



Pós-Graduação em Ciência da Computação

**“UM ESTUDO SOBRE AS PRINCIPAIS
DIFICULDADES DOS SCRUM MASTERS EM
RELAÇÃO AO USO DE FERRAMENTAS PARA
AUXILIAR SUAS ATIVIDADES”**

Por

ANA CAROLINA ORAN FONSECA E TOLEDO

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, AGOSTO/2010



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ANA CAROLINA ORAN FONSECA E TOLEDO

**“UM ESTUDO SOBRE AS PRINCIPAIS DIFICULDADES DOS SCRUM
MASTERS EM RELAÇÃO AO USO DE FERRAMENTAS PARA AUXILIAR
SUAS ATIVIDADES”**

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO
PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA
COMPUTAÇÃO.*

ORIENTADOR(A): Fernando da Fonseca e Souza, PhD

RECIFE, AGOSTO/2010

Toledo, Ana Carolina Oran Fonseca e

Um estudo sobre as principais dificuldades dos Scrum Masters em relação ao uso de ferramentas para auxiliar suas atividades / Ana Carolina Oran Fonseca e Toledo - Recife: O Autor, 2010.

105 folhas : il., fig., quadros., gráf.

Dissertação (mestrado) Universidade Federal de Pernambuco. Cln. Ciência da computação, 2010.

Inclui bibliografia e apêndice.

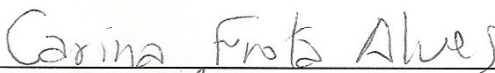
1. Engenharia de software. 2. Gerenciamento de equipes Scrum. 3. Ferramentas de gerenciamento Scrum. I. Título.

005.1

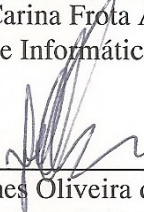
CDD (22. ed.)

MEI2010 – 0173

Dissertação de Mestrado apresentada por **Ana Carolina Oran Fonseca e Toledo** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Um Estudo sobre as Principais Dificuldades dos Scrum Masters em Relação ao Uso de Ferramentas para Auxiliar suas Atividades**", orientada pelo **Prof. Fernando da Fonseca de Souza** e aprovada pela Banca Examinadora formada pelos professores:



Profa. Carina Frota Alves
Centro de Informática / UFPE

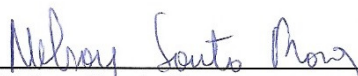


Prof. Jones Oliveira de Albuquerque
Departamento de Estatística e Informática / UFRPE



Prof. Fernando da Fonseca de Souza
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 26 de agosto de 2010.



Prof. Nelson Souto Rosa

Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho aos meus familiares em especial à minha mãe, minha melhor amiga, a qual com seu amor sem dimensão caminhou ao meu lado em todos os momentos difíceis e que merece esta minha vitória mais que eu.

A minha querida irmã Carla, que fez o possível e o impossível para que eu não desistisse deste trabalho, pessoa que me fez acreditar que eu era capaz e a quem serei eternamente grata por mais esta vitória.

Ao meu irmão lindo Carlos Eduardo um presente divino em nossas vidas que me fez rir quando queria chorar,

Minha avó do coração Onezima pelos anos de dedicação e pelas orações.

E por fim, ao meu saudoso Tio Vale que com suas palavras lindas e inteligentes me mostraram que este era o caminho certo a percorrer.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela minha existência.

Ao meu orientador Fernando Fonseca que desde o início do curso deixou de ser um simples professor para se tornar um homem que admiro por sua simplicidade e inteligência. Agradeço por ter acreditado em mim até mesmo nos momentos em que não acreditei e por ter chamado tantas vezes minha atenção para caminhar em frente e não desistir nunca.

Agradeço a minha Tia Lenir que com sua sabedoria e brilhantismo me ajudou nas horas que mais precisei.

Ao meu more, Rayfran, que esteve ao meu lado, me deu ombro muitas vezes, que todo dia conversa comigo para eu não me sentir só, por acreditar em mim e que sempre torceu por mim, pessoa importante em minha vida assim como toda sua família.

Agradeço ao meu amigo irmão de coração Sérgio Roberto e minhas amigas e irmãs Josy, Rose e Carol por todas as palavras carinhosas e de incentivo me dada nos momentos de saudades.

Aos meus amigos de faculdade Heliton, André, Thales e Marxlenin que mesmo não estando em Manaus sempre me acompanharam nesta jornada e acreditaram em mim. Aos meus amigos de mestrado, Aliny, Joseane, Aninha, Catarina, Rebeka e sua família, Daniel, Marcos, Rodrigo, Juliana, Gustavo, Claudia e todos aqueles que estiveram comigo nesta jornada.

Aos meus novos amigos de Recife, Dona Ivone, Ana, Ju, Taci e Mary que me abrigaram em suas vidas e que me deram colo nos momentos de tristeza e riram nos momentos de alegria, jamais vou esquecer vocês.

Ao meu professor de faculdade e amigo Wagner que me incentivou a me inscrever no mestrado e que tantas vezes me deu apoio como pai e amigo quando estava longe da minha família.

A FAPEAM e FUCAPI que acreditaram em mim e me apoiaram financeiramente para a realização deste sonho.

A todos, muito obrigada!

“Para realizar grandes conquistas, devemos não apenas agir, mas também sonhar; não apenas planejar, mas também acreditar.”

Anatole France

RESUMO

Com a mudança das metodologias tradicionais para as metodologias ágeis de gerenciamento de projetos, houve também a evolução dos profissionais que atuam nessa área. Com isso, metodologias como o Scrum passaram a ganhar espaço e quebrar paradigmas nas grandes organizações, promovendo o aparecimento de novos papéis relacionados à gestão de projetos, tais como os Scrum Masters. Estes profissionais, comparados ao gerente de projeto do futuro, têm a responsabilidade não de impor suas ordens, mas sim facilitar o trabalho de pessoas em um projeto. Entretanto, percebeu-se, com os profissionais entrevistados que há uma diferença entre os fundamentos teóricos e a prática quanto às atividades desempenhadas por estes profissionais e a respeito das ferramentas disponíveis para ajudá-los nas suas atividades diárias. Desta forma o objetivo geral desta pesquisa é analisar as principais dificuldades dos Scrum Masters em relação ao uso de ferramentas para auxiliar suas atividades. Para tanto foi realizado um estudo aprofundado com utilização de questionário aplicado a uma amostra de 13 profissionais a respeito das atividades desempenhadas por eles em um projeto que utiliza Scrum, verificando também quais as ferramentas utilizadas pelos mesmo que os auxiliam na execução de suas atividades. Foi verificado ainda se essas ferramentas atendiam às necessidades destes profissionais enquanto Scrum Masters. A partir dos dados coletados, foi possível, entre outros resultados, identificar as reais atividades executadas pelos Scrum Masters no seu dia-a-dia, as quais incluem além daquelas descritas na literatura, outras atividades complementares. Percebeu-se, também que as ferramentas utilizadas por estes profissionais não atendem satisfatoriamente suas necessidades em suas atividades diárias. Assim, foi realizada uma análise comparativa entre os dados coletados nas entrevistas com os Scrum Masters e aqueles obtidos nas revisões bibliográficas que resultou no desenvolvimento de uma ferramenta chamada JustScrum, a qual disponibiliza as funcionalidades listadas nos questionários pelos Scrum Masters, a fim de unir em uma única ferramenta as funcionalidades essenciais desejadas pelos entrevistados para auxiliá-los nas suas atividades profissionais.

Palavras-Chave: Métodos Ágeis. Scrum. Scrum Master.

ABSTRACT

With the change from traditional methodologies into agile project managing methodologies, there was also an evolution of professionals who act on this area. With that, methodologies such as the Scrum have got space and broken paradigms at great organizations, promoting the rising of new roles related to project management, such as the Scrum Masters. These professionals compared to the future project manager, has the responsibility not to impose their commandments, but facilitate the work of people in a project. However, it was noticed, according with the interviewed professionals, that there is a difference between theoretical foundations and practical actions in relation to activities performed by these professionals with regard to the tools available for helping them on their daily activities. Thus the general objective of this research is analyzing the main difficulties of the Scrum Masters in relation to the use of tools to help their activities. So a deep study was done with the use of a questionnaire applied to a sample of 13 professionals with respect to the activities performed by them in a project that uses Scrum, also checking which tools were used to help on performing the activities. It was still checked if these tools attended to the needs of these professionals in the role of Scrum Masters. From the collected data, it was possible, amongst other results, to identify the actual activities performed by the Scrum Masters on their every day job, which include besides those ones described in the literature, other complementary activities. It was also noticed that the tools used by these professionals do not attend satisfactorily their daily activities. Thus, a comparative analysis was done between the data collected on the interviews with the Scrum Masters and those ones gotten on the bibliographic reviews which have resulted on the development of a tool called JustScrum which makes available the essential functionalities listed in the questionnaires by the Scrum Masters in order to join in a unique tool the functionalities wanted by the interviewed people to help them in their professional activities.

