido como “Grau de autonomia da Equipe” não foi possível obter um resultado satisfatório, pois o mesmo não foi identificado em nenhuma publicação analisada e não foi possível sua inferência a partir das informações fornecidas.

A partir dos 32 relatos analisados e apresentados na Tabela 4.3, foi realizada uma avaliação quantitativa por elemento, visando obter um perfil das organizações a partir do índice de ocorrência de cada um. Esta avaliação é apresentada nas subseções a seguir.

4.3.1 Ambiente organizacional

Com relação à visão “ambiente organizacional”, foram avaliados os elementos: tamanho da organização (Figura 4.2), nível de investimento em tecnologia (Figura 4.3), nível de maturidade (Figura 4.4), interação com o cliente (Figura 4.5), estrutura organizacional (Figura 4.6) e porte do projeto (Figura 4.7).

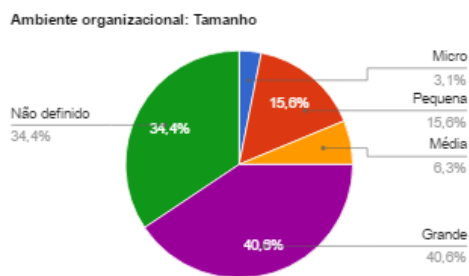


Figura 4.2: Tamanho

Ambiente organizacional: Nível de investimento em tecnologia

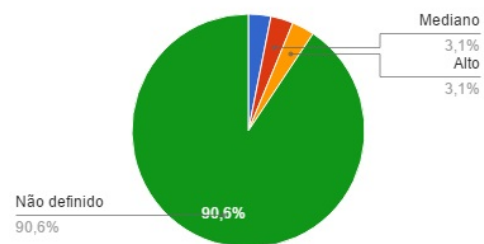


Figura 4.3: Nível de Investimento em TI

A partir da Figura 4.2, verifica-se que a maioria das organizações são grandes, ou seja, apresentam uma receita operacional bruta anual maior que R\$ 300 milhões ou, se avaliadas por quantidade de funcionários, possuem 100 colaboradores ou mais. Esta constatação da pesquisa não condiz com a maioria dos trabalhos que afirmam que o Scrum é mais apropriado para pequenas e médias organizações [BARBOSA e GREGÓRIO 2014]. Assim, há indícios de que o Scrum pode ser adaptado para diferentes tamanhos de organizações.

Quanto ao “nível de investimento em TI” (Figura 4.3) esta informação não estava descrita em 90,6% das publicações. Dentre as publicações que apresentam esta informação, houve o mesmo índice de ocorrência para os níveis baixo, médio e alto. Desta maneira é difícil determinar um resultado satisfatório a partir dos relatos avaliados.



Figura 4.4: Nível de maturidade

Ambiente organizacional: Nível de interação com o cliente

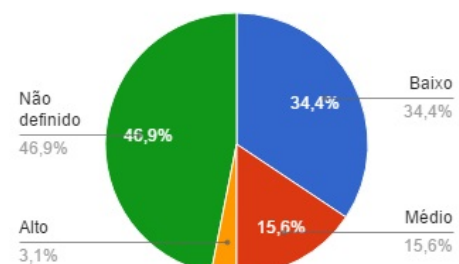


Figura 4.5: Interação com o cliente

Na Figura 4.4, observa-se que grande parte das publicações (71,9%) não apresentam informações sobre o “nível de maturidade”. Porém, das publicações (28,1%) que continham a informação relacionada a este elemento, 15,6% apresentaram um alto nível de maturidade, isto é, as organizações possuem maturidade de nível 3 (todos os projetos utilizam padrões formais organizacionalmente institucionalizados), nível 4 (uso de métricas para gerenciamento de projetos e para maximização do desempenho global da organização) ou nível 5 (lições aprendidas são estudadas rotineiramente para melhorar os processos). De acordo com estes resultados, há indícios de que um maior nível de maturidade é melhor

para a adoção do *framework*.

O Scrum preza pela constante interação com o cliente, mas é possível observar na Figura 4.5 que prevalece entre os relatos analisados o baixo nível de interação com o cliente nas organizações antes da implantação do *framework*. Consta-se então, que mesmo em organizações onde comumente o cliente só se envolve no início e no final do projeto, pode haver a implantação do Scrum e a adaptação da organização e do cliente para que este seja mais presente durante o desenvolvimento do produto.

Ambiente organizacional: Estrutura Organizacional

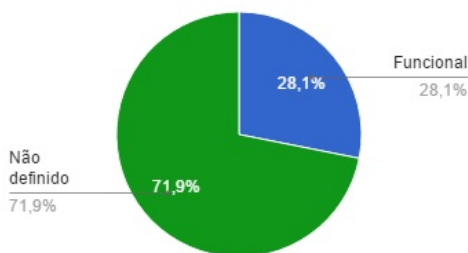


Figura 4.6: Estrutura organizacional

Ambiente organizacional: Porte do Projeto

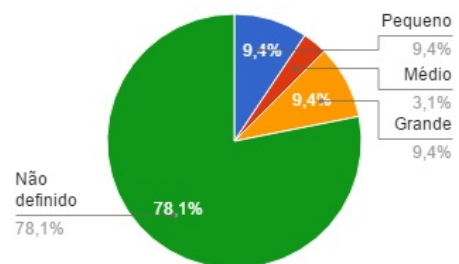


Figura 4.7: Porte de projeto

O elemento “estrutura organizacional” não foi informado pela maioria das publicações (71,9%). Em todos os relatos que apresentaram este elemento, as organizações eram funcionais clássicas, onde há uma hierarquia, de modo que, cada funcionário responde a um superior definido; os funcionários são agrupados por especialidades e os departamentos trabalham de modo independente uns dos outros [PMBOK 2013]. Um ponto positivo deste tipo de estrutura organizacional é que por ser dividida por especialidades e departamentos os membros de cada departamento possuem facilidade de comunicação por serem da mesma área, porém a comunicação interdepartamental tende a ser complicada e burocrática [SILVA 2011, PIRES 2010].

Conforme pode ser observado na Figura 4.7, o elemento “porte do projeto” também não é apresentado em grande parte das publicações (78,1%). Observou-se que parte das organizações optaram pela implantação do *framework* em projetos de pequeno porte (custo inferior a R\$ 200.000,00) ou grande porte (acima de R\$ 1.000.001,00). Esta constatação pode indicar que o Scrum pode ser implantado em projetos de tamanhos distintos.

4.3.2 Equipe de desenvolvimento

Quanto à perspectiva da “equipe de desenvolvimento” foram avaliados os seguintes elementos: “nível de especialização” (Figura 4.8), “tamanho da equipe” (Figura 4.9), “grau de autonomia da equipe” (Figura 4.10), “localização” (Figura 4.11), “nível de experiência” (Figura 4.12) e “nível de comunicação entre os membros da equipe” (Figura 4.13).



Figura 4.8: Nível de especialização

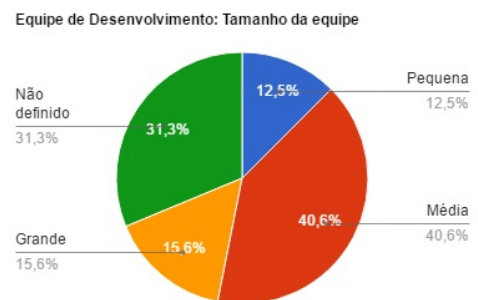


Figura 4.9: Tamanho da equipe

Para o elemento “nível de especialização” (Figura 4.8) a maioria das publicações não apresentam esta informação. Predominou a ocorrência de conhecimentos generalistas. Deste modo, é possível verificar indícios de que a implantação do Scrum é mais propícia em uma equipe, cujos membros estão aptos a atuar em mais de um papel durante o processo de desenvolvimento de *software*, conforme sugerido pelo guia do Scrum [SCHWABER e SUTHERLAND 2013]

Vários trabalhos, inclusive o guia do Scrum [SCHWABER e SUTHERLAND 2013], indicam que a equipe de tamanho médio é o ideal para a implantação do Scrum, pois em equipes pequenas pode haver limitação das habilidades necessárias ao Scrum e equipes grandes são difíceis de coordenar [SCHWABER e SUTHERLAND 2013]. Esta característica é congruente com os resultados obtidos neste trabalho, como pode ser observado na Figura 4.9, onde na maioria dos relatos as equipes que compõem os projetos possuem tamanho médio, isto é, são compostas de 5 a 9 membros.

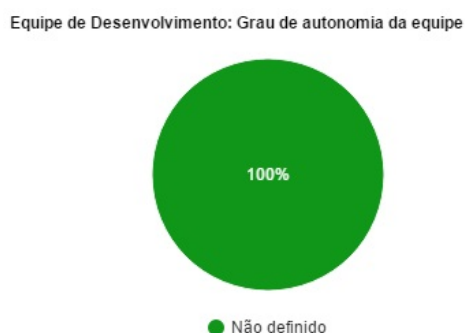


Figura 4.10: Grau de autonomia da equipe

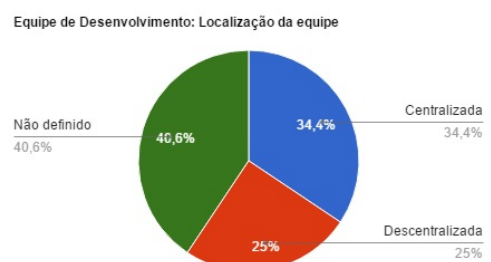


Figura 4.11: Localização da equipe

Para o elemento “Grau de autonomia da equipe”, não foi obtido nenhum resultado a partir dos relatos identificados no mapeamento sistemático. Portanto, não há como fornecer um parecer embasado na pesquisa realizada.

Na Figura 4.11, observa-se que a “localização da equipe” era principalmente centralizada. Para o Scrum os membros estarem dispostos todos no mesmo ambiente realmente é mais favorável, visto que este prioriza a interação e o compartilhamento de informações

entre a equipe. Mas, através dos resultados, é possível observar o aumento da implantação do Scrum em ambientes descentralizados, sendo necessária a adaptação do *framework* para tal.



Figura 4.12: Experiência da Equipe



Figura 4.13: Nível de comunicação entre a equipe

Com relação à experiência da equipe (Figura 4.12), parte das equipes apresentaram um baixo ou médio nível de experiência, isto é, em média é composta por membros com tempo de prática de até 4 anos. Isto sugere que mesmo equipes pouco habituadas com as técnicas de desenvolvimento utilizadas pela organização são capazes de entender e desempenhar as práticas do Scrum.

As reuniões diárias, além de permitir uma visão e a sincronização do trabalho realizado, possui o intuito de aproximar a equipe, melhorando a comunicação entre os membros e favorecendo o compartilhamento de conhecimentos. Considerando-se isto é possível perceber a importância da comunicação entre a equipe para o Scrum. No entanto, a Figura 4.13 mostra que, a maioria das equipes apresentaram um baixo nível de comunicação, principalmente aquelas dispostas de maneira descentralizada. Posto isto, há sinais de que o *framework* pode ser implantado até mesmo em equipes que enfrentam a incompatibilidade de horário e estão acostumadas a se comunicar através de artefatos e não de maneira direta e oral, conforme ideal sugerido pelo Scrum.

4.3.3 Produto de software

No que concerne à visão “produto de *software*” avaliou-se: o nível de documentação exigida (Figura 4.14) e a mudança de escopo (Figura 4.15).

Quanto ao “nível de documentação exigida”, observou-se que predominava nos relatos analisados, um alto nível de documentação para os produtos de *software* desenvolvidos, ou seja, 90% do processo de desenvolvimento era documentado. Um dos quatro valores descritos no “Manifesto Ágil” e respeitado pelo Scrum é que se deve priorizar o funcionamento do *software* em vez de gerar uma documentação abrangente. Percebe-se que mesmo organizações que comumente geram artefatos detalhados para todas as atividades são capazes de se adaptar com os valores do Scrum.