Key-words: Agile Methods. Scrum. Scrum Master.

LISTA DE FIGURAS

FIGURA 1: CERTIFICAÇÕES SCRUM.....	25
FIGURA 2: FLUXO SCRUM.....	28
FIGURA 3: FASES DA PESQUISA.....	36
FIGURA 4: QUADRO DE KANBAN	65
FIGURA 5: ESTIMANDO O PRODUCT BACKLOG COM PLANNING POKER	66
FIGURA 6: SPRINT E PROJECT BURN DOWN.....	67
FIGURA 7: ARQUITETURA MVC	77
FIGURA 8: CICLO DE DESENVOLVIMENTO	78
FIGURA 9: EXEMPLO DE USER STORY	79
FIGURA 10: PROJETO DE TELA US_ 2 - O PRODUCT OWNER PODER PRIORIZAR OS ITENS DO PRODUCT BACKLOG	81
FIGURA 11: BANCO DE DADOS US_ 2 - O PRODUCT OWNER PODER PRIORIZAR OS ITENS DO PRODUCT BACKLOG	81
FIGURA 12:TELA IMPLEMENTADA DE SPRINT BACKLOG.....	82
FIGURA 13: TELA IMPLEMENTADA DE PRODUCT BACKLOG	83

LISTA DE QUADROS

QUADRO 1: OS 12 PRINCÍPIOS DOS MÉTODOS ÁGEIS.....	19
QUADRO 2: VANTAGENS E DESVANTAGENS NO USO DE SCRUM.....	23
QUADRO 3: RESUMO DAS RESPONSABILIDADES DOS SCRUM MASTERS.....	34
QUADRO 4: FUNCIONALIDADES.....	62
QUADRO 5: ANÁLISE COMPARATIVA	75
QUADRO 6: PRIORIZAÇÃO DAS USER STORIES	80
QUADRO 7: ASSOCIAÇÃO ENTRE FUNCIONALIDADES E USER STORES	84
QUADRO 8: ANÁLISE COMPARATIVA DAS FERRAMENTAS	88

LISTA DE GRÁFICOS

GRÁFICO 1: UTILIZAÇÃO DO SCRUM	20
GRÁFICO 2: OBSTÁCULOS NA ADOÇÃO DE METODOLOGIAS ÁGEIS.	24
GRÁFICO 3: CERTIFICAÇÃO	45
GRÁFICO 4: TEMPO DE TRABALHO COM A METODOLOGIA DE GERENCIAMENTO SCRUM.....	47
GRÁFICO 5: TEMPO COMO SCRUM MASTER.....	48
GRÁFICO 6: ATUAÇÃO EM OUTRA FUNÇÃO ALÉM DE SCRUM MASTER.....	49
GRÁFICO 7: QUAIS FUNÇÕES QUE ATUA ALÉM DE SCRUM MASTER.....	50
GRÁFICO 8: ATIVIDADES DESEMPENHADAS NOS PROJETOS	51
GRÁFICO 9: FERRAMENTAS NÃO-COMPUTACIONAIS UTILIZADAS	54
GRÁFICO 10: FERRAMENTAS NÃO-COMPUTACIONAIS QUE PODERIAM SER DESCARTADAS	55
GRÁFICO 11: FERRAMENTAS COMPUTACIONAIS UTILIZADAS	57
GRÁFICO 12: FERRAMENTA COMPUTACIONAL QUE PODERIA SER DESCARTADA	59
GRÁFICO 13: FERRAMENTAS DEDICADAS A GESTÃO ÁGIL	69

LISTA DE ABREVIATURAS/ACRÔNIMOS

SA	Scrum Alliance
CSM	Certified ScrumMaster
CSPO	Certified Scrum Product Owner
CSD	Developer Certified Scrum
CSP	Certified Scrum Practitioner
CST	Certified Scrum Trainer
CSC	Certified Scrum Coach
REP	Registered Education Provider
PO	Product Owner
SM	Scrum Master
PB	Product backlog
PBIs	Itens do Product Backlog

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 MOTIVAÇÃO	13
1.2 DESCRIÇÃO DO PROBLEMA	14
1.3 OBJETIVOS	14
1.3.1 Objetivo geral	14
1.3.2 Objetivos específicos.....	14
1.4 RESULTADO DO PROJETO.....	15
1.5 ESTRUTURA DO TRABALHO.....	15
 2 REFERENCIAL TEÓRICO	16
2.1 METODOLOGIAS ÁGEIS	16
2.2 SCRUM	21
2.2.1 Certificação.....	25
2.2.2 Os papéis.....	27
2.2.3 O processo	28
2.3 SCRUM MASTER.....	31
 3 METODOLOGIA	35
3.1 MÉTODOS CIENTÍFICOS	35
3.2 DETALHAMENTO DA PESQUISA	36
3.3 CLASSIFICAÇÃO DA PESQUISA	38
3.3.1 Pesquisa descritiva e exploratória	38
3.3.2 Pesquisa de campo	39
3.3.3 Revisão bibliográfica.....	39
3.3.4 Análise quantitativa e qualitativa	40
3.4 ELABORAÇÃO DE INSTRUMENTOS DE COLETA DE DADOS	41
3.5 DEFINIÇÃO DA POPULAÇÃO PARA COLETA DE DADOS	42
3.6 COLETA, TABULAÇÃO, ANÁLISE E INTERPRETAÇÃO DOS DADOS	43
 4 RESULTADOS DA PESQUISA.....	44
4.1 REPRESENTAÇÃO E ANÁLISE DOS DADOS	45
4.1.1 Parte I - Identificação do sujeito da pesquisa	45
4.1.2 Parte II - Identificação das atividades desempenhadas pelos Scrum Masters.....	51
4.1.3 Parte III - Identificação das ferramentas e funcionalidades utilizadas pelos Scrum Masters	53
4.1.4 Parte IV - Identificação das funcionalidades de uma ferramenta computacional para auxiliar os Scrum Masters	61
4.2 ANÁLISE COMPARATIVA DAS FERRAMENTAS	63
4.2.1 Ferramentas não computacionais	63
4.2.1.1 Quadro de kanban.....	64
4.2.1.2 Planning poker	65
4.2.1.3 Burn down chart.....	66
4.2.2 Comparação entre as ferramentas não computacionais.....	68
4.2.3 Ferramentas computacionais.....	68
4.2.3.1 Scrumpy.....	70
4.2.3.2 Clockingit	70
4.2.3.3 Planningpoker	71
4.2.3.4 Scrumworks	71

4.2.3.5 Firescrum	72
4.2.3.6 Targetprocess	72
4.2.3.7 Rally	73
4.2.3.8 Versionone	73
4.2.4 Comparação entre as ferramentas computacionais	74
5 A FERRAMENTA JUSTSCRUM.....	76
6 CONSIDERAÇÕES FINAIS.....	86
REFERÊNCIAS.....	90
APÊNDICE A – QUESTIONÁRIO DA PESQUISA.....	96
APÊNDICE B – QUESTIONÁRIO DE SATISFAÇÃO	103
APÊNDICE C – INSTRUÇÃO DE CONFIGURAÇÃO DA FERRAMENTA JUSTSCRUM.....	104

1 INTRODUÇÃO

Atualmente, tem-se verificado que a evolução cultural e a dos negócios se dão de maneira rápida, requerendo que as organizações superem os desafios provenientes desta evolução de forma criativa, com a finalidade de competir num mercado cada vez mais exigente.

A constante pressão por inovações, a concorrência acirrada, a necessidade de responder rapidamente às expectativas dos clientes, e de tentar solucionar problemas, as empresas passaram a vivenciar um momento de mudanças: a transição das metodologias tradicionais para a uma metodologia que correspondesse às necessidades que esse momento requeria, as chamadas metodologias ágeis. Esse tipo de metodologia é bastante utilizada em projetos de desenvolvimento de software, pois responde rapidamente às constantes mudanças que esses projetos exigem.