Atualmente, é comum ocorrer constantes mudanças durante o processo de desenvolvimento de *software*, o que se dá devido à globalização e ao aumento da concorrência

entre as organizações, ou até mesmo devido à dificuldade do cliente em definir os requisitos do sistema. A partir da Figura 4.15, nota-se que para a maioria das publicações (68,8%) não foi possível obter informações relativas ao elemento “mudança de escopo”. Porém, em 28,1% das publicações que possibilitaram a extração deste elemento, constatou-se que muitos projetos passam por alta “mudança de escopo” durante o processo de desenvolvimento. A capacidade do Scrum de comportar frequentes alterações o faz adequado a estas situações.

Produto de software: Nível de documentação exigida

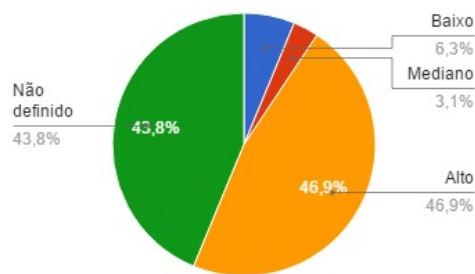


Figura 4.14: Nível de documentação exigida

Produto de software: Mudança de escopo

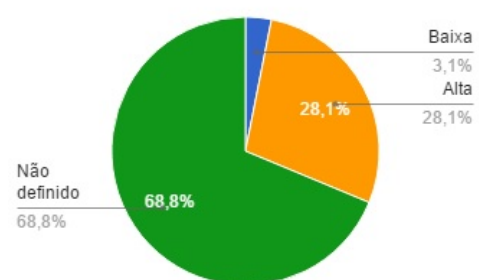


Figura 4.15: Mudança de escopo

4.3.4 Comparação

Barbosa e Gregório (2014) apresentam um modelo de decisão para adoção do método mais apropriado para ser utilizado em projetos de desenvolvimento. Os autores também definem quais elementos estruturais do processo de desenvolvimento de *software* são mais apropriados ao *framework* Scrum, mas não explicitam o critério utilizado para definir esta adequação.

Com o propósito de verificar a efetividade da análise realizada por [BARBOSA e GREGÓRIO 2014] para o Scrum, comparou-se o resultado obtido por estes autores com o presente trabalho. Para tal, considerou-se o resultado predominante obtido na análise das publicações, excluindo-se a opção “não definido”. Esta comparação é apresentada na Tabela 4.4.

Tabela 4.4: Tabela comparativa entre o resultado do mapeamento sistemático e [BARBOSA e GREGÓRIO 2014]

Elemento	[BARBOSA e GREGÓRIO 2014]	Resultado do mapeamento sistemático
Tamanho da organização	Média	Grande
Nível de investimento de TI	Mediano	Baixo/Mediano/Alto
Nível de maturidade	Mediano	Alto
Interação com o cliente	Alta	Baixo
Estrutura organizacional	Não definido	Funcional
Porte de projeto	Médio	Pequeno/Grande
Tamanho da equipe	Médio	Médio
Nível de especialização	Generalista	Generalista
Grau de autonomia	Alto	Não definido
Localização da equipe	Centralizada	Centralizada
Experiência da equipe	Médio	Baixo
Comunicação	Não definido	Baixo
Nível de documentação exigida	Mediano	Alto
Mudança de escopo	Alta	Alta

Diferentemente desta pesquisa que teve como resultado organizações de tamanho grande, Barbosa e Gregório (2014) definiram que idealmente a organização deve ser média, isto é, possuir uma receita operacional bruta anual maior que R\$ 16 milhões e menor ou igual a R\$ 90 milhões; e se avaliada por quantidade de funcionários possuir 50 a 99 colaboradores.

Houve também divergência entre as análises para os seguintes elementos: “nível de maturidade”, “interação com o cliente”, “porte de projeto”, “tamanho da equipe”, “nível de experiência da equipe” e “nível de documentação exigida”.

Quanto ao “nível de maturidade”, [BARBOSA e GREGÓRIO 2014] definiram que um nível mediano é o suficiente, de modo que a organização possua maturidade para saber como conduzir e organizar um projeto e gerar um produto de qualidade. O presente trabalho observou que, das organizações que permitiram inferir informações para este elemento, a maioria apresentou um alto “nível de maturidade”. Mesmo que um “nível de maturidade” mediano seja suficiente, é possível que seja preferível que a organização tenha um alto nível de maturidade para a implantar o Scrum.

De acordo com os autores é mais adequado ao Scrum que a “interação com o cliente” seja alta, tendo o cliente participação durante todo o processo de desenvolvimento. Mas a partir da análise realizada neste trabalho, constatou-se que mesmo organizações que tenham uma baixa interação com o cliente podem se adaptar, de forma que o cliente seja assíduo durante o processo de desenvolvimento.

Vale ressaltar que em grande parte das publicações analisadas no presente não foi possível obter informações relativas ao elemento “estrutura organizacional”, porém das informações extraídas obteve-se a maioria de organizações funcionais clássicas. Este elemento não foi avaliado por [BARBOSA e GREGÓRIO 2014].

Em [BARBOSA e GREGÓRIO 2014], os autores apresentaram que o Scrum é mais adequado para projetos de médio porte, ou seja, com custo entre R\$ 200.001,00 e R\$ 1.000.000,00. Tendo como justificativa a melhor adaptação às mudanças e o fornecimento satisfatório de documentação. Apesar do elemento “porte de projeto” não ter sido apresentado em grande parte das publicações analisadas, há indícios de que o Scrum pode ser adotado por projetos de diferentes tamanhos, visto que parte das organizações optaram pela implantação do *framework* em projetos de porte pequeno ou grande.

No presente trabalho, não foi obtido nenhum resultado para o elemento “grau de autonomia da equipe”. Sendo assim, não foi possível obter um parecer baseado na pesquisa realizada e nem mesmo fazer uma correspondência com [BARBOSA e GREGÓRIO 2014], que define que a equipe deve ter alta autonomia, isto é, a equipe deve ser capaz de se auto-organizar e tomar decisões sozinha.

Em grande parte das publicações analisadas, as equipes apresentaram um baixo nível de experiência o que diverge do conceito definido por [BARBOSA e GREGÓRIO 2014] de que o nível de experiência da equipe tem que ser ao menos médio, ou seja, a equipe deve ser composta em média por membros com 5 a 8 anos de experiência.

Houve também discordância entre as análises para o elemento “nível de documentação exigida”, onde [BARBOSA e GREGÓRIO 2014] determinaram que este deve ser mediano, de maneira que o processo de desenvolvimento do produto gere uma documentação parcial, porém sem exigência legal. E a partir da análise realizada neste trabalho infere-se que organizações que comumente geram um alto nível de documentação também são capazes de se adequar aos valores do Scrum.

No que se refere ao elemento “nível de investimento em TI”, a compatibilidade foi parcial, visto que para este trabalho o baixo, mediano e alto obtiveram a mesma proporção. E segundo os autores de [BARBOSA e GREGÓRIO 2014] o “nível de investimento em TI” deve ser mediano.

Houve equivalência para o “tamanho da equipe”, cujo resultado foi médio em ambos os trabalhos. E também para o elemento “nível de especialização”, que foi definido em ambas análises como generalista, ou seja, os membros de equipes com este nível de especialização possuem uma ou mais especialidades, sendo aptos a atuar em mais de um

papel no processo de desenvolvimento de *software*.

As análises também foram compatíveis para os elementos “localização da equipe” e “mudança de escopo”. O resultado para estes elementos foi uma equipe localizada no mesmo ambiente (centralizada) e alta mudança de escopo, isto é, mesmo que o escopo seja definido no início do projeto são permitidas alterações durante o projeto.

O elemento “nível de comunicação entre a equipe” não foi analisado por [BARBOSA e GREGÓRIO 2014]. Porém foi avaliado neste trabalho devido à sua relevância para o Scrum. Grande parte das equipes analisadas apresentaram um baixo nível de comunicação antes da implantação do Scrum.

4.4 Considerações finais

A partir da análise realizada não foi possível estabelecer um perfil estrutural comum às organizações que optam pela implantação do Scrum. Devido à insuficiência de informações nas publicações obtidas, houve uma grande incidência de elementos “não definidos”, impossibilitando uma melhor avaliação e determinação dos resultados.

O perfil definido por [BARBOSA e GREGÓRIO 2014] apresenta o caso de uma organização ideal para a implantação do Scrum, não sendo preciso uma grande transformação organizacional para a adoção do *framework*, assim minimizando o risco de fracasso do projeto. Mas, através da pesquisa realizada, observa-se que este perfil não é único e obrigatório, pois há relatos de sucesso de implantação do Scrum em organizações com outros perfis (às vezes, opostos aos apresentados em [BARBOSA e GREGÓRIO 2014]), conforme apresentado na seção anterior.

Ao examinar os resultados da análise e a comparação realizada, há indícios de que o Scrum é um *framework* que pode ser adaptado e implantado em qualquer organização e projeto. Organizações que já possuem características sugeridas no Scrum e realizam certas práticas definidas pelo *framework*, possivelmente terão mais facilidade para se habituar e aceitar a mudança organizacional. Mas a pesquisa indica que mesmo organizações que não apresentam nenhum fator comum ao Scrum podem obter sucesso com a adoção do *framework*.

A fim de avaliar esta análise realizada a partir da revisão da literatura, um questionário foi aplicado e uma entrevista foi planejada e realizada com profissionais que possuem experiência no uso e implantação do Scrum em organizações de desenvolvimento do *software*. O planejamento e a análise do questionário e da entrevista são apresentados no próximo capítulo.

CAPÍTULO 5

ENTREVISTA: SCRUM NAS ORGANIZAÇÕES

Este capítulo apresenta a pesquisa realizada utilizando como técnica a entrevista. Tal entrevista tem o intuito de agregar informações ao trabalho através dos conhecimentos de pessoas com prática e experiência na implantação e utilização do *framework* Scrum.

5.1 Introdução

Em muitas áreas é comum o uso de métodos de pesquisa qualitativa com o intuito de obter respostas a questões relacionadas ao comportamento humano. Em relação à Engenharia de *Software*, o uso de tais técnicas também tem se expandido e por muitas vezes são associados a métodos quantitativos [SEAMAN 1999].

A fim de obter mais informações e conhecimento quanto à implantação do Scrum e agregar valor à pesquisa realizada neste trabalho, foi realizada uma pesquisa, utilizando como técnica a entrevista. A entrevista é uma técnica de coleta de dados que dá relevância ao modo como o entrevistado detalha oralmente as informações [PRODANOV e FREITAS 2013]. Está incluída na categoria de interação frente a frente entre duas ou mais pessoas e refere-se a um diálogo que tem como foco um objetivo específico, e não apenas o prazer do diálogo propriamente dito [FRASER e GONDIM 2004].

A entrevista permite que sejam obtidas do entrevistado informações relativas a determinado tema [PRODANOV e FREITAS 2013]. Dependendo do intuito do entrevistador, a entrevista pode possuir vários objetivos, tais como: esclarecimento de dúvidas; coleta de informações ou impressões; descobrir planos de ação; averiguar fatos; entre outros. Há diferentes tipos de entrevistas, podendo ser [PRODANOV e FREITAS 2013, FRASER e GONDIM 2004, MANZIN 2004]: 1) padronizada e estruturada - segue-se um roteiro definido anteriormente e as perguntas são preestabelecidas; tal padronização auxilia na comparação de respostas; 2) semi-estruturada - defini-se como um roteiro composto de perguntas principais relacionadas ao tema, estas são completadas se necessário por outras

questões de acordo com o transcorrer da entrevista; 3) despadronizada e desestruturada - é mais informal, normalmente não possui um roteiro definido e 4) painel - periodicamente, as mesmas perguntas são feitas às mesmas pessoas a fim de analisar alterações nas respostas obtidas.

Em todos os casos as perguntas podem ser abertas ou fechadas. Nas perguntas abertas, o entrevistado fica livre para dar sua opinião e responder com suas próprias palavras, sem se limitar a selecionar uma entre várias opções pré-estabelecidas pelo entrevistador. Já nas perguntas fechadas, as respostas estão condicionadas a um padrão de opções dado pelo entrevistador [PRODANOV e FREITAS 2013].

Este capítulo está estruturado em outras cinco seções. A seção 5.2 descreve o planejamento da entrevista, cuja execução é apresentada na seção 5.3. A análise do questionário e a análise da entrevista são descritas nas seções 5.4 e 5.5, respectivamente. Por fim, a seção 5.6 apresenta as considerações finais.