Dentre os vários métodos que compõem a metodologia ágil encontra-se o Scrum. Sendo definido como uma metodologia flexível, que tem por objetivo definir um processo de desenvolvimento iterativo e incremental, podendo ser aplicado a qualquer produto ou no gerenciamento de qualquer atividade complexa. (SCHWABER, 2004).

O *Scrum* estabelece o papel do *Scrum Master*, o qual é uma espécie de maestro de um projeto *Scrum* e responsável por remover as barreiras entre o *Team* e o *Product Owner*, melhorar o dia-a-dia dos integrantes do *Team* combatendo os impedimentos surgidos e, em especial, garantir que o *Scrum* seja seguido pelo *Team*.

1.1 Motivação

Em função de tantas responsabilidades que são atribuídas a esses profissionais, é fundamental que os mesmos estejam amparados por ferramentas capazes de auxiliá-los e de fornecerem as informações das quais necessitam o mais rápido possível.

Neste sentido, destaca-se a importância de um estudo capaz de identificar as ferramentas existentes, verificando se estas atendem ou não as reais necessidades e expectativas destes profissionais. A partir das limitações encontradas, é importante também considerar o desenvolvimento de uma ferramenta mais aderente às necessidades dos mesmos.

1.2 Descrição do problema

Este trabalho tem como propósito analisar as atividades dos *Scrum Masters* e as ferramentas de apoio às suas atividades profissionais, avaliando as limitações existentes nas mesmas e considerando o desenvolvimento de uma ferramenta baseada em software livre como proposta para eliminação das limitações encontradas durante as entrevistas realizadas com *Scrum Masters*, como parte deste trabalho. Para tanto, a pesquisa faz os seguintes questionamentos: As ferramentas existentes atendem às reais expectativas e necessidades dos *Scrum Masters*? Como resolver as limitações encontradas?

1.3 Objetivos

Os objetivos da pesquisa foram divididos em geral e específicos. O objetivo geral direciona a intenção global da investigação. E os objetivos específicos limitam as metas pretendidas.

Assim, os mesmos foram organizados em:

1.3.1 Objetivo geral

Analisar as principais dificuldades dos *Scrum Masters* em relação ao uso de ferramentas para auxiliar suas atividades.

1.3.2 Objetivos específicos

- Realizar um levantamento teórico sobre a metodologia *SCRUM*, em especial a respeito das responsabilidades e atividades do *Scrum Master*;
- Investigar, por meio de questionários, as atividades desempenhadas pelos *Scrum Masters*, as principais ferramentas utilizadas por eles para auxiliar na execução de suas atividades, verificar se estas atendem as suas necessidades, e em caso contrário, identificar um conjunto mínimo de funcionalidades que estes profissionais esperam de uma ferramenta;
- Identificar as vantagens e desvantagens das ferramentas existentes que têm o propósito de auxiliar os *Scrum Masters* na execução de suas responsabilidades;
- Realizar uma análise comparativa entre as ferramentas pesquisadas e os resultados das entrevistas com os *Scrum Masters*;

- Desenvolver uma ferramenta capaz de auxiliar os *Scrum Masters* na realização de suas atividades, baseando-se na análise comparativa; e
- Realizar testes da ferramenta desenvolvida com *Scrum Masters* de modo a avaliar a referida ferramenta e promover melhorias em função do resultado dos testes.

1.4 Resultado do Projeto

Como resultado do projeto, pretende-se apresentar a análise comparativa realizada com os dados coletados nas entrevistas com os *Scrum Masters* e aqueles obtidos no estudo das ferramentas desenvolvidas para apoiar as atividades dos mesmos. O intuito da apresentação de tal análise é responder ao questionamento se as reais expectativas dos *Scrum Masters* entrevistados são atendidas ou não por essas ferramentas. Para tratar as limitações encontradas, tanto sob o ponto de vista de tais profissionais quanto os levantados pela análise da pesquisadora, será apresentada também uma ferramenta desenvolvida nesse trabalho, a qual se propõe ser mais aderente às necessidades dos *Scrum Masters* que participaram da pesquisa.

1.5 Estrutura do trabalho

Este trabalho foi estruturado em capítulos, conforme descrição a seguir:

Capítulo 1 descreve inicialmente o problema, a motivação e justificativas, além do objetivo geral e objetivos específicos do mesmo, o resultado esperado pela pesquisa e a estrutura da dissertação.

O Capítulo 2 corresponde ao referencial teórico. Este capítulo aborda os principais conceitos que envolvem a metodologia *Scrum* de desenvolvimento de software e ferramentas que auxiliam o *Scrum Master* em suas atividades diárias em projetos que utilizam a metodologia ágil *Scrum*.

O Capítulo 3 contempla a metodologia utilizada, explicando as fases e atividades que a compõem, assim como sua classificação.

O Capítulo 4 apresenta os resultados da pesquisa com a representação e análise dos dados e análise comparativa das ferramentas.

O Capítulo 5 apresenta a ferramenta desenvolvida como parte do resultado da pesquisa e, por fim, o Capítulo 6 apresenta as considerações finais sobre a pesquisa realizada, destacando os resultados obtidos e trazendo sugestões de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo reúne as principais teorias e conceitos relacionados ao tema desta pesquisa. E está dividido em três (03) seções. Na seção 2.1 são tratados os fundamentos teóricos sobre as metodologias ágeis, na seção 2.2 os conceitos sobre a metodologia *Scrum* e na seção 2.3 são destacadas referências mais específicas a respeito do papel *Scrum Master*.

2.1 Metodologias ágeis

Vargas (2005) afirma que ao longo do tempo, a sociedade evoluiu tanto em sua cultura como em seus negócios. O mercado, cada vez mais competitivo, exige das organizações a superação de desafios, a capacidade de fazer mais com menos, administrando bem suas atividades, para que se tornem diferentes de seus concorrentes e alcancem um lugar de destaque.

Nestes termos, Sommerville (2003) afirma que devido aos avanços tecnológicos e à necessidade de desenvolvimento de software com qualidade, as empresas do ramo de tecnologia têm intensificado a utilização de metodologias de desenvolvimento, como forma de aumentar as chances de sucesso de seus projetos.

Assim, surgiu uma série de metodologias e processos focados no planejamento dos produtos e fortemente voltados para a documentação para permitir melhor controle. Porém, observou-se que as metodologias criadas não trouxeram exatamente os benefícios esperados, segundo Mariz (2009), pois com elas observou-se um descontentamento generalizado, em vista que a maioria dos projetos de software apresentava problemas como: atraso no cronograma; estouro de orçamento e funcionalidades que não atendiam às necessidades reais dos clientes, gerando assim sistemas de alto custo e baixa qualidade, além de insatisfação dos clientes.

A preocupação constante dos gerentes de projetos com esses problemas, fez com que estudiosos da área de engenharia de software observassem que as metodologias tradicionais não estavam mais atendendo de forma satisfatória seus objetivos. Passaram então a enxergar a necessidade de criação de práticas que estivessem de acordo e que respondessem à dinâmica dos dias atuais. Assim, surgiram novos padrões de produção denominados metodologias ágeis.

Segundo Highsmith (2004), o termo agilidade é: "habilidade de criar e responder a mudanças com respeito ao resultado financeiro do projeto em um turbulento ambiente de negócios. É a habilidade de balancear flexibilidade com estabilidade".

Esse conceito pode induzir o desenvolvedor a supor que agilidade corresponde à falta de estrutura ou organização. Entretanto esse termo está diretamente relacionado à capacidade de adaptação a várias situações, sendo necessária para isso, a adoção de métodos a serem seguidos.

De forma complementar, Highsmith (2004) ressalta que a ausência de uma estrutura ou de estabilidade pode levar ao caos, no entanto uma estrutura excessiva gera rigidez.

Em vista disso, segundo Mariz (2009), as metodologias ágeis propõem um meio termo entre "nenhum processo" e "processo demais", tornando-se assim uma ferramenta de auxílio e não um obstáculo ao desenvolvimento de software.

Soares (2004) explica que as metodologias tradicionais devem ser aplicadas somente em situações em que os requisitos do projeto são estáveis e previsíveis. Enquanto que para projetos em que existam muitas mudanças, nos quais refazer parte do código é uma atividade aceitável e o desenvolvimento rápido é fundamental, é mais adequada a utilização de uma metodologia ágil.

Vale destacar que grande parte dos projetos de software enquadra-se de forma bastante aderente a esse segundo tipo de projeto descrito anteriormente, nos quais o dinamismo apresenta-se como um risco com o qual a equipe deve conviver, conforme afirma Sommerville (2003): "Durante o processo de software, a compreensão dos desenvolvedores sobre o problema está constantemente se modificando, e essas mudanças se refletem nos requisitos."

Complementando o pensamento, Rodrigues e Rost (2007) elucidam que as metodologias ágeis surgiram com uma proposta de "desburocratizar" o processo de desenvolvimento de software, permitindo que as equipes sejam mais adaptáveis, respondendo rapidamente às constantes mudanças nos projetos de software.

Segundo Mountain (*apud* Cavalcanti, 2009), com o objetivo principal de desburocratizar o processo, as metodologias ágeis não criam uma sobrecarga de trabalho nas informações necessárias para o acompanhamento e aumentam a colaboração e visibilidade do time do projeto. O autor cita que as metodologias ágeis sugerem a utilização de artefatos simples, muitas vezes construídos por meio de planilhas e quadro de tarefas baseados em cartões, murais, cartolinas e pôsteres.

Os métodos ágeis passaram a ser mais conhecidos a partir de 2001, quando um grupo de 17 (dezessete) especialistas em processos de desenvolvimento de software reuniu-se para estabelecer e identificar um padrão comum para métodos previamente existentes, tais como: *eXtreme Programming* (XP) [<http://www.extremeprogramming.org/>], *Scrum* [www.scrumalliance.org/], *Dynamic System Development Method* (DSDM) [<http://www.dsdm.org/>], *Adaptive Software Development* (ASD) [<http://www.adaptivesd.com/>], *Crystal* [<http://alistair.cockburn.us/>], *Feature Driven Development* (FDD) [www.featuredrivendevelopment.com/] e *Lean Development* [<http://www.poppendieck.com/>]. O resultado foi a criação do chamado “manifesto ágil” o qual, segundo Beck *et al.* (2001), estabelece um conjunto de valores, descritos a seguir, os quais devem ser adotados nos projetos ágeis:

- Indivíduos e interações são mais importantes que processos e ferramentas;
- Software funcionando é mais importante do que documentação completa e detalhada;
- Colaboração com o cliente é mais importante do que negociação de contratos; e
- Responder a mudanças é mais importante do que seguir um plano inicial.

O “Manifesto Ágil” não rejeita os processos, ferramentas, documentação, negociação de contratos ou o planejamento, mas simplesmente mostra que eles têm importância secundária quando comparado com os indivíduos e interações, com o software estar executável, com a colaboração do cliente e as respostas rápidas a mudanças. .

Neste sentido, Oliveira *et al* (2003) dizem que devido à sua natureza adaptativa e pelo seu foco em resultados, as metodologias ágeis atendem melhor às necessidades de cada uma das partes envolvidas no processo de construção de software, pois com essas: os usuários têm a oportunidade de obter um produto mais próximo de suas necessidades; o cliente tem um retorno mais rápido do seu investimento; os desenvolvedores têm a oportunidade de trabalhar em um ambiente melhor e o fornecedor se beneficia com o trabalho de equipes mais eficientes e produtivas.

O uso da agilidade, de acordo com Pereira *et al* (2007), proporciona vantagens como:

- Cria um ambiente propício para definição de mudanças de requisitos e inovação durante o desenvolvimento do produto, além de mais colaborativo e produtivo entre desenvolvedores e cliente, resultando em entregas mais rápidas de produto, melhor adaptados à realidade do cliente e com a qualidade esperada;

- Facilita o gerenciamento do projeto, uma vez que existe maior integração e comprometimento da equipe do projeto, que, conseqüentemente, se sente mais motivada e valorizada; e
- Reforça o planejamento constante do projeto, minimizando os riscos e considerando que o planejamento é mais importante do que o plano. Não se deve parar de planejar até que se tenha encontrado a satisfação do cliente com a entrega do produto final; e
- Valoriza a satisfação do cliente em primeiro lugar.

Além de valores e vantagens já mencionadas, o Manifesto Ágil destaca doze (12) princípios às práticas de métodos ágeis de desenvolvimento de software, que são descritos no Quadro 1.

Quadro 1: Os 12 princípios dos métodos ágeis

1. Nossa prioridade é satisfazer o cliente através da entrega rápida e contínua de um software de valor;
2. Receber bem as mudanças de requisitos, mesmo que em uma etapa avançada do desenvolvimento;
3. Entregue novas versões de software frequentemente;
4. Pessoas de negócio e programadores devem trabalhar juntos, diariamente, ao longo de todo projeto;
5. Construa projetos com pessoas motivadas. Ofereça a elas o ambiente e todo o apoio necessário e acredite em sua capacidade de realização do trabalho;
6. O método mais eficiente e efetivo de distribuir a informação para e entre uma equipe de desenvolvimento é a comunicação face à face;
7. O software funcionando é a medida primária do progresso do projeto;
8. Os processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
9. A atenção contínua na excelência técnica e num bom projeto acelera a agilidade;
10. A simplicidade é essencial;
11. As melhores arquiteturas, requisitos e projetos emergem de equipes autogerenciadas; e
12. Equipes de projeto avaliam seu desempenho em intervalos regulares e ajustam seu comportamento de acordo com os resultados.

Fonte Adaptada: Beck et al. (2001)

Em vista disso, é notório que as metodologias ágeis dão ênfase às pessoas e à simplicidade, características que vão de encontro com o que as metodologias tradicionais pregam.

Neste trabalho, será adotada a metodologia *Scrum* como foco da pesquisa, pois segundo autores como Mariz (2009), Silveira (2007) e Santos (2008) o *Scrum*, dentre os métodos ágeis é o que vem ganhando cada vez mais destaque com seus princípios, papéis e processos voltados para o gerenciamento ágil de projetos de software.

Essa afirmação pode ser comprovada por uma pesquisa feita pela VersionOne (2008), onde se obteve 2.319 respondentes distribuídos em 80 países a respeito da adoção global de práticas ágeis. O Gráfico 01 ilustra que dentre as empresas entrevistadas que utilizavam métodos ágeis, 71% usavam *Scrum*, das quais 49% utilizavam *Scrum* puro e 22% empregavam-no associado à outra metodologia, como o XP.

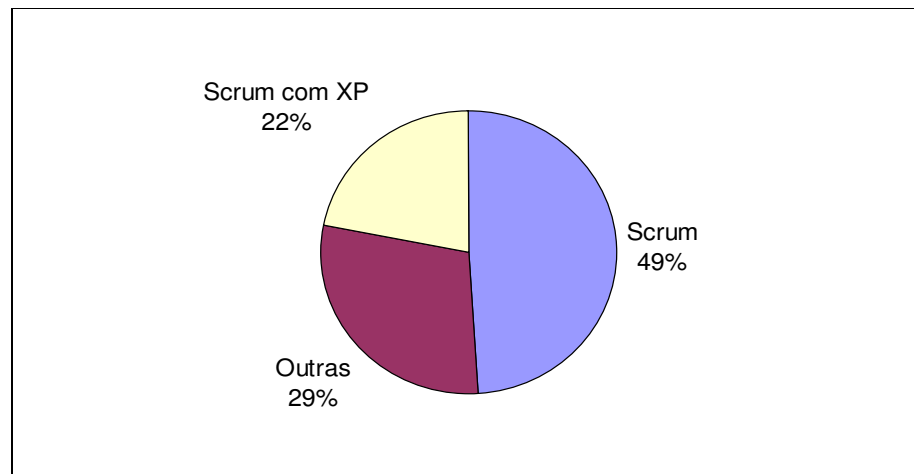


Gráfico 1: Utilização do Scrum
Fonte Adaptada: VersionOne (2008)

Com sucesso comprovado, o *Scrum* é utilizado por grandes empresas, como: a IBM [<http://www.ibm.com>], HP [www.hp.com], Intel [www.intel.com], Siemens [www.siemens.com], amazon.com [www.amazon.com], AMD [www.amd.com], Motorola [www.motorola.com], Philips [www.philips.com], Nokia [www.nokia.com], Microsoft [www.microsoft.com], Yahoo [www.yahoo.com], Google [www.google.com], BBC [<http://www.bbc.co.uk/>], Time Warner [www.timewarner.com], Xerox [www.xerox.com] e Oracle [www.oracle.com], que fazem uso desta metodologia para gerenciar seus projetos com êxito. No Brasil muitas empresas também já utilizam o *Scrum*, tais como: C.E.S.A.R [www.cesar.org.br], Ci&T [www.cit.com.br], Dito Idéias & Soluções [www.dito.com.br], FEMAHOSP [<http://www.femahosp.com.br>], IEA - Instituto de Estudos Avançados [www.iea.com.br], dentre muitas outras segundo Ferreira (2009) e *Scrum Recife* [s.d.]. A metodologia *Scrum* será detalhada na próxima seção.