5.2 Planejamento

Como já mencionado, a entrevista pode ser estruturada, semi-estruturada, desestruturada ou painel. Neste trabalho, optou-se por uma entrevista semi-estruturada, sendo tanto de caráter exploratório quanto para coleta de informações. Para este tipo de entrevista houve um planejamento, no qual as perguntas foram elaboradas com antecedência e foi definido um roteiro a ser seguido, não descartando a possibilidade de se fazer perguntas adicionais, caso necessário.

Em primeiro lugar, definiu-se o objetivo desta entrevista, sendo este obter informações sobre o *framework* Scrum de uma fonte com prática e experiência no assunto e que tem conhecimento sobre a adoção do Scrum em diversas organizações, priorizando a opinião do entrevistado(a) em relação aos elementos estruturais analisados neste trabalho e também quanto ao perfil das organizações que optam pela adoção do Scrum. Desta forma, espera-se obter uma visão além da literatura, completando o estudo realizado.

Segundo Prodanov e Freitas (2013), o entrevistado(a) deve ser alguém apto e com ciência do assunto em questão para que assim possa fornecer as informações necessárias. Sendo assim, optou-se por escolher pessoas com vivência e experiência no *framework* Scrum, sendo esta experiência não apenas teórica, mas também prática. Os entrevistados foram selecionados a partir dos contatos obtidos pela autora deste trabalho.

A entrevista foi precedida de um questionário, cujo planejamento é apresentado na subseção 5.2.1. Houve a necessidade do preenchimento do questionário antes da entrevista, pois haviam perguntas na entrevista baseadas em algumas respostas dadas no questionário. Na subseção 5.2.2 é apresentado o planejamento da entrevista propriamente dita.

5.2.1 Questionário

O questionário é um meio de coleta de informações composto por perguntas elaboradas de maneira clara e objetiva que devem ser respondidas por escrito pelo participante [PRODANOV e FREITAS 2013].

O questionário realizado neste trabalho foi estruturado em duas partes. A primeira parte tem o intuito de obter informações do entrevistado(a), e sua experiência com o *framework* Scrum; informações tais como formação acadêmica e certificações, conhecimento e experiência com métodos ágeis (especificamente o Scrum) etc. A segunda parte do questionário tem como objetivo saber a opinião do entrevistado(a) quanto ao nível de influência dos elementos estruturais estudados no presente trabalho na adoção do Scrum. Além disso, esta parte tem como propósito obter a opinião do(a) entrevistado(a) sobre algumas características de organizações e projetos que são importantes na adoção do *framework* e o quanto a ausência destas características pode afetar o processo de implantação.

Quanto à forma das perguntas do questionário, foram elaboradas combinando perguntas abertas e de múltipla escolha. As perguntas abertas fornecem autonomia ao entrevistado, de modo que forneça sua opinião; já as perguntas de múltipla escolha são fechadas e apresentam um conjunto de respostas possíveis [PRODANOV e FREITAS 2013].

As perguntas de múltipla escolha foram as relacionadas aos elementos estruturais, estas foram elaboradas através da utilização da Escala Likert de quatro níveis. A Escala Likert permite mensurar o grau de concordância do entrevistado com a afirmação proposta [LLAURADÓ 2015]. Neste trabalho, para cada elemento estrutural, os níveis apresentados aos entrevistados para seleção, foram: não impacta, possui baixo impacto, possui moderado impacto e possui grande impacto.

As perguntas foram disponibilizadas através de uma ferramenta do Google, o Formulários Google ¹, que gratuitamente permite a coleta de informações, seja em pequena ou grande escala. Por meio desta ferramenta é possível compartilhar o link do formulário, para que este possa ser respondido; e as respostas podem ser analisadas através de um resumo (quando há mais de uma pessoa respondendo) ou de modo individual.

O link do questionário foi enviado - via e-mail - aos convidados à entrevista e foi acompanhado de uma explicação sucinta do objetivo da pesquisa e instruções quanto ao preenchimento. O questionário completo é apresentado no Apêndice A.

5.2.2 Entrevista

As perguntas da entrevista (apresentadas no Apêndice B) tinham um objetivo mais abrangente, pois incluíam assuntos relacionados ao perfil organizacional, e à implantação do Scrum em geral. Com relação às perguntas relacionadas ao perfil organizacional, estas foram derivadas das respostas fornecidas no questionário com o intuito de obter mais

¹<https://www.google.com/intl/pt-BR/forms/about/>

detalhes sobre a opinião dos participantes.

As perguntas foram enviadas anteriormente ao(à) entrevistado(a) para que este tivesse ciência delas, porém as respostas foram obtidas durante a entrevista presencial.

Um roteiro para a entrevista foi definido com o intuito de facilitar a condução da entrevista. Neste roteiro foram definidos alguns passos, baseados em [PRODANOV e FREITAS 2013] e [SEAMAN 1999]:

- Solicitar permissão para gravação da entrevista;
- Apresentar o objetivo do trabalho, de modo resumido para não influenciar nas respostas;
- Fazer as perguntas, cuidadosamente, para que a entrevista não se torne um “questionário oral”, buscando um fluxo onde as perguntas surjam naturalmente, e caso haja necessidade, sejam feitas perguntas adicionais.

5.3 Execução

Para execução da entrevista (questionário e entrevista), optou-se por profissionais que possuem, além de experiência teórica, a experiência prática. Sendo assim, após o planejamento da entrevista, foi enviado o convite de participação a cinco profissionais da área. Destes, três responderam ao questionário e apenas um se propôs a participar da entrevista.

No convite, via e-mail, foi enviado o link do questionário, juntamente com uma breve explicação do objetivo do trabalho e as perguntas da entrevista (para conhecimento prévio). O convite foi enviado em Novembro/2016 à quatro participantes e em Janeiro/2017 ao participante sugerido pelo entrevistado.

Visto que algumas perguntas da entrevista estavam associadas às respostas dadas no questionário, assim que os entrevistados responderam ao questionário tais perguntas foram adaptadas.

Aos participantes foi dada a liberdade de optar pela entrevista via Skype² ou pessoalmente, respeitando sua disponibilidade. Como já mencionado, dos três participantes que responderam ao questionário apenas um se propôs a participar da entrevista, optando pela entrevista realizada pessoalmente, que ocorreu em Dezembro/2016.

Ao iniciar a entrevista solicitou-se a permissão para gravação, que foi concedida pelo entrevistado. A entrevista foi conduzida em ritmo de conversa, de modo que não se tornasse exaustiva, possibilitando também citar alguns pontos observados através do estudo teórico realizado a partir dos relatos das organizações. Por fim, foram acrescentadas poucas perguntas que surgiram no decorrer da entrevista.

²<https://www.skype.com/pt-br/>

5.4 Análise do questionário

Nesta seção, serão apresentadas as conclusões obtidas a partir das informações coletadas no questionário aplicado. Como já mencionado anteriormente, três participantes responderam ao questionário que serão identificados como participante A, participante B e participante C. O resumo do questionário é apresentado no Apêndice C.

Quanto à caracterização dos participantes, o participante A possui experiência de doze anos na área de TI, sendo 8 anos com o *framework* Scrum, atuando no início no Time de Desenvolvimento e depois como *Scrum Master*. O participante B possui dez anos de experiência com o desenvolvimento de *software*, utilizando métodos tradicionais e ágeis de desenvolvimento. Por cinco anos trabalhou em uma empresa que utiliza o Scrum em todos os seus projetos e que desenvolveu uma ferramenta para gestão de projetos com o *framework*. O participante C há anos pratica atividades relacionadas à Engenharia de *Software* e leciona sobre a disciplina. Desempenha o papel de tutor e mentor e fornece instruções concernentes ao desenvolvimento de *software*, além de atuar como *Scrum Master*, *Product Owner* e no Time de Desenvolvimento. Além disso, os participantes A e C possuem as certificações: CSP (*Certified Scrum Professional*), CSM (*Certified Scrum Master*) e CSPO (*Certified Scrum Product Owner*) e o participante B possui as certificações: CSM, CSPO e CSD (*Certified Scrum Developer*).

Sendo o principal objetivo deste trabalho definir, se possível, um perfil organizacional que seja mais adequado ao Scrum, buscou-se obter na opinião dos participantes características que sejam necessárias às organizações e projetos que desejam adotar o *framework*, e o quanto a ausência das características citadas pode afetar a implantação. Chegou-se à conclusão de que, apesar de não serem necessárias características específicas, é importante que a organização forneça autonomia para que a equipe trabalhe e que haja realmente o intuito de aprender e melhorar a forma de trabalho, ganhando eficácia. Quanto às características de projeto, o Scrum é muito útil quando se possui pouco conhecimento do negócio e os objetivos não estão claros. A ausência destas características não prejudica a implantação do Scrum, mas pode afetar os resultados obtidos e talvez não se obtenha todos os benefícios que a agilidade e o *framework* podem fornecer.

A partir do questionário, também foi possível obter o nível de impacto que cada elemento estrutural pode ter na implantação do Scrum, do ponto de vista de cada um dos participantes, representado nos gráficos apresentados a seguir.

Para os elementos “tamanho da organização” (Figura 5.1) e “nível de investimento da tecnologia” (Figura 5.2) os participantes A, B e C apresentaram respostas distintas, onde o impacto variou de nenhum a moderado. Disto pode-se concluir que mesmo que estes elementos precisem ser avaliados no momento da implantação, eles não afetam negativamente e a variação de tamanho das organizações e de nível de investimento em TI

não impossibilita a adoção do Scrum.

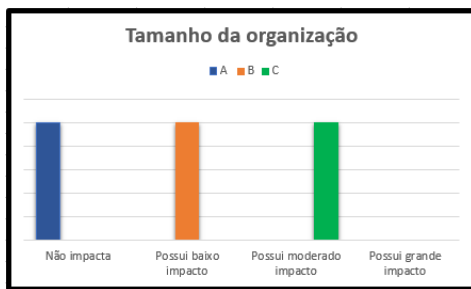


Figura 5.1: Tamanho da organização

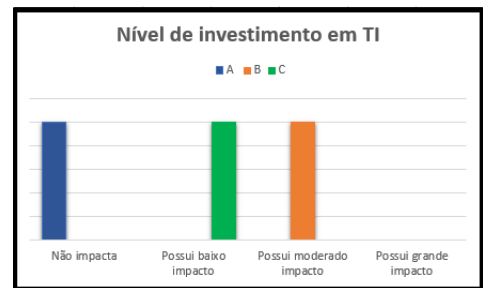


Figura 5.2: Nível de investimento em TI

Quanto ao elemento “nível de maturidade” (Figura 5.3), dos três participantes dois apresentaram a mesma opinião, declarando que o elemento possui impacto moderado e apenas um dos participantes afirmou que o elemento possui grande impacto na adoção do Scrum. Infere-se então que as organizações devem possuir maturidade em relação à gestão de projetos e quanto mais maturidade melhor para que o Scrum e seus conceitos possam ser aceitos.

O elemento “interação com o cliente” (Figura 5.4) ficou definido como de grande impacto. Isto posto, conclui-se que na opinião dos participantes a presença e contribuição do cliente é muito importante para que a implantação do Scrum possa ter sucesso e os resultados possam ser positivos. O cliente deve estar disposto a fazer parte do projeto e fornecer *feedbacks* para a melhoria do desenvolvimento do produto, e seja entregue um produto de qualidade que atenda às suas expectativas e necessidades.

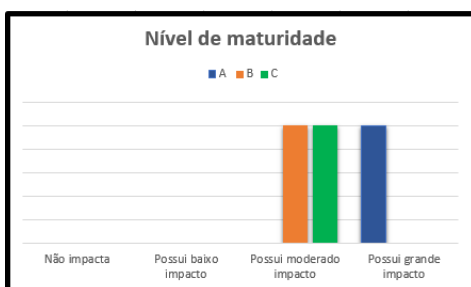


Figura 5.3: Nível de maturidade

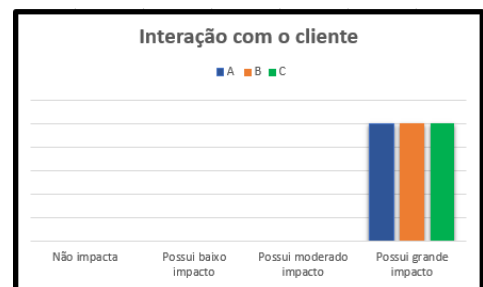


Figura 5.4: Interação com o cliente

Quanto à “estrutura organizacional” (Figura 5.5) no ponto de vista de dois participantes possui grande impacto e um definiu como impacto moderado. Ao indagar sobre as características necessárias para a adoção do Scrum, percebeu-se a importância da autonomia da equipe para o *framework*, e dependendo da estrutura da organização, se funcional, matricial ou projetizada pode ser que a equipe seja impossibilitada de tomar suas próprias decisões. Sendo assim, caso haja uma hierarquia definida na organização esta deve avaliar

como será feita a implantação do Scrum, e se será possível fornecer autonomia a equipe e ter uma transparência das atividades.

O elemento “porte de projeto” (Figura 5.6) para cada um dos participantes possui um impacto diferente variando de baixo a grande impacto. Neste trabalho, este elemento foi classificado com base nos valores de custo, um *software* mais complexo tende a custar mais caro, demandar mais tempo e esforço, e necessitar de uma equipe maior, posto isto, é preciso verificar se o Scrum atenderá as necessidades do projeto.



Figura 5.5: Estrutura organizacional

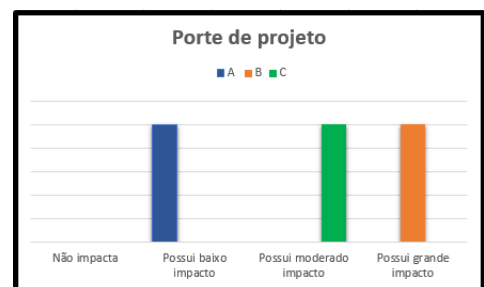


Figura 5.6: Porte do projeto

Em relação ao elemento “nível de especialização da equipe” (Figura 5.7), houve um consenso entre os participantes de que este é um elemento de impacto moderado. É preferível que os membros da equipe Scrum sejam multifuncionais, podendo desempenhar diversas tarefas dentro do projeto, para tal, devem possuir especialização e capacitação. Quanto ao “nível de experiência da equipe” (Figura 5.8), na opinião de um dos participantes possui baixo impacto, os outros dois definiram o elemento como de impacto moderado. Os conceitos do Scrum são fáceis de se entender, porém dependendo do contexto não são tão simples de se aplicar, e muitas vezes são necessárias adaptações no *framework*. Para isto é preciso que se tenha experiência, a fim de não causar o insucesso do projeto.

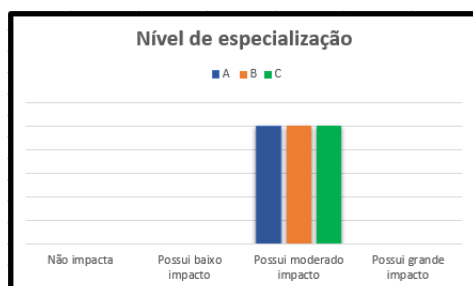


Figura 5.7: Nível de especialização da equipe

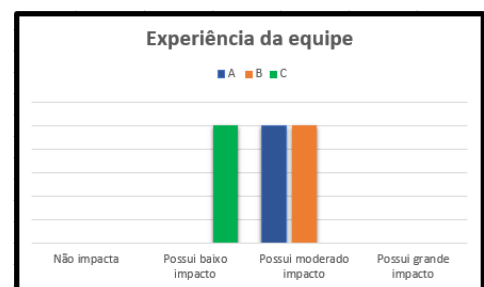


Figura 5.8: Experiência da equipe

Para os elementos “tamanho” (Figura 5.9) e “grau de autonomia” (Figura 5.10) da equipe de desenvolvimento, dois participantes definiram os elementos como de impacto

moderado e um os definiu como de grande impacto na implantação do Scrum. A equipe Scrum deve ser composta por uma quantidade de membros que facilite a coordenação e que possua capacidade de atender as necessidades do projeto, apresentando todas as habilidades necessárias. Sendo assim, o ideal é que a equipe não seja nem muito pequena e nem muito grande. Além disto, a equipe deve ser capaz e ter permissão para tomar decisões relacionadas ao projeto, ou seja, deve possuir autonomia para trabalhar.

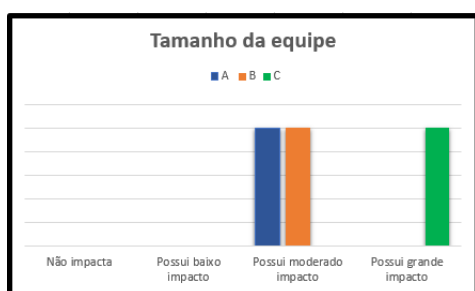


Figura 5.9: Tamanho da equipe

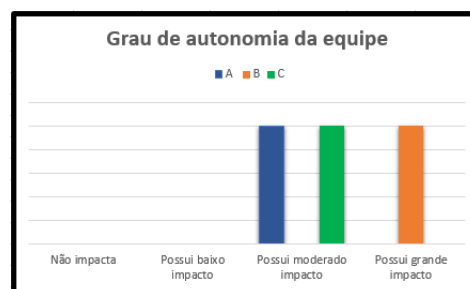


Figura 5.10: Grau de autonomia da equipe

Quanto à “comunicação” (Figura 5.11) entre os membros da equipe, dois participantes definiram os elementos como de impacto moderado e um os definiu como de grande impacto na implantação do Scrum. O Scrum preza a transparência; sendo assim, os membros (incluindo o cliente) devem interagir entre si, expondo as dificuldades, apresentando as atividades realizadas e o andamento do projeto de modo geral. O elemento “localização da equipe” (Figura 5.12) na opinião de um dos participantes não possui impacto na adoção do *framework*, os outros dois participantes definiram o elemento como de baixo impacto. Observa-se que apesar da comunicação e da interação entre os membros da equipe serem elementos importantes para o Scrum a localização da equipe não possui um impacto significativo na adoção do *framework*, pois as reuniões podem ser realizadas virtualmente, sendo preciso apenas avaliar se necessário a diferença de fuso horário.

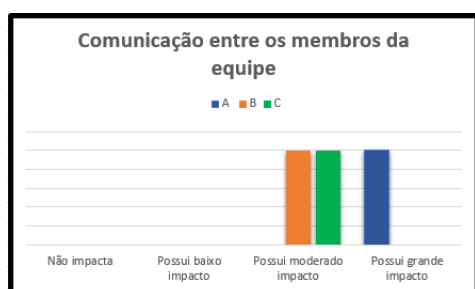


Figura 5.11: Comunicação

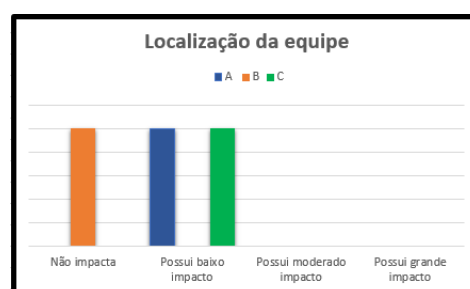


Figura 5.12: Localização da equipe

Quanto ao produto de *software*, avaliou-se os elementos “nível de documentação exigida” (Figura 5.13) e “mudança de escopo”(Figura 5.14). Para ambos, as opiniões dos

participantes foram diferentes. O elemento nível de documentação exigida apresentou um impacto variando de baixo a grande. De acordo com o *framework* Scrum, deve ser produzida somente a documentação necessária, de modo que não se perca tempo produzindo artefatos em excesso e que não serão utilizados. A partir das respostas obtidas, é possível entender que a exigência de uma grande quantidade de documentação não impossibilita a adoção do Scrum, mas pode privar o projeto de alguns benefícios da agilidade fornecidos pelo *framework*, pois gerar documentação desnecessária demanda um tempo que poderia ser empregado no desenvolvimento e melhoria do *software*.

Quanto ao nível de mudança de escopo, houve grande divergência entre os participantes, sendo avaliado como não possui impacto, possui baixo impacto e possui grande impacto. O Scrum é indicado para projetos onde não se tem prévio conhecimento de todos os requisitos do *software*; sendo assim, entende-se que para dois dos participantes, o Scrum pode ser adotado caso tanto o escopo esteja definido no início do projeto como não esteja. O participante B que definiu o elemento como de grande impacto, ao ser questionado quanto às características necessárias para adoção do Scrum, citou como características relevantes a falta de conhecimento do escopo total, e de meios para se alcançar o objetivo, e que a ausência dessas características diminui a contribuição do Scrum.

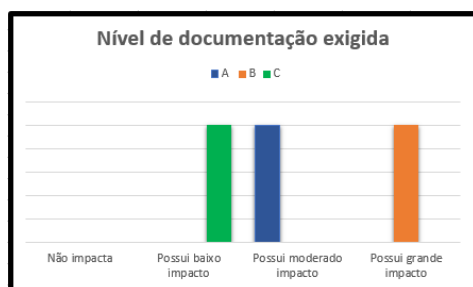


Figura 5.13: Nível de documentação exigida

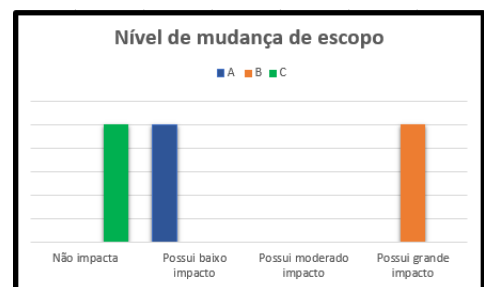


Figura 5.14: Nível de mudança de escopo

Na seção seguinte são apresentadas detalhadamente as considerações do participante A, que participou da entrevista, permitindo uma melhor avaliação de suas respostas.

5.5 Análise da entrevista

Nesta seção serão apresentadas as considerações embasadas nas respostas do participante A - que foi único a ter disponibilidade para a realização da entrevista - coletadas a partir do questionário e da entrevista. As subseções 5.5.1 e 5.5.2, apresentam as considerações quanto ao perfil das organizações que adotam o Scrum e quanto ao desenvolvimento de *software* em geral, respectivamente.

Vale ressaltar que neste texto não será realizada a transcrição fiel da entrevista

gravada. Serão considerados somente os pontos principais das respostas fornecidas pelo participante, tomando-se cuidado para que não seja alterada a ideia original.

5.5.1 Considerações relacionadas ao perfil organizacional e aos elementos estruturais estudados

A partir do estudo teórico realizado não foi possível definir um perfil mais adequado ao Scrum. Sendo assim, o objetivo desta parte da entrevista foi obter, na opinião do entrevistado, dada sua experiência, qual o nível de influência que os elementos estruturais estudados podem ter na adoção do Scrum. Na Figura 5.15 é possível observar, na opinião do entrevistado, o impacto dos elementos estruturais relacionados ao ambiente organizacional.

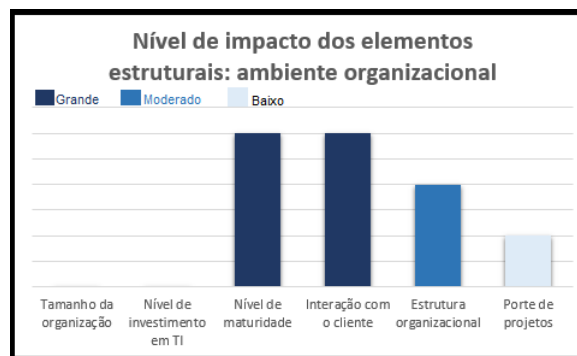


Figura 5.15: Nível de impacto dos elementos estruturais: ambiente organizacional

Quanto ao “tamanho da organização” e ao “nível de investimento em TI”, na opinião do entrevistado, estes elementos não impactam na adoção do *framework* Scrum. Do seu ponto de vista, dado que o Scrum é indicado para times multidisciplinares e pequenos (7 +/- 2 membros), se isto estiver sendo aplicado, não importa o tamanho da organização. No que diz respeito ao nível de investimento em TI, o participante declarou que este elemento não possui influência sobre o processo do Scrum em si; pode ser que influencie eventualmente na eficiência do desenvolvimento do produto. Disto entende-se que uma organização com baixo nível de investimento em TI pode implantar o Scrum, mas aquela que possui um certo nível de investimento pode ser que obtenha um processo de desenvolvimento de maior qualidade. O entrevistado declarou também que o uso do *framework* pode, inclusive, expor um problema relacionado a este elemento, possivelmente, relacionado à necessidade de melhoria na tecnologia utilizada pela organização.