2.2 SCRUM

Scrum, segundo Rodrigues e Rost (2007) é uma metodologia ágil para gerenciamento de projetos, que permite responder rapidamente às mudanças, mesmo que estas sejam tardias, reduzindo o impacto das mesmas nos projetos. Favorece ainda à satisfação do cliente, uma vez que este participa ativamente das decisões, além de haver constantemente liberação de funcionalidades 100% desenvolvidas.

Foi descrito inicialmente por Takeuchi e Nonaka (1986) onde os autores defendiam a idéia de que projetos usando equipes pequenas e multidisciplinares, conhecidas como *cross-functional*, geravam resultados superiores e relacionaram esses times de grande desempenho a uma jogada de *Rugby*, chamada *Scrum*. Nela, todos os integrantes do grupo devem agir em conjunto para mover a bola ao longo do campo. Assim, o time mantém um foco em comum e cada um, desempenhando seu papel bem definido, coopera para atingir um objetivo, ou seja, o trabalho em equipe determina o time vencedor.

Apesar de ser muito utilizado no desenvolvimento de software, Rodrigues e Rost (2007) afirmam que o *Scrum* foi originalmente concebido para gerenciamento de projetos de fabricação de automóveis e produtos de consumo. E que sua popularização no desenvolvimento de software ocorreu apenas em 1995, com a formalização de sua descrição, feita por Schwaber (1995).

Para Magno (2009), *Scrum* é um processo iterativo e incremental, que pode ser empregado para o desenvolvimento de qualquer produto e gerenciamento de qualquer projeto. É muito mais um framework do que uma metodologia, muito mais atitude que um processo.

Ferreira (2009), em concordância com Magno (2009), também define o *Scrum* como um framework que propõe um método simples para gerenciar projetos complexos. Ferreira (2009) esclarece que os métodos convencionais direcionam o foco para o desenvolvimento de regras, enquanto que o *Scrum* visa o desenvolvimento prioritário de funcionalidades que são mais importantes para o negócio do cliente.

Na visão de Rodrigues e Rost (2007), *Scrum* é uma metodologia de desenvolvimento ágil, focada no trabalho em equipe, com equipes autogerenciadas e participação ativa do cliente.

Para Villas Bôas (2005), *Scrum* é uma abordagem empírica baseada na flexibilidade, adaptabilidade e produtividade, em que a escolha das técnicas de desenvolvimento fica a cargo dos programadores. Enquanto que para Asfora (2009), o *Scrum* se destaca das demais metodologias ágeis por dar mais enfoque à área de gerenciamento de projetos.

Com isso, desde que começou a ser adotada, essa metodologia vem atingindo grande sucesso tanto entre os desenvolvedores como entre os gerentes, pois, conforme Schwaber e Beedle (2001), o *Scrum* reúne atividades de monitoramento e *feedback*, das quais toda a equipe participa, visando a identificação e a correção de qualquer deficiência e/ou impedimento no processo de desenvolvimento de um projeto *Scrum* continuamente.

Com todas essas características, o *Scrum* permite um melhor acompanhamento do que está acontecendo durante o projeto, facilitando o ajuste durante o desenvolvimento do referido projeto e possibilitando alcançar seus objetivos de forma ágil.

Abrahamsson (*apud* Gomes, 2007) explica que o *Scrum*, fornece um processo para o desenvolvimento do software, concentrando em como a equipe deve trabalhar em um projeto em constante mudança.

Em vista disso, *Scrum* é uma metodologia ideal para construir software, pois o ambiente em que esses projetos são desenvolvidos é complexo e os requisitos muitas vezes não são claros ou mudam com bastante frequência.

Entretanto, é importante ressaltar que o *Scrum* não vai descrever exatamente o que deve ser feito, como também não se propõe a resolver todos os problemas, porém, em função do monitoramento contínuo realizado por todo o *Team*, os problemas serão mais facilmente identificados.

Essa facilidade para lidar e remover problemas vem, especialmente, de algumas características intrínsecas do *Scrum*, as quais são sintetizadas por Ferreira (2009) como:

- Permite o gerenciamento de mudanças de forma muito eficiente, além de estreitar a comunicação entre o cliente e a equipe que está desenvolvendo o projeto, mantendo sempre as funcionalidades revisadas e corrigindo possíveis falhas rapidamente;
- Possibilita a construção do software em iterações. Dessa forma, o cliente sempre vê o resultado final de cada iteração e define se está de acordo com suas expectativas correto ou não, apontando os erros e evoluindo gradativamente o projeto; e
- Transfere a responsabilidade do gerente para toda a equipe do projeto.

Além das características intrínsecas do *Scrum*, existem também os seus princípios básicos, citados por Schwaber (2004), os quais podem ser sumarizados da seguinte forma:

- Equipes pequenas de trabalho, buscando a maximização da comunicação e da troca de conhecimento tácito e informal e a minimização de overhead;

- Adaptação às solicitações de mudanças técnicas ou de cliente/usuário, assegurando a entrega do melhor software possível;
- Entregas frequentes de versões que podem ser testadas, ajustadas, executadas, documentadas e liberadas para produção;
- Divisão do trabalho e das responsabilidades da equipe de projeto em pequenas entregas; e
- Habilidade em entregar um software pronto, quando da necessidade do cliente ou do negócio.

Segundo Norton (*apud*, Mariz, 2009), existem algumas vantagens e desvantagens no uso de *Scrum*, como as apresentadas no Quadro 2.

Quadro 2: Vantagens e Desvantagens no uso de Scrum

Vantagens	Desvantagens
• Aumento de produtividade; • <i>Scrum</i> permite desenvolvimento <i>bottom-up</i> rápido e contínuo; • O produto torna-se uma série de partes gerenciáveis; • Ocorre progresso no sistema, mesmo quando os requisitos não estão estáveis; • Tudo é visível para todos; • Aumenta a comunicação do time; • O time compartilha sucesso ao longo do caminho até o fim; • Clientes recebem na data programada as entregas dos produtos; • Clientes recebem informações frequentes de como está o desenvolvimento do produto; • É desenvolvida uma forte relação com o cliente, constrói-se confiança e o conhecimento cresce; • É criada a cultura onde todos esperam que o projeto seja bem sucedido.	• A gerência deve estar disposta a realizar mudanças para ajudar o time <i>Scrum</i> a ser bem sucedido; • Requer constante monitoramento, tanto qualitativo quanto quantitativo; • Requer gerenciamento para delegar a autoridade de tomada de decisão para o time; • Gerentes devem levar o time a tomar suas próprias decisões, até permitindo que eles falhem se necessário; • <i>Scrum</i> é novo e diferente e, frequentemente, as pessoas são resistentes a mudanças; • Alguns profissionais não se sentem confortáveis com a responsabilidade.

Fonte: Mariz (2009)

Analisando o Quadro 02, é possível observar que as vantagens de se utilizar o *Scrum* superam os argumentos de desvantagem desta metodologia. Além disso, pode-se considerar que as principais desvantagens relativas à aplicação do *Scrum* estão relacionadas às dificuldades das pessoas em lidar com mudanças e inovações.

Para VersionOne (2008) as principais dificuldades em se adotar o *Scrum* em uma organização são a capacidade de mudança da cultura organizacional e a resistência das pessoas quanto à mudança, conforme detalha o Gráfico 2.

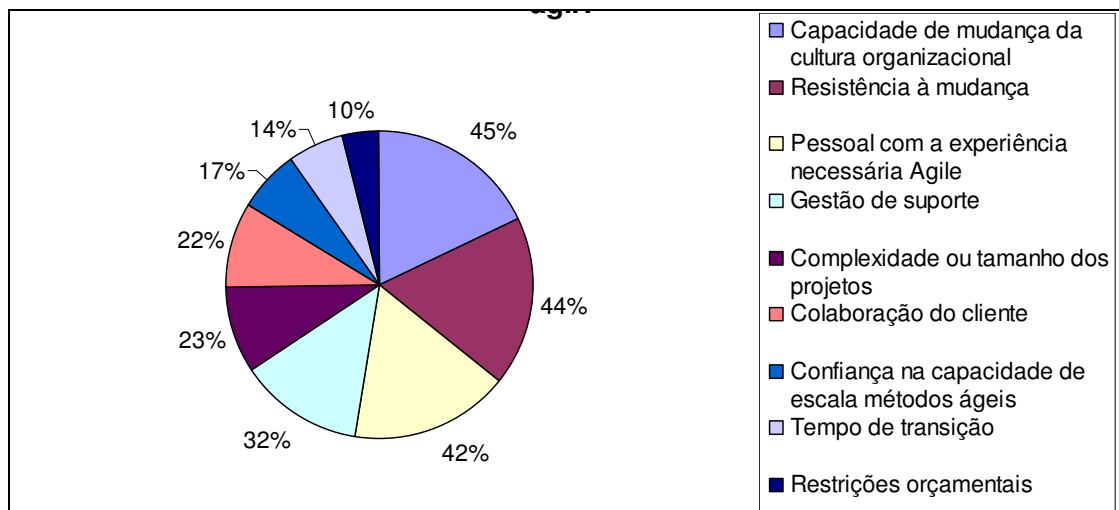


Gráfico 2: Obstáculos na adoção de metodologias ágeis.

Fonte Adaptada: VersionOne (2008)

Isso pode ser confirmado quando Mariz (2009), afirma que o *Scrum* promove uma mudança estrutural na forma como os times de desenvolvimento se organizam e cultural na forma de relação entre os envolvidos no projeto. A responsabilidade pelo desenvolvimento do software passa a ser compartilhada pela equipe e não se concentra apenas no gerente de projetos, o qual passa a ser mais um mediador e facilitador desse processo, o que é o oposto ao que as metodologias tradicionais aplicam.

VersionOne (2008) afirma que outro grande obstáculo na adoção das metodologias ágeis é não ter pessoas com experiência no assunto. Por tanto, em virtude dessa enorme mudança de cultura em gerenciar projeto, seria importante antes de iniciar um projeto *Scrum*, que pelo menos um dos membros da equipe tenha obtido alguma certificação *Scrum*, principalmente a de *Certified ScrumMaster* que se trata da certificação introdutória ao universo *Scrum*. As certificações definidas pela *Scrum Alliance* [www.scrumalliance.org] para o *Scrum* serão detalhadas na próxima seção.

2.2.1 Certificação

Com a crescente evolução das metodologias tradicionais para as ágeis, houve também a procura pela certificação dessas metodologias. A certificação serve para atestar que o profissional tem conhecimento sobre o assunto que é certificado e esteja apto a trabalhar com a metodologia, porém muito mais importante do que uma certificação é a mudança de hábito do profissional para trabalhar bem com essa nova metodologia.

O programa de certificação *Scrum Alliance* (SA) se divide em seis (6) níveis de certificados: *Certified ScrumMaster* (CSM), *Certified Scrum Product Owner* (CSPO), o mais novo *Developer Certified Scrum* (CSD), *Certified Scrum Practitioner* (CSP), *Certified Scrum Trainer* (CST) e por fim *Certified Scrum Coach* (CSC), conforme apresentado na Figura 1. Existe ainda *Registered Education Provider* (REP) que não é uma certificação, mas também será abordado neste trabalho.

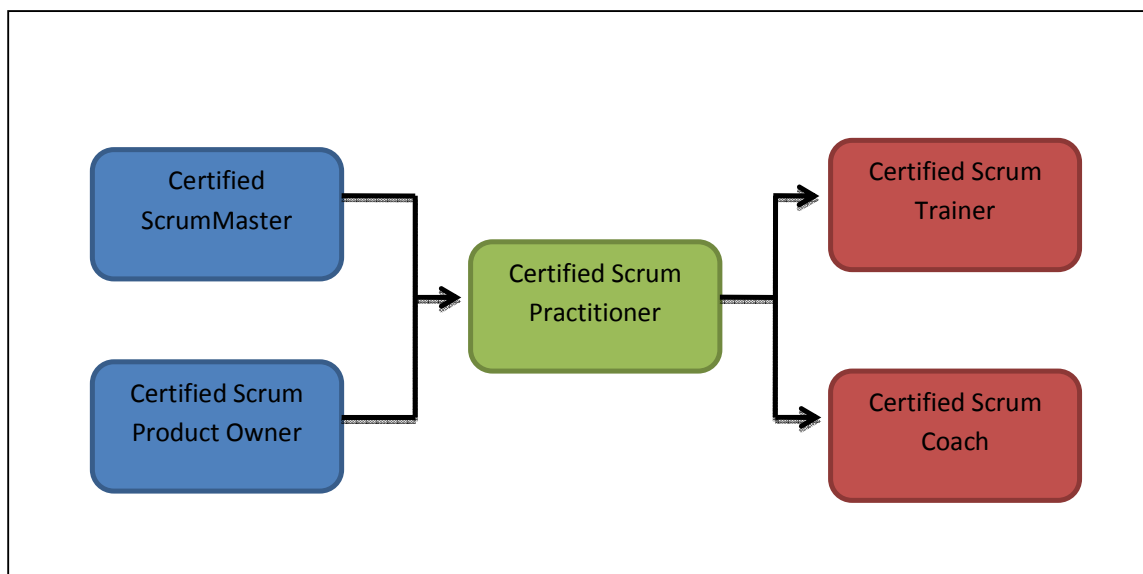


Figura 1: Certificações SCRUM

Fonte: Magno (2009)

Conforme *Scrum Alliance*, o primeiro nível de certificação é composto por CSM e CSPO. O profissional que procura ter essas duas (2) certificações deve passar por um treinamento aprovado pela *Scrum Alliance* de 16 horas cada. Este treinamento atesta que o participante possui conhecimento nesses papéis e é capaz de aplicar o *Scrum*, porém não comprova experiência ou habilidade para essa tarefa. A partir de 1 de outubro de 2009 para se obter a certificação CSM é necessário fazer uma avaliação com 25 perguntas sobre

terminologia, práticas e princípios de *Scrum*, porém essa avaliação não reprova nenhum participante.

A certificação CSD é a mais nova certificação da *Scrum Alliance*. Ela visa credenciar desenvolvedores aptos nas práticas ágeis de engenharia de software e nos princípios *Scrum*. Para obter o CSD o profissional precisa participar de 5 (cinco) dias de treinamento com os seguintes critérios:

- No mínimo 03 dias de treinamento em desenvolvimento de software, em um ou mais cursos desejáveis para o CSD. Esses cursos serão ministrados por entidades (*Scrum Alliance - Registered Education Provider*) certificadas pela *Scrum Alliance*;
- No mínimo 01 dia de treinamento em *Scrum*, em um curso desejável para o CSD. O CSM é um desses cursos; e
- O quinto dia é eletivo, ou seja, o profissional pode escolher dentre os cursos desejáveis para desenvolver as habilidades que sentir trazer mais benefícios a seu papel.

Para o certificado CSP, o profissional deve demonstrar que tem pelo menos um ano de experiência real usando *Scrum* em um projeto, precisa mostrar que sabe aplicar os conceitos, práticas e princípios do *Scrum* e saber como e por que *Scrum* funciona. Para demonstrar isto, Roman (2009) afirma que é preciso responder a uma série de questões sobre a utilização de *Scrum* em um projeto em que participou. As respostas são avaliadas por uma comissão da *Scrum Alliance*, que pode fazer novos questionamentos para esclarecer algo que não esteja claro. Esta comissão verifica se no projeto avaliado os conceitos de *Scrum* foram seguidos corretamente, conferindo (ou não) o título de *Certified Scrum Practitioner*. Nesse nível, a certificação também atesta a experiência e habilidade do profissional na implantação e utilização da metodologia *Scrum*, e o profissional deve ter pelo menos CSM, CSPO ou CSD.

As certificações CST e CSC são para profissionais que querem ensinar *Scrum* e treinar novos CSM e CSPO. A CST certifica que o profissional pretende lecionar na certificação CSM ou CSPO. A CSC certifica que o profissional pretende atuar como treinador (*coach*) da metodologia. O candidato a esta certificação passa por uma avaliação, onde uma comissão pode aprová-lo como *Certified Scrum Trainer* ou *Certified Scrum Coach*. Nesse nível, a certificação atesta que a pessoa é capaz de transmitir conhecimento para outras pessoas que estejam interessadas em aprender *Scrum*.