O “porte de projeto” foi considerado como de baixo impacto pelo entrevistado, desde que seja respeitada a ideia de poucas pessoas compondo a equipe do projeto. Acrescentou também que, no caso de um projeto de pequeno porte, este pode ser composto por uma equipe pequena (7 +/- 2 membros) ou, caso seja de grande porte, o ideal é que

possua várias equipes Scrum, com poucos componentes cada. Então, é possível perceber que é possível ter mais de uma equipe Scrum trabalhando no mesmo projeto, porém responsáveis por módulos diferentes do *software* e que há a necessidade de equilibrar a quantidade de equipes com o porte de projeto para que não haja sobrecarga e a adoção do *framework* seja bem sucedida.

Com relação à “estrutura organizacional”, esta foi avaliada como de impacto moderado. “Se a gente tem uma maturidade, uma mentalidade ágil, a estrutura é indiferente, então, tem impacto? Tem! a gente vai precisar fazer alguns ajustes, eventualmente, até na própria estrutura; ela acaba se acertando em relação à agilidade”. Disto, infere-se que, se a organização possui maturidade e compreende todos os conceitos de agilidade proposto pelo Scrum, sua estrutura não será um obstáculo para a implantação do *framework*.

Os elementos “nível de maturidade” e “interação com o cliente” foram avaliados com grande impacto na adoção do Scrum. Quanto ao nível de maturidade, o participante declarou que o nível de maturidade é importante, pois se a organização não aceitar, verdadeiramente, o uso da agilidade e não passar a ter uma mentalidade ágil a adoção do Scrum pela equipe pode ser prejudicada. Pode ser que a equipe consiga, a princípio, adotar o Scrum, mas possivelmente encontrará impedimentos posteriores, dadas as interseções e dependências com outras pessoas.

Também avaliado como de grande nível de impacto, o elemento “interação com o cliente”, é considerado importante para o *framework* Scrum. Na opinião do entrevistado, o cliente é parte da solução, pois o produto agrega valor para ele; sendo assim, sua proximidade é essencial. Além disso, há a necessidade de que o cliente compreenda o funcionamento do *framework*. Conclui-se que esta compreensão por parte do cliente é necessária para que ele esteja disposto a realmente participar do processo e validar as entregas, e também entenda que as mudanças admitidas pelo Scrum devem ser realizadas a fim de agregar valor ao produto.

A Figura 5.15 apresenta, na opinião do entrevistado, o impacto dos elementos estruturais relacionados à visão de equipe de desenvolvimento.

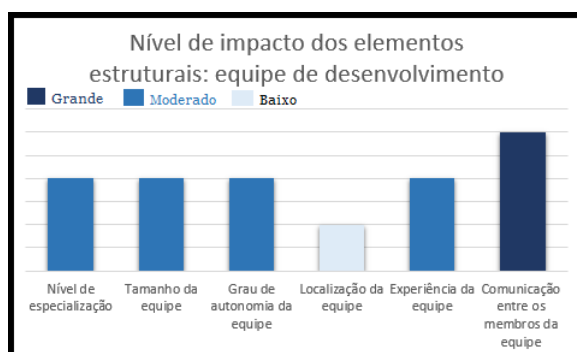


Figura 5.16: Nível de impacto dos elementos estruturais: equipe de desenvolvimento

O “nível de especialização” ficou classificado como de impacto moderado, pois de acordo com o entrevistado, é necessário que membros do time sejam aptos a desempenhar as diversas atividades dentro do projeto. O “tamanho da equipe” também ficou definida como de impacto moderado; levando em consideração o elemento anterior (nível de especialização), uma equipe com poucos membros pode não apresentar a aptidão necessária para a realização das tarefas, e caso seja composta por muitos membros o projeto pode não fluir corretamente.

Com relação ao “grau de autonomia da equipe”, o entrevistado classificou este elemento com um nível de impacto moderado. Segundo o participante, uma das premissas do Scrum é que o time seja auto-organizável, e caso o time não possa empregar sua autonomia, pode se desmotivar. Uma questão relevante apresentada pelo entrevistado, é que o Scrum propõe a ideia de auto-organização, que é diferente de autogerenciamento; sendo assim, nada impede que o time Scrum possua um gerente para auxiliá-lo.

É bom que a equipe interaja entre si e, se está localizada em um mesmo ambiente, a interação e comunicação fica mais fácil. Porém, o fato de a equipe estar distribuída não é impedimento para a adoção do Scrum. Sendo assim, a localização possui baixo impacto. Segundo o entrevistado, o ideal é que a equipe esteja centralizada, mas se a organização possui uma boa infraestrutura é possível que haja comunicação remota, permitindo assim implantar o Scrum.

Na opinião do entrevistado, a “experiência da equipe” também possui impacto moderado na adoção do Scrum. O ideal é que ao se implantar o Scrum a equipe entenda todos os conceitos, o porquê e a finalidade de sua aplicação. Portanto, é importante que a equipe tenha conhecimento e procure sempre obter mais experiência com relação ao *framework*, procurando aperfeiçoar as práticas.

A “comunicação entre os membros da equipe” foi classificada como de grande impacto. De acordo com as considerações do entrevistado, o conceito de time é muito importante dentro do Scrum e a agilidade está pautada na transparência; e a transparência é obtida através da comunicação, que evita esforço duplicado e auxilia na otimização das tarefas.

Na Figura 5.17, é apresentada a opinião do entrevistado com relação ao impacto dos elementos estruturais, tendo como visão o produto de *software*.

O “nível de documentação exigida” na opinião do entrevistado possui impacto moderado. A documentação é importante, mas é necessária uma avaliação a fim de definir o que é preciso documentar para que não haja excesso e seja gerada documentação desnecessária.

O “nível de mudança de escopo” foi classificado como de baixo impacto. No ponto de vista do entrevistado, “a ideia de usar o Scrum é exatamente porque a gente não consegue saber o escopo. A gente está em um ambiente de imprevisibilidades, um ambiente complexo e não tem conhecimento total”. Disso é possível inferir que o Scrum pode ser

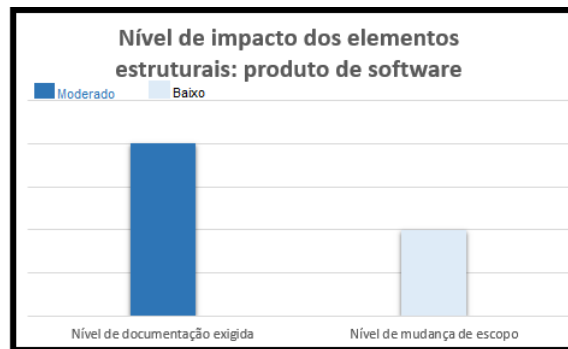


Figura 5.17: Nível de impacto dos elementos estruturais: produto de *software*

utilizado em projetos onde não se possui todos os requisitos definidos no início do projeto, sendo possível, caso necessário, inserir novas solicitações com um risco reduzido.

5.5.2 Considerações em relação ao desenvolvimento de software em geral

Há anos procura-se por modos de tornar o desenvolvimento de *software* mais eficiente, reduzindo custos, riscos, prazos e, por fim, entregando um produto de qualidade. Nestas circunstâncias, surge a dúvida se os métodos ágeis são ideais e mais vantajosos que os métodos tradicionais, principalmente para o atual mercado de desenvolvimento de *software*.

De acordo com o entrevistado, cada método é específico para um determinado contexto; isto posto, entende-se que os métodos ágeis não são mais vantajosos que os métodos tradicionais, e nem os métodos tradicionais são mais vantajosos que os métodos ágeis. A qualidade e eficiência de um método é definida de acordo com o contexto em que é aplicado. Conclui-se também que dado um contexto ágil, o uso de métodos ágeis é mais vantajoso e o uso de métodos tradicionais torna-se desfavorável, pois se o ambiente é propício à agilidade, a aplicação de métodos ágeis pode fornecer benefícios e ótimos resultados de maneira mais rápida.

Em relação à importância dos métodos ágeis para o atual mercado de desenvolvimento de *software* tem-se, no ponto de vista do entrevistado, que a importância está associada à vantagem que os métodos ágeis fornecem meios de se conseguir explorar melhor novas soluções para desenvolver determinado produto. Além disso, os métodos ágeis possuem um processo mais leve, e fornecem a possibilidade de construir um produto muito mais consciente do que está sendo feito e de muito mais valor para o cliente.

O Scrum é um *framework* muito promissor, podendo sua adoção, em um contexto adequado, ser muito favorável à organização. No que diz respeito ao *framework* Scrum possuir vantagens sobre os demais métodos ágeis, a partir da opinião do participante, é possível inferir que a vantagem do Scrum está na simplicidade de seus conceitos, na faci-

lidade de compreender seus eventos, papéis e artefatos. Sendo muito útil, principalmente, em times que estão iniciando em agilidade, pois permite um melhor entendimento.

Mesmo com a simplicidade dos conceitos do *framework* Scrum, sua adoção pode trazer conflitos dependendo da estrutura da organização. Segundo o entrevistado, uma das dificuldades está relacionada à ausência de conhecimento; em não conhecer adequadamente o *framework* e não explorar todos os seus conceitos. Além da falta de conhecimento, há dificuldade quanto ao conceito de auto-organização por parte da equipe, pois muitas vezes os limites não são impostos. Esta falta de limites pode fazer com que a equipe tome decisões que não tem permissão para tomar ou fazer com que a equipe se sinta limitada para trabalhar, quando por parte da organização a equipe poderia possuir mais liberdade.

A partir do estudo teórico realizado, foi possível observar que durante o processo de adoção do Scrum muitas organizações ao adaptar o *framework* às suas necessidades, tendem a não implantar alguns conceitos (artefatos, eventos, papéis, etc). No que se refere a esta adaptação do *framework*, na opinião do entrevistado, ela até pode ser realizada, mas o ideal é que a equipe já tenha experiência com o Scrum original (como apresentado no guia do Scrum). Desta forma, a decisão de modificação ou retirada de algum conceito deve ser tomada a partir de um processo de reflexão e que seja feito por questão de eficiência; modificações realizadas sem experiência para tal, podem causar insucesso. Também de acordo com o participante, caso realmente seja necessária a adaptação, é importante que seja, ao menos, realizada a reunião de retrospectiva. Em sua opinião este é um dos principais eventos do Scrum e o ideal é que não deixe de ser realizado, pois é nesta reunião que se expõem os problemas e se discutem as soluções.

5.6 Considerações finais

Neste capítulo foi apresentado o planejamento e a análise do questionário e da entrevista aplicada a profissionais com experiência teórica e prática no *framework* Scrum, através da qual pudemos obter informações relevantes quanto ao impacto que cada elemento estrutural, na opinião dos participantes, pode possuir na implantação do Scrum. Além disso, apresentou-se opiniões sobre questões relacionadas ao desenvolvimento de *software* em geral.

A partir da entrevista infere-se que não é necessário um determinado perfil organizacional para que o Scrum seja implantado, corroborando os achados na pesquisa teórica. Baseando-se no ponto de vista dos participantes, há algumas características que podem auxiliar na adoção do *framework*, tais como: valorização das pessoas, autonomia e motivação do time, transparência do objetivo a ser atingido, interesse pela aprendizagem organizacional e a presença do cliente. Porém, a ausência destas características não impede diretamente a implantação do Scrum, mas pode afetar a qualidade do resul-

tado obtido e os benefícios da agilidade. Desta forma, a análise de alguns dos elementos apresentados neste estudo, pode ser relevante no momento da avaliação de viabilidade de adoção do *framework*.

A Tabela 5.1 fornece um resumo geral da análise e do questionário aplicado, apresentando a comparação entre a classificação fornecida por Barbosa e Gregório (2014) para o Scrum e os resultados obtidos através do mapeamento sistemático, além disso, apresenta o nível de impacto da adoção do Scrum na opinião dos especialistas.

Tabela 5.1: Resumo geral da análise

Elemento Estrutural	Classificação		Nível de Impacto na adoção do Scrum		
	Barbosa e Gregório (2014)	Resultado do Mapeamento Sistemático	A	B	C
Tamanho da organização	Média	Grande	Não impacta	Baixo Impacto	Moderado Impacto
Nível de investimento de TI	Mediano	Baixo/Mediano/Alto	Não impacta	Moderado impacto	Baixo impacto
Nível de maturidade	Mediano	Alto	Grande impacto	Moderado impacto	Moderado impacto
Interação com o cliente	Alta	Baixo	Grande impacto	Grande impacto	Grande impacto
Estrutura organizacional	Não definido	Funcional	Moderado impacto	Grande impacto	Grande impacto
Porte de projeto	Médio	Pequeno/-Grande	Baixo impacto	Grande impacto	Moderado impacto
Tamanho da equipe	Médio	Médio	Moderado impacto	Moderado impacto	Grande impacto
Nível de especialização	Generalista	Generalista	Moderado impacto	Moderado impacto	Moderado impacto
Grau de autonomia	Alto	Não definido	Moderado impacto	Grande impacto	Moderado impacto
Localização da equipe	Centralizada	Centralizada	Baixo impacto	Não impacta	Baixo impacto
Experiência da equipe	Médio	Baixo	Moderado impacto	Moderado impacto	Baixo impacto

Tabela 5.1 – Tabela comparativa (continuação)

Elemento Estrutural	Classificação		Nível de Impacto na adoção do Scrum		
	Barbosa e Gregório (2014)	Resultado do Mapeamento Sistemático	A	B	C
Comunicação	Não definido	Baixo	Grande impacto	Moderado impacto	Moderado impacto
Nível de documentação exigida	Mediano	Alto	Moderado impacto	Grande impacto	Baixo impacto
Mudança de escopo	Alta	Alta	Baixo impacto	Grande impacto	Não impacta

O capítulo seguinte apresenta as considerações finais do presente trabalho, suas contribuições, suas limitações e os trabalhos futuros.

CAPÍTULO 6

CONCLUSÃO

Este capítulo apresenta as considerações finais deste trabalho, suas contribuições, suas limitações e os trabalhos futuros.

6.1 Considerações finais

Para que se obtenha sucesso na utilização de um método de desenvolvimento de *software*, é necessário avaliar o contexto em que este será aplicado. Um método aplicado inadequadamente pode ocasionar o insucesso em sua adoção. Sendo assim, há situações mais propensas a métodos ágeis e outras aos métodos tradicionais.

Este trabalho apresentou o Scrum, um *framework* voltado para o gerenciamento de projetos, que possui como base os princípios e valores apresentados no Manifesto Ágil. O Scrum tem o intuito de otimizar o processo de desenvolvimento de *software*, prevenindo riscos e falhas, e comportando mudanças no decorrer do processo.

O Scrum engloba em sua estrutura pilares, papéis, eventos, artefatos e regras bem definidas e simples. Porém, a simplicidade de seus conceitos não torna sua implantação fácil, pois primeiramente deve-se avaliar o contexto em que será aplicado, e verificar se este é propício à agilidade. Além disso, deve-se ter conhecimento e aceitação da finalidade de cada prática proposta pelo *framework*. O ideal é que o Scrum seja adotado como um todo para se obtenha bons resultados.

É possível que a adoção seja realizada em um contexto já ágil, ou seja, em um ambiente propício aos métodos ágeis, onde não serão necessárias muitas adaptações para implantação. Porém, nem sempre o contexto em que se tentará aplicar o Scrum será ágil; sendo assim, há a necessidade de avaliar este contexto e verificar se há a viabilidade ou não da adoção do *framework*.

Neste contexto, este trabalho realizou um estudo com o objetivo de identificar um perfil organizacional mais adequado ao Scrum, a fim de auxiliar organizações de desen-

volvimento de *software* na análise de seu perfil e assim verificar se há ou não viabilidade de implantação. O estudo teórico foi realizado a partir de publicações de organizações de desenvolvimento de *software* que descrevem sua experiência na adoção do Scrum.

Além do estudo teórico, com o intuito de avaliar a relevância dos elementos estruturais selecionados para análise e agregar informações gerais ao trabalho, foi elaborado e aplicado um questionário e realizada uma entrevista com profissionais com experiência teórica e prática no Scrum.

A partir do estudo realizado, não foi possível determinar um perfil organizacional que seja mais adequado ao Scrum. Sendo assim, infere-se que o *framework* pode ser adotado por qualquer organização desde que o contexto seja propício à agilidade. Ou seja, mesmo que a organização seja tradicional, há a possibilidade de adaptação e aceitação das práticas, de modo que posteriormente não surjam barreiras que impeçam a realização do Scrum como um todo.

Mesmo que não tenha sido identificado um perfil organizacional adequado ao Scrum, a partir do questionário e da entrevista foi possível obter, tendo como base as opiniões dos participantes, o nível de impacto que cada elemento pode ter no momento da implantação do Scrum. Desta forma, este estudo auxilia na análise de viabilidade de implantação do *framework*, permitindo que as organizações possam focar nos elementos que possuem mais impacto e verificar a necessidade de adaptação. E assim possam se adequar para que os resultados sejam satisfatórios, e a implantação do Scrum não venha fracassar.

O Scrum mostrou-se adaptável a qualquer contexto, sendo assim, ele apresenta a proposta de um processo que pode ser seguido de várias formas, de acordo com a necessidade de cada organização e projeto. Por exemplo, o Scrum propõe reuniões diárias; estas podem ser realizadas presencialmente, caso a equipe seja centralizada, mas também pode ser realizada remotamente, caso a equipe seja distribuída; o importante é que as reuniões sejam realizadas. Como dito anteriormente, o Scrum funciona como uma guia, mas não fornece uma fórmula pronta que, se aplicada, todos os problemas serão solucionados. Ele fornece ideias que devem ser trabalhadas pela equipe e implantadas de acordo com as circunstâncias.

Um fator que foi possível identificar como essencial para implantação é a mentalidade tanto da organização, da equipe, quanto do cliente. Se estes compreendem a finalidade de cada papel, evento etc. e contribuem para que sejam realizados, há grandes chances do Scrum ser implantado com sucesso e os benefícios da agilidade serem atingidos.

6.2 Contribuições

A principal contribuição deste trabalho foi prover uma análise sobre os elementos estruturais críticos para a implantação do Scrum.

A partir dos resultados de um mapeamento sistemático e da experiência de especialistas na área, os elementos estruturais foram avaliados, permitindo que as organizações analisem os elementos mais relevantes ao verificar a viabilidade de implantação.

Também foram apresentados, baseando-se nos relatos estudados, algumas informações relevantes que podem auxiliar no início da implantação do Scrum. Além de apresentar os principais problemas que indicam a necessidade de se implantar o Scrum, as dificuldades encontradas e os benefícios obtidos com a implantação.

6.3 Limitações

Dentre as limitações deste trabalho, destacam-se:

- A escolha de apenas alguns elementos estruturais do processo de desenvolvimento de *software*. Desta forma, as organizações que queiram avaliar a viabilidade de implantação do *framework* deverão se embasar apenas nestes elementos;
- A impossibilidade de acesso completo a todas as publicações obtidas através do mapeamento sistemático reduziu o conjunto analisado. Além disso, a escassez de informações relevantes ao trabalho impossibilitou uma melhor avaliação e determinação de resultados;
- A escolha e a quantidade de entrevistas. Foram contactados outros profissionais para a realização da entrevista, porém não foi possível sua execução devido à falta de disponibilidade;
- A amostra das organizações na pesquisa teórica pode estar enviesada, pois o mapeamento sistemático só identificou as organizações que possuem hábito de publicarem seus resultados, o que não é padrão das organizações de desenvolvimento de *software*.

6.4 Trabalhos futuros

Como um dos trabalhos futuros, pretende-se realizar um estudo de campo mais abrangente, a fim de conhecer experiências de organizações de desenvolvimento de *software* que adotaram o Scrum e identificar informações que não foram possíveis de obter através das publicações, do questionário e da entrevista.

Além disto, espera-se, se possível definir um processo que auxilie organizações que pretendem adotar o *framework* Scrum, tornando sua adaptação mais fácil e evitando os riscos causados por uma mudança estrutural e cultural muito grande.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ABRANTES e TRAVASSOS 2013]ABRANTES, J. F.; TRAVASSOS, G. H. *Towards pertinent characteristics of agility and agile practices for software processes*. *CLEI Electronic Journal*, Centro Latinoamericano de Estudios en Informática, v. 16, n. 1, p. 6, 2013.
- [AMÉRICO 2004]AMÉRICO, T. F. S. *XWEBPROCESS: Um processo ágil para o desenvolvimento de aplicações web*. Dissertação (Mestrado), 2004. Universidade Federal de Pernambuco, Recife.
- [AYED 2014]AYED, H. *Supported Approach for Agile Methods Adaptation: An Adoption Study*. In: ACM. *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*. [S.l.], 2014. p. 36–41.
- [BARBOSA e GREGÓRIO 2014]BARBOSA, C. N.; GREGÓRIO, T. F. Modelo de decisão para processo de desenvolvimento de software utilizando práticas de mercado. 2014. Monografia - Universidade do Estado do Rio de Janeiro - UERJ RJ.
- [BATRA et al. 2010]BATRA, D. et al. *Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry*. *Communications of the Association for Information Systems*, v. 27, n. 1, p. 21, 2010.
- [BECK et al. 2001]BECK, K. et al. *Manifesto para o desenvolvimento ágil de software*. 2001. Disponível em: <http://manifestoagil.com.br/>. Acesso em: Fevereiro/2016.
- [BENEFIELD 2008]BENEFIELD, G. *Rolling out agile in a large enterprise*. In: Hawaii International Conference on System Sciences, Proceedings of the 41st Annual - IEEE. [S.l.], 2008. p. 461.
- [BNDES 2011]BNDES. *Normas Reguladoras do Produto BNDES Automático*. 2011. Disponível em: http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes_

- [pt/Galerias/Arquivos/produtos/download/Circ034_11.pdf](#). Acesso em: Março/2016.
- [BONATO 2002]BONATO, A. S. F. *Extreme Programming e qualidade de software*. 2002. Monografia- Escola Politécnica -Universidade de São Paulo.
- [BORGES 2011]BORGES, A. N. J. A utilização de práticas scrum no desenvolvimento de software com equipes grandes e distribuídas: um relato de experiência. In: *V WDDS-Workshop on Distributed Software Development*. [S.l.: s.n.], 2011.
- [BORGES 2010]BORGES, E. N. *Conceitos e Benefícios do Test Driven Development*. 2010. Disponível em: <http://www.inf.ufrgs.br/~cesantin/TDD-Eduardo.pdf>. Acesso em: Janeiro/2016.
- [CABALLERO et al. 2011]CABALLERO, E. et al. *Introducing scrum in a very small enterprise: a productivity and quality analysis*. In: SPRINGER. *European Conference on Software Process Improvement*. [S.l.], 2011. p. 215–224.
- [CARVALHO et al. 2009]CARVALHO, B. V. et al. Aplicação preliminar do scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. In: *XXIX Encontro Nacional De Engenharia De Produção - Salvador, BA, Brasil*. [S.l.: s.n.], 2009.
- [CARVALHO et al. 2011]CARVALHO, C. E. C. D. et al. Métodos ágeis de desenvolvimento de *software*: Um caso prático de aplicação do scrum. In: ENEGEP. *XXXI Encontro Nacional de Engenharia de Produção*. [S.l.], 2011.
- [CRISTAL et al. 2008]CRISTAL, M. et al. *Usage of Scrum practices within a global company*. In: IEEE. *2008 IEEE International Conference on Global Software Engineering*. [S.l.], 2008. p. 222–226.
- [DICIONÁRIO 2009-2016]DICIONÁRIO. 2009–2016. Disponível em: <https://www.dicio.com.br/>. Acesso em: Março/2016.
- [DINGSØYR et al. 2006]DINGSØYR, T. et al. *Developing software with scrum in a small cross-organizational project*. In: SPRINGER. *European Conference on Software Process Improvement*. [S.l.], 2006. p. 5–15.
- [ELWER 2008]ELWER, P. *Agile Project Development at Intel: A Scrum Odyssey. Danube Case Study: Intel Corporation*, p. 1–14, 2008.
- [FARIA 2011]FARIA, A. *Melhorando a visibilidade dos projetos de desenvolvimento*. 2011. Disponível em: https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/december_20_2011_6_33_am1?lang=en. Acesso em: Fevereiro/2016.