REP (*Registered Education Provider*) é uma espécie de filiação fornecida às organizações, a qual oferece a capacidade das organizações de cadastrar cursos de formação

que irão beneficiar a comunidade de *Scrum*. Com o REP, as organizações de formação, seus instrutores e cursos serão examinados pela *Scrum Alliance* constantemente e seu desempenho será monitorado.

Mesmo sendo desejável que antes de começar a trabalhar e a introduzir o *Scrum* em um projeto ou organização seria bom ter algum profissional com a certificação na equipe, é perfeitamente aceitável e cabível um grupo de pessoas iniciar seus projetos no *Scrum* sem essas certificações. Para isto, Schwaber (2004) propõe que para o desenvolvimento de um projeto *Scrum* é necessária a formação de equipes de desenvolvimento compostas por três papéis principais: o *Scrum Master*, o *Product Owner* e o *Team*, a ser descritos na próxima seção.

2.2.2 Os papéis

Em um projeto *Scrum*, os profissionais assumem responsabilidades bem específicas. Na visão de Schwaber (2004), essas responsabilidades são divididas em três papéis, os quais são descritos a seguir:

- *Product Owner* (PO) - este papel é responsável por definir a visão do produto, gerenciar o retorno de investimento (ROI), apresentar ao *team* os requisitos necessários para a entrega do produto, priorizar cada requisito de acordo com seu valor para o negócio/cliente, gerenciar a entrada de novos requisitos e suas priorizações, planejar entregas (*releases*), atuar como facilitador quando mais de um cliente estiver envolvido no projeto, garantir que especialistas de domínio estejam disponíveis para o *team* e dar o aceite final no *sprint* (iteração).
- *Scrum Master* - é responsável por remover a barreira entre desenvolvimento e cliente, ensinar o cliente a maximizar o ROI e atingir seus objetivos através do *Scrum*, melhorar o dia-a-dia dos membros do *team* facilitando a criatividade e fortalecimento, combater a ilusão do comando-controle, priorizar os impedimentos e combatê-los, determinar onde *Scrum* está, comparado com onde poderia estar, auxiliar o *Product Owner* com o *Product Backlog*, garantir o uso do *Scrum* e facilitar reuniões.
- *Team* (time) - a equipe deve ser auto-organizada, multidisciplinar, composta por no máximo 09 integrantes, comprometida com o trabalho e comunicativa. Deve ser responsáveis por fazer aquilo que for necessário para atingir a meta do *Sprint*.

Esses papéis relacionam-se e seguem em um processo básico do *Scrum*, o qual será explicado na próxima seção.

2.2.3 O processo

O *Scrum*, conforme Schwaber (2004), Mariz (2009), Rodrigues e Rost (2007), Magno (2009) e vários autores renomados na área de *Scrum*, divide o desenvolvimento em ciclos, chamados de *sprints*, os quais devem ter duração de duas (02) a quatro (04) semanas.

Os *Teams* devem ser pequenos, entre seis (06) e dez (10) pessoas, os quais são geralmente formados por projetistas, programadores, engenheiros, testadores e gerentes de qualidade, não existindo na literatura um consenso a respeito de quais papéis funcionais compõem essa equipe. Esses profissionais cooperam para atingir uma meta definida no início de cada *sprint* em função das prioridades estabelecidas pelo *Product Owner*.

Tanto o *Team* quanto o *Scrum Master* e o *Product Owner* devem seguir o fluxo proposto pelo *Scrum*. De acordo com Magno (2009), um projeto *Scrum* deve seguir a estrutura ilustrada na Figura 02.

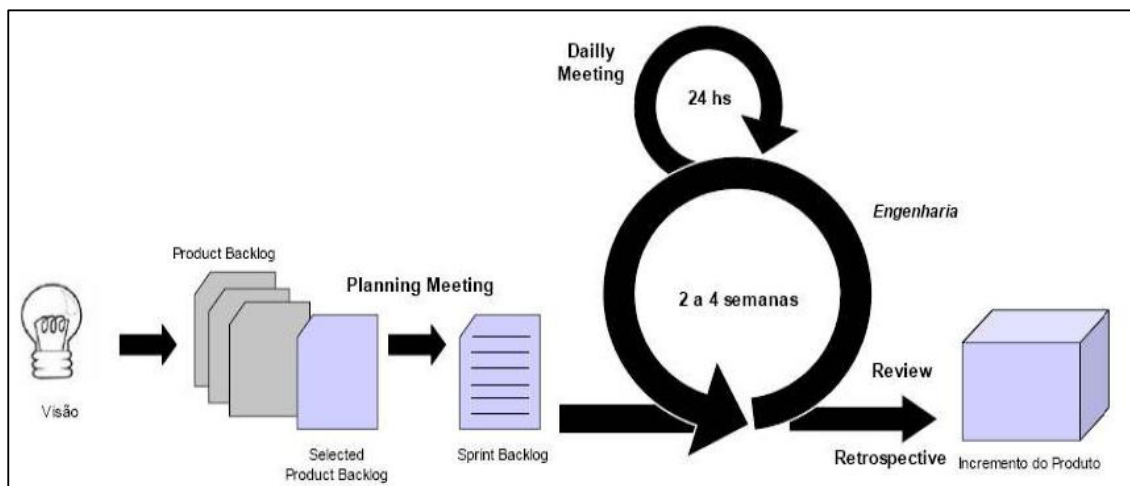


Figura 2: Fluxo SCRUM
Fonte: Magno (2009)

Inicialmente, o *Product Owner (PO)* define uma visão geral do produto. Essa visão representa a sua necessidade, ou seja, o que deve ser satisfeito ao final do projeto. É importante destacar que se o PO for algum membro da empresa desenvolvedora é o próprio cliente, é necessário que colha informações junto aos clientes, usuários finais, *Team*, gerentes, *stakeholders*, executivos, dentre outros, a fim de obter a real necessidade dos envolvidos.

Após a definição da visão geral, o PO começa a criação do *Product Backlog* (PB), que é uma lista de necessidades que precisam ser desenvolvidas para que a visão do projeto seja bem sucedida. Os itens dessa lista devem ser priorizados e ordenados de acordo com o valor que representam para o cliente e o negócio.

O PB irá existir por todo o ciclo de vida do projeto e é regularmente atualizado pelo PO para refletir as mudanças e necessidades do cliente. Um *Product Backlog* é composto de vários tipos de itens, como de funcionalidades, requisitos de desenvolvimento, exploração técnica, estudo, documentação, *bugs*, entre outros. Deve existir apenas um único *Product Backlog* para o projeto, o qual deve refletir uma visão de tudo o que precisa ser feito. O *Scrum Master* deve auxiliar o PO na elaboração dessa lista.

No início de cada iteração (*sprint*) o *Team* deve se reunir para realizar a Reunião de Planejamento. Nesta reunião, o *Team* realizará o planejamento das atividades a serem realizadas durante o *sprint*.

Vale destacar que na primeira parte da Reunião de Planejamento, o PO deverá definir os itens mais prioritários do *Product Backlog* e apresentar a meta do *sprint* ao *Team*.

É importante destacar que para o estabelecimento de uma meta, o *Team* deve ter um conceito de pronto único para o projeto. Conforme exemplificado por Santos (2009), o conceito de “pronto/feito” para um desenvolvedor é quando a codificação está encerrada, para um analista de teste é quando a execução dos testes e verificação dos problemas encontrados estejam concluídas, para o PO uma funcionalidade pronta se dá quando foi entregue aos usuários finais.

Assim, se não houver um consenso a respeito do que é considerado pronto para todos os envolvidos no projeto, torna-se inviável avaliar se a meta estabelecida foi atingida realmente ou não.

Uma vez definida a meta, o *Team* deve estimar o tamanho dos itens a serem desenvolvidos (caso ainda não estejam estimados) e selecionar o que acredita que possa ser feito durante o *sprint*. Essa lista selecionada chama-se *Product Backlog* Selecionado.

Na segunda parte da reunião de planejamento, o *Team* deverá colher mais detalhes a respeito de itens do *Product Backlog* Selecionado e decompô-los em tarefas, gerando assim o *Sprint Backlog*. Após a decomposição, cada membro do *Team* deve selecionar as tarefas de acordo com as prioridades e de acordo com sua função dentro do *Team* para executar durante o *sprint* (sempre negociando com o *Team*) e estimá-las em horas.