- [FRASER e GONDIM 2004]FRASER, M. T. D.; GONDIM, S. M. G. Da fala do outro ao texto negociado: discussões sobre a entrevista na pesquisa qualitativa. *Paidéia*, SciELO Brasil, v. 14, n. 28, p. 139–152, 2004.
- [FREIRE 2009]FREIRE, F. Desvendando o scrum. *Revista TI Digital*, Arteccom, p. 36–43, 2009. Disponível em: <http://www.artecom.com.br/revistatidigital/downloads/Scrum.pdf> - Acesso em: Setembro/2016.
- [HAJJDIAB et al. 2012]HAJJDIAB, H. et al. *An industrial case study for scrum adoption*. *Journal of Software*, v. 7, n. 1, p. 237–242, 2012.
- [INAYAT et al. 2012]INAYAT, I. et al. *Facilitating an Off-Site Customer in Product-Based Agile Software Development: An Industrial Case Study*. In: SPRINGER. *International Multi Topic Conference*. [S.l.], 2012. p. 210–221.
- [ISABEL 2011]ISABEL, S. L. S. *Seleção de Abordagens de Teste para Aplicações Web*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2011.
- [JUNIOR 2013]JUNIOR, A. R. *Scrum—Mais um caso de sucesso*. 2013. Disponível em: <http://www.uniedu.sed.sc.gov.br/wp-content/uploads/2015/01/Artigo-Alvarino-Ribeiro-Junior.pdf>. Acesso em: Agosto/2016.
- [JUNIOR e SANTOS 2010]JUNIOR, D. d. A.; SANTOS, C. M. *Desenvolvimento de Software: Processos Ágeis ou Tradicionais? Uma visão crítica*. 2010. Disponível em: http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010_4.pdf. Acesso em: Novembro/2016.
- [KAUTZ et al. 2014]KAUTZ, K. et al. *The perceived impact of the agile development and project management method scrum on information systems and software development productivity*. *Australasian Journal of Information Systems*, v. 18, n. 3, 2014.
- [KITCHENHAM e CHARTERS 2007]KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing systematic literature reviews in software engineering*. In: *Technical report, Ver. 2.3 EBSE Technical Report*. EBSE. [S.l.: s.n.], 2007.
- [KROLL e AUDY 2013]KROLL, J.; AUDY, J. L. N. *Adopting agile methods for follow-the-sun software development*. In: *Proceedings of the Nineteenth Americas Conference on Information Systems*. [S.l.: s.n.], 2013.
- [LEIDEMER 2014]LEIDEMER, R. H. *Implantação de scrum em uma empresa de desenvolvimento de software*. 2014. Monografia - Centro Universitário UNIVATES.

- [LEITE e LUCRÉDIO 2014]LEITE, L. M.; LUCRÉDIO, D. Desenvolvimento de software utilizando o *Framework* scrum: um estudo de caso. *Tecnologias, Infraestrutura e Software*, Universidade Federal de São Carlos - UFSCar, v. 3, n. 2, p. 114–121, 2014. ISSN 231 6-2872.
- [LIMA et al. 2012]LIMA, I. R. et al. Implantação e adaptação do scrum em um laboratório de pesquisa e desenvolvimento de projetos de software. *Revista de Sistema de Informação, Macaé*, n. 9, p. 16–23, 2012.
- [LLAURADÓ 2015]LLAURADÓ, O. *Escala de Likert: O que é e como utilizá-la*. 2015. Disponível em: <https://www.netquest.com/blog/br/blog/br/escala-likert>. Acesso em: Janeiro/2017.
- [MANZIN 2004]MANZIN, E. J. *Entrevista semi-estruturada: Análise de objetivos e de roteiros*. 2004. Disponível em: <http://www.sepq.org.br/IIisipeq/anais/pdf/gt3/04.pdf>. Acesso em: Dezembro/2016.
- [MARQUES 2012]MARQUES, A. N. Metodologias ágeis de desenvolvimento: Processos e comparações. 2012. Monografia - Faculdade de Tecnologia de São Paulo.
- [MARTINS 2014]MARTINS, J. G. F. *Proposta de método para classificação do porte das empresas*. Dissertação (Mestrado) — Universidade Potiguar-UnP, 2014.
- [MASSIMUS 2016]MASSIMUS. *Quais as diferenças entre Métodos Ágeis e Tradicionais na gestão de projetos?* 2016. Disponível em: <http://massimus.com/2016/04/metodos-ageis-e-tradicionais-na-gestao-de-projetos-qual-diferenca/>. Acesso em: Novembro/2016.
- [MELLO, SILVA e TRAVASSOS 2014]MELLO, R.; SILVA, P.; TRAVASSOS, G. Agilidade em processos de software: Evidências sobre características de agilidade e práticas ágeis. In: *XIII - Simpósio Brasileiro de Qualidade de Software. SBQS*. [S.l.: s.n.], 2014.
- [MELO e FERREIRA 2010]MELO, C. O.; FERREIRA, G. R. M. Adoção de métodos ágeis em uma instituição pública de grande porte-um estudo de caso. In: *Workshop Brasileiro de Métodos Ágeis (Agile Brasil 2010)*. [S.l.]: Centro de Eventos da PUCRS (CEPUC), Porto Alegre, 2010.
- [MENDES 2011]MENDES, F. *Ágil e Agilidade*. 2011. Disponível em: <https://fernandomendespereira.wordpress.com>. Acesso em: Fevereiro/2016.
- [MIRANDA 2011]MIRANDA, R. *Uma Revisão Sistemática Sobre Equipes de Desenvolvimento de Software: Tipologia, Características e Critérios de Formação*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2011.

- [MOE et al. 2012]MOE, N. B. et al. *Challenges of shared decision-making: A multiple case study of agile software development*. *Information and Software Technology*, Elsevier, v. 54, n. 8, p. 853–865, 2012.
- [MONTEIRO 2015]MONTEIRO, C. S. P. R. C. *O Scrum como Alternativa ao Modelo em Cascata Aplicado à Zurich Portugal*. Dissertação (Mestrado) — Universidade Europeia - Escola de Tecnologias, Artes e Comunicação, 2015.
- [PAASIVAARA et al. 2008]PAASIVAARA, M. et al. Distributed agile development: Using scrum in a large project. In: IEEE. *2008 IEEE International Conference on Global Software Engineering*. [S.l.], 2008. p. 87–95.
- [PEREIRA et al. 2013]PEREIRA, A. et al. *Funcionamento do Scrum*. 2013. Disponível em: <http://ned.unifenas.br/cursosgratuitos/201302/scrum/funcionamento.html>. Acesso em: Novembro/2016.
- [PIEBER et al. 2016]PIEBER, B. et al. *University of Vienna’s U: SPACE Turning Around a Failed Large Project by Becoming Agile*. In: SPRINGER. *International Conference on Agile Software Development*. [S.l.], 2016. p. 217–225.
- [PIRES 2010]PIRES, E. R. *O estilo da sua empresa*. 2010. Disponível em: <http://ogerente.com.br/rede/projetos/o-estilo-da-sua-empresa>. Acesso em: Novembro/2016.
- [PMBOK 2013]PMBOK, G. Um guia do conjunto de conhecimentos em gerenciamento de projetos. In: *Project Management Institute*. [S.l.: s.n.], 2013. ISBN 978-1-62825-007-7.
- [PORTO e SILVA 2014]PORTO, K. T.; SILVA, L. A. Dificuldades na implantação do scrum: Estudo de casos múltiplos. 2014. Monografia - Universidade Federal de Alfenas.
- [PRIKLADNICKI, WILLI e MILANI 2014]PRIKLADNICKI, R.; WILLI, R.; MILANI, F. *Métodos Ágeis para Desenvolvimento de Software*. [S.l.]: Bookman Editora, 2014. ISBN 978-85.
- [PRODANOV e FREITAS 2013]PRODANOV, C. C.; FREITAS, E. C. D. *Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico-2ª Edição*. [S.l.]: Editora Feevale, 2013.
- [RAJPAL 2016]RAJPAL, M. *Lessons Learned from a Failed Attempt at Distributed Agile*. In: Springer. *International Conference on Agile Software Development*. [S.l.], 2016. p. 235–243.
- [ROLLAND et al. 2016]ROLLAND, K. H. et al. *Tailoring Agile in the Large: Experience and Reflections from a Large-Scale Agile Software Development Project*. In: Springer. *International Conference on Agile Software Development*. [S.l.], 2016. p. 244–251.

- [RUOSO 2009]RUOSO, D. *Diferenças entre métodos ágeis e tradicionais*. 2009. Disponível em: <http://softwarelivre.org/ruoso/blog/diferencas-entre-metodos-ageis-e-tradicionais>. Acesso em: Novembro/2016.
- [SAMI 2014]SAMI, M. *User Story Estimation Techniques in Agile*. 2014. Disponível em: <http://www.mohammadsami.com/blogs/2014/03/user-story-estimation-techniques-in-agile/>. Acesso em: Novembro/2016.
- [SANTOS e CANEDO 2014]SANTOS, V. S. A. D.; CANEDO, E. *Agile methodology in the software development: Case study: Electoral justice of Brazil*. In: IEEE. *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.], 2014. p. 1–6.
- [SCHATZ e ABDELSHAFI 2005]SCHATZ, B.; ABDELSHAFI, I. *Primavera Gets Agile: A Successful Transition to Agile Development*. *IEEE software*, v. 3, n. 7, 2005.
- [SCHWABER e SUTHERLAND 2013]SCHWABER, K.; SUTHERLAND, J. *Um guia definitivo para o Scrum: As regras do jogo*. [S.l.], 2013. 19 p.
- [SEAMAN 1999]SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, IEEE, v. 25, n. 4, p. 557–572, 1999.
- [SEBRAE 2013]SEBRAE. *Anuário do trabalho na micro e pequena empresa*. 2013. Disponível em: [http://www.sebrae.com.br/Sebrae/PortalR\\$\\$\\$20Sebrae/Anexos/AnuarioR\\$\\$\\$20doR\\$\\$\\$20TrabalhoR\\$\\$\\$20NaR\\$\\$\\$20MicroR\\$\\$\\$20eR\\$\\$\\$20PequenaR\\$\\$\\$20Empresa_2013.pdf](http://www.sebrae.com.br/Sebrae/PortalR$$$20Sebrae/Anexos/AnuarioR$$$20doR$$$20TrabalhoR$$$20NaR$$$20MicroR$$$20eR$$$20PequenaR$$$20Empresa_2013.pdf). Acesso em: Janeiro/2016.
- [SEBRAE 2014]SEBRAE. *Ramos de atividades*. 2014. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/ramos-de-atividades,8ef89e665b182410VgnVCM100000b272010aRCRD>. Acesso em: Março/2016.
- [SILVA 2011]SILVA, J. R. D. *Tipos de Estruturas Organizacionais*. 2011. Disponível em: <http://www.efetividade.blog.br/tipos-de-estruturas-organizacionais/>. Acesso em: Novembro/2016.
- [SILVEIRA et al. 2013]SILVEIRA, G. A. et al. Fatores condicionantes do nível de maturidade em gerenciamento de projetos: um estudo empírico em empresas brasileiras. *Revista de Administração*, Universidade de São Paulo, FEA-Departamento de Administração, v. 48, n. 3, p. 574, 2013.
- [SINÉSIO et al. 2013]SINÉSIO, T. et al. *Crystal Clear Methodologies*. 2013. Disponível em: <http://pt.slideshare.net/ThiagoSinsio/metodologia-crystal-clear>. Acesso em: Julho/2016. Centro Universitário UNA, Belo Horizonte – MG.

- [SOARES et al. 2007]SOARES, F. S. F. et al. Adoção de scrum em uma fábrica de desenvolvimento distribuído de *Software*. In: *I Workshop de Desenvolvimento Distribuído de Software. João Pessoa*. [S.l.: s.n.], 2007.
- [SOMMERVILLE 2003]SOMMERVILLE, I. *Engenharia de software*. [S.l.]: Pearson Addison Wesley São Paulo, 2003.
- [SOUZA 2013]SOUZA, G. D. S. Um estudo de caso sobre adoção de práticas Ágeis em um ambiente tradicional. 2013. Monografia - Universidade Federal do Estado do Rio de Janeiro - UNIRIO.
- [SOUZA et al. 2016]SOUZA, T. L. D. et al. *Using Scrum in Outsourced Government Projects: An Action Research*. In: IEEE. *2016 49th Hawaii International Conference on System Sciences (HICSS)*. [S.l.], 2016. p. 5447–5456.
- [TAKEUCHI e NONAKA 1986]TAKEUCHI, H.; NONAKA, I. The new new product development game. *Harvard Business Review*, v. 64, n. 1, p. 137–146, 1986.
- [TEIXEIRA et al. 2005]TEIXEIRA, D. D. et al. *DSDM–Dynamic Systems Development Methodology*. 2005. Disponível em: [http://paginas.fe.up.pt/~aaguilar/es/artigos\\$\\$\\$20 finais/es_final_14.pdf](http://paginas.fe.up.pt/~aaguilar/es/artigos$$$20 finais/es_final_14.pdf). Acesso em: Julho/2016.
- [TELES 2006]TELES, V. M. *Práticas do XP*. 2006. Disponível em: <http://www.desenvolvimentoagil.com.br/xp/praticas>. Acesso em: Outubro/2016.
- [TORRECILLA-SALINAS et al. 2013]TORRECILLA-SALINAS, C. et al. *Agile in Public Administration: Oxymoron or reality? An experience report*. In: CAISE. [S.l.], 2013.
- [UHLE 2010]UHLE, J. *Agile Software Development in Small Projects*. 2010. Monografia - Hasso Plattner Institut.
- [UTIDA 2009]UTIDA, K. H. Metodologias tradicionais e metodologias ágeis: Análise comparativa entre *Rational Unified Process* e *Extreme Programming*. 2009. Monografia - Faculdade de Tecnologia do Estado de São Paulo.
- [VARASCHIM 2009]VARASCHIM, J. Implantando o scrum em um ambiente de desenvolvimento de produtos para internet. 2009. Monografia - Pontifícia Universidade Católica do Rio de Janeiro.
- [VASCONCELOS e PEREIRA 2012]VASCONCELOS, B.; PEREIRA, C. H. Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. *Gestão & Produção*, SciELO Brasil, São Carlos, v. 19, n. 3, p. 557–573, 2012.

[VERGOPIA 2008]VERGOPIA, C. *Project review maturity and project performance: an empirical case study*. [S.l.]: ProQuest, 2008.

APÊNDICE A

QUESTÕES DO QUESTIONÁRIO

Este questionário possui o objetivo de caracterizar o perfil do entrevistado, bem como coletar suas opiniões sobre o perfil das organizações para a adoção do Scrum.

*** Obrigatório**

Parte 1 - Caracterização do entrevistado

1. Qual é sua formação acadêmica? *

Marcar apenas uma oval.

- ☐ Curso de graduação em andamento
- ☐ Curso de graduação concluído
- ☐ Especialização em andamento
- ☐ Especialização concluída
- ☐ Mestrado em andamento
- ☐ Mestrado concluído
- ☐ Doutorado em andamento
- ☐ Doutorado concluído

2. Qual é o seu grau de conhecimento sobre métodos ágeis, especificamente o Scrum?*

Marque todas que se aplicam.

- ☐ Não tenho nenhum conhecimento sobre Scrum
- ☐ Eu tenho um conhecimento superficial sobre Scrum
- ☐ Eu estudei/estudo Scrum em um ou mais cursos/disciplinas
- ☐ Eu estudei/estudo Scrum em um ou mais livros
- ☐ Eu trabalhei/trabalho com Scrum assumindo o papel de *Scrum Master*
- ☐ Eu trabalhei/trabalho com Scrum assumindo o papel de *Product Owner*
- ☐ Eu trabalhei/trabalho com Scrum compondo o Time

3. Possui certificações ágeis em Scrum? Se sim, quais?*
4. Qual é a sua experiência profissional relacionada às atividades de Engenharia de Software de uma forma geral?*
5. Qual é a sua experiência profissional relacionada especificamente com à aplicação do Scrum?*

Parte 2 Perfil das organizações que adotam o Scrum

1. Na sua opinião, quais características uma organização deve possuir para que possa adotar adequadamente o Scrum?*
2. Na sua opinião, quais características um projeto deve possuir para que possa adotar adequadamente o Scrum?*
3. O quanto a ausência das características mencionadas por você nas questões anteriores pode afetar a implantação do Scrum?*
4. Tendo como visão o ambiente organizacional, na sua opinião o quanto que cada elemento abaixo influencia na adoção do Scrum:*

	Não impacta	Possui baixo impacto	Possui moderado impacto	Possui grande impacto
Tamanho da organização	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nível de investimento em tecnologia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nível de maturidade(em relação à gestão de projetos)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interação com o cliente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estrutura organizacional(funcional, matricial ou por projeto)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Porte de projetos(em relação ao custo do projeto)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Tendo como visão a equipe de desenvolvimento, na sua opinião o quanto que cada elemento abaixo influencia na adoção do Scrum:*

	Não impacta	Possui baixo impacto	Possui moderado impacto	Possui grande impacto
Nível de especialização	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tamanho da equipe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Grau de autonomia da equipe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Localização da equipe (centralizada ou distribuída)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Experiência da equipe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Comunicação entre os membros da equipe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Tendo como visão o produto de software, na sua opinião o quanto que cada elemento abaixo influencia na adoção do Scrum:*

	Não impacta	Possui baixo impacto	Possui moderado impacto	Possui grande impacto
Nível de documentação exigida	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nível de mudança de escopo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

APÊNDICE B

QUESTÕES DA ENTREVISTA

Parte 3 - Questões para a entrevista

1. Perguntas baseadas nas respostas fornecidas para as questões 4, 5 e 6 apresentadas na Parte 2 - perfil das organizações que adotam o Scrum - do questionário.

a. Tendo como visão o ambiente organizacional:

- i. Por que o elemento tamanho organizacional não impacta na adoção do Scrum?
- ii. Por que o elemento nível de investimento em tecnologia não impacta na adoção do Scrum?
- iii. Por que o elemento nível de maturidade possui grande impacto na adoção do Scrum?
- iv. Por que o elemento interação com o cliente possui grande impacto na adoção do Scrum?
- v. Por que o elemento estrutura organizacional possui impacto moderado na adoção do Scrum?
- vi. Por que o elemento porte de projeto possui baixo impacto na adoção do Scrum?

b. Tendo como visão a equipe de desenvolvimento:

- i. Por que o elemento nível de especialização possui impacto moderado na adoção do Scrum?
- ii. Por que o elemento tamanho da equipe possui impacto moderado na adoção do Scrum?
- iii. Por que o elemento grau de autonomia da equipe possui impacto moderado na adoção do Scrum?
- iv. Por que o elemento localização da equipe possui baixo impacto na adoção do Scrum?
- v. Por que o elemento experiência da equipe possui impacto moderado na adoção do Scrum?
- vi. Por que o elemento comunicação entre os membros da equipe possui grande impacto na adoção do Scrum?

- c. Tendo como visão o produto de software:
- i. Por que o elemento nível de documentação possui impacto moderado na adoção do Scrum?
 - ii. Por que o elemento nível de mudança de escopo possui baixo impacto na adoção do Scrum?
2. Você considera que os métodos ágeis possuem vantagens sobre os métodos tradicionais? Se sim, qual seria sua principal vantagem?
3. Qual a importância dos métodos ágeis para o atual mercado de desenvolvimento de software?
4. Na sua opinião há vantagens do Scrum em relação a outros métodos ágeis? Se sim, qual (is) seria (am)?
5. Com seu tempo de experiência, qual a maior dificuldade que as organizações enfrentam ao implantar o Scrum?
6. Na sua opinião qual principal e mais importante prática e característica do *framework*?
7. Na sua opinião, comumente, qual o maior erro das organizações durante a adoção do Scrum?
8. Há algum relato de fracasso com a implantação do *framework*? Se sim, o que ocasionou?
9. Durante a adoção ao Scrum algumas organizações tendem a adaptar o *framework* à sua necessidade. Na sua opinião esta adaptação pode trazer riscos e até mesmo o insucesso do projeto?
10. Durante a adoção caso haja adaptações o que não pode deixar de ser aplicado?
11. O Scrum é mais indicado para projetos que ainda não iniciaram ou podem ser implantado em qualquer momento do projeto?

APÊNDICE C

RESUMO DO QUESTIONÁRIO

1. Na sua opinião, quais características uma organização deve possuir para que possa adotar adequadamente o Scrum?(3 respostas)

- A. A organização deve acima de tudo buscar valorizar as pessoas, dando autonomia aos times e sendo transparente quanto aos propósitos a serem alcançados. Prezar a aprendizagem organizacional, buscando formas de fazê-la acontecer.
- B. Pouca hierarquia, possibilidade de realizar mudanças, e, principalmente, ter o poder de decisão sobre o escopo que deve ser realizado.
- C. Isso é uma pergunta muito ampla, mas não vejo necessidade de características específicas, além da vontade de adotar com o intuito de melhorar sua forma de trabalho e eficácia.

2. Na sua opinião, quais características um projeto deve possuir para que possa adotar adequadamente o Scrum?(3 respostas)

- A. Projetos onde há pouco conhecimento à respeito do negócio em si. Que seja composto por um time pequeno. Que o cliente (dono do produto) atue colaborativamente para o incremento do produto. Que o time tenha autonomia e motivação para adotar o Scrum.
- B. Deve ser um projeto onde não se sabe qual o escopo total, apenas se tem uma ideia de onde se quer chegar, mas o como se atinge o objetivo não é claro.
- C. Deve poder ser trabalhado em lotes, incrementalmente.

3. O quanto a ausência das características mencionadas por você nas questões anteriores pode afetar a implantação do Scrum?(3 respostas)

- A. A ausência afeta pouco a implantação. Ela afeta mesmo os resultados obtidos. Pois o Scrum pode até mesmo ser adotado em um ambiente de comando-controle. Vai ajudar o time a organizar seus trabalhos, mas não vai ser obtido os benefícios do uso da Agilidade.

B. Bastante, pois em um projeto onde já se sabe tudo que deve ser feito, o Scrum não contribui muito.

C. Um projeto que depende de fluxo contínuo não se beneficiará muito do Scrum.

4. O quanto que cada elemento estrutural influencia na adoção do Scrum, no ponto de vista dos participantes:

Elemento	A	B	C
Tamanho da organização	Não impacta	Possui baixo impacto	Possui moderado impacto
Nível de investimento em tecnologia	Não impacta	Possui moderado impacto	Possui baixo impacto
Nível de maturidade (em relação à gestão de projetos)	Possui grande impacto	Possui moderado impacto	Possui moderado impacto
Interação com o cliente	Possui grande impacto	Possui grande impacto	Possui grande impacto
Estrutura organizacional (funcional, matricial ou por projeto)	Possui moderado impacto	Possui grande impacto	Possui grande impacto
Porte de projetos (em relação ao custo do projeto)	Possui baixo impacto	Possui grande impacto	Possui moderado impacto
Nível de especialização	Possui moderado impacto	Possui moderado impacto	Possui moderado impacto
Tamanho da equipe	Possui moderado impacto	Possui moderado impacto	Possui grande impacto
Grau de autonomia da equipe	Possui moderado impacto	Possui grande impacto	Possui moderado impacto
Localização da equipe (centralizada ou distribuída)	Possui baixo impacto	Não impacta	Possui baixo impacto
Experiência da equipe	Possui moderado impacto	Possui moderado impacto	Possui baixo impacto
Comunicação entre os membros da equipe	Possui grande impacto	Possui moderado impacto	Possui moderado impacto
Nível de documentação exigida	Possui moderado impacto	Possui grande impacto	Possui baixo impacto
Nível de mudança de escopo	Possui baixo impacto	Possui grande impacto	Não impacta

Figura C.1: Impacto dos elementos na opinião dos participantes