Vale ressaltar que durante a execução do *sprint*, devem ser consideradas as práticas de engenharia definidas para o projeto. Por exemplo, se o projeto for a construção de uma casa, a engenharia será a civil, se for um sistema de computador, a engenharia será a de software.

Durante a execução do *sprint*, o *Scrum Master*, por meio da sua capacidade de liderança e colaboração, facilita o trabalho do *Team* removendo os impedimentos encontrados e garantindo a boa aplicação do *Scrum*. Os impedimentos podem ser adicionados a uma lista denominada de *Impediment Backlog* que pode ser preenchida por qualquer membro do *Team*, facilitando assim o *Scrum Master* a identificar os impedimentos de todos os membros da equipe em tempo real.

Durante o *sprint*, o *Team* pode sentir necessidade de consultar especialistas de domínio ou mesmo o *Product Owner* para esclarecimento de dúvidas. Essa prática é perfeitamente aceitável.

Diariamente, o *Team* deve realizar uma reunião de, no máximo 15 minutos (Reunião Diária), na qual cada membro deve responder a três perguntas:

- O que fiz desde a última reunião?
- O que pretendo fazer até a próxima?
- Tive (estou tendo) algum impedimento?

Por meio dessa reunião, o *Team* e o *Scrum Master* ganham visibilidade de como está o caminho para a meta e planejam o dia seguinte de trabalho.

Ao final da execução de cada *sprint*, deve ser realizada a reunião de revisão, na qual o *Team* apresenta para o *Product Owner* e convidados o que foi desenvolvido durante o *sprint*. Nessa reunião, o *Product Owner* avalia se a meta do *sprint* foi alcançada e faz anotações que poderão se transformar em novos itens para o *Product Backlog*.

Após a realização dessa reunião, é realizada a Reunião de Retrospectiva, a qual é facilitada pelo *Scrum Master* e tem o propósito de reunir o *Team* para avaliar as lições aprendidas durante o *sprint*. Durante a reunião, o *Team* deve avaliar:

- O que foi bom no último *sprint*?
- O que poderia ter sido melhor?
- Quais as melhorias para o próxima *sprint*?

Essa reunião representa o espírito de inspeção-adaptação dentro do *Scrum*. É uma reunião do *Team*, mas – caso todos os integrantes estejam de acordo – o PO também pode participar.

Por fim, ao encerrar um *sprint*, caso o *Team* tenha alcançado a meta, haverá uma parte do produto desenvolvida, com funcionalidade que pode ser entregues ao cliente.

O papel do *Scrum Master*, como mostrado no processo, é uma espécie de maestro de um projeto *Scrum* e responsável por remover as barreiras entre o *Team* e o *Product Owner*, melhorar o dia-a-dia dos integrantes do *Team* combatendo os impedimentos surgidos e, em especial, garantir que o *Scrum* seja seguido pelo *Team*. Em vista disso, na seção seguinte, será detalhado o papel do *Scrum Master*, para o qual está direcionado esse trabalho.

2.3 Scrum Master

Nas metodologias tradicionais, os gerentes de projeto exercem seu papel, normalmente, por meio do exercício de “comando e controle”, no qual o líder deve ditar o que, quando e por quem as atividades devem ser executadas.

No *Scrum*, diferentemente desse cenário, não existe a figura de um líder que estabelece o que deve ser feito. De forma oposta, existe um profissional que trabalha com e, principalmente, para o *Team*. Esse profissional exerce o papel conhecido como *Scrum Master*. De acordo com Magno (2008), esse papel tem como principais responsabilidades:

- **Permitir que o *Team* seja auto-gerenciável** - essa é uma das mais desafiadoras responsabilidades de um *Scrum Master*, pois o *Team* espera que exista na equipe alguém como o perfil de comando-controle para dizer o que cada um deve fazer e, consequentemente, controlar o desenvolvimento que foi atribuído a cada membro, como acontece no projeto com metodologia tradicional e ao qual por anos as pessoas foram acostumadas. Porém, em *Scrum* espera-se que os *Teams* sejam autogerenciáveis, o que significa que cada membro do *Team* é responsável por decidir o que vai fazer, como e em quanto tempo de acordo com as prioridades estabelecidas;
- **Garantir que os caminhos para a comunicação do *Team* estejam abertos permanentemente** - em qualquer projeto a comunicação é uma peça chave para o sucesso do projeto, logo, em um projeto que utiliza *Scrum* não seria diferente. Por este motivo, o *Scrum Master* é responsável por garantir que a comunicação do *Team* esteja funcionando de forma satisfatória, ou seja, deve agir como um maestro e garantir que não haja nenhuma barreira, seja física, política, hierárquica ou qualquer outra, existente entre os membros do *Team* e com o *Product Owner*;

- **Garantir e auxiliar o *Team* a seguir corretamente as práticas do *Scrum*** - como um bom maestro, o *Scrum Master* além de garantir a boa comunicação, deve assegurar que as práticas do *Scrum* estão sendo seguidas corretamente durante todo o projeto. O *Team* deve enxergá-lo como uma referência em *Scrum* e uma pessoa capaz de corrigir qualquer deslize na utilização de suas práticas e para tirar as dúvidas relativas a *Scrum* de qualquer pessoa envolvida no projeto;
- **Remover qualquer impedimento que o *Team* encontre** - durante a execução dos *sprints*, os membros do *Team* estarão envolvidos em uma constante comunicação que fará com que os problemas apareçam quase que de imediato. E cabe ao *Scrum Master* a remoção de impedimentos e, por este motivo, ele deve conhecer habilidades que o façam enxergar os impedimentos o quanto antes e removê-los com rapidez;
- **Proteger o *Team* de interferências externas, para garantir que sua produtividade não seja afetada** - o *Scrum Master* deve proteger o *Team* de distrações e manter o *Team* focado na meta do *sprint*. Para isso, ele deve garantir que fatores externos não atrapalhem na produtividade do *Team*; e
- **Facilitar as reuniões do projeto** - O *Scrum Master* deve conduzir a reunião sem influenciá-la, a fim de que elas sejam rápidas e eficientes. Ele tem o papel de facilitador e precisa utilizar todas as técnicas e conhecimento em *Scrum* para fazer com que as reuniões sejam bem sucedidas, fazendo com que o *Team* não perca o foco na reunião.

Schwaber (2004) e Pereira *et al* (2007) concordam com as responsabilidades elencadas por Magno (2008) e ainda complementam as seguintes atribuições ao *Scrum Master*:

- Participar das reuniões diárias, revisão do *sprint*, planejamento e lições aprendidas;
- Garantir que o *Team* esteja totalmente funcional e produtivo;
- Ajudar o *Product Owner* na priorização dos itens de *backlog*;
- Remover as barreiras entre o *Team* e *Product Owner*;
- Ensinar o *Product Owner* a maximizar o ROI; e
- Manter as informações sobre o progresso do *Team* atualizadas e visível para todos os envolvidos no projeto.

Para Araujo *et al* (2006), o *Scrum Master* é responsável por guiar o *Team* e intermediar negociações entre o *Product Owner* e o *Team*. Sua principal função é ensinar e

acompanhar a utilização do *Scrum*. Pereira *et al* (2007) defende que o *Scrum Master* deve gerenciar o processo, ensinando o *Scrum* a todos os envolvidos no projeto e implementando o *Scrum* de modo que esteja adequado à cultura da organização. Além disso, deve garantir que todos sigam as regras e práticas do *Scrum* e é também responsável por remover os impedimentos do projeto.

Diante das responsabilidades apresentadas, é notório o papel importantíssimo do *Scrum Master* em um *Team Scrum*, pois, desempenhando bem seu papel, pode aumentar consideravelmente a probabilidade de bons resultados para o projeto.

Ferreira (2007) comenta que este papel é comparado a de um pai, que tem a responsabilidade de levar o *Team* de um estado de imaturo até a maturidade. Neste estágio, o *Team* consegue se organizar sozinho e se tornar multifuncional, em outras palavras, torna-se autogerenciável.

Pode ser comparado também, segundo Schwaber (*apud* Ferreira, 2007), a um técnico, responsável por animar o *Team*, liderá-lo e ser um guia, ou ainda como um juiz que garante que o *Team* siga as regras do jogo.

De forma complementar, Schwaber (2004) compara o *Scrum Master* a uma espécie de pastor que faria qualquer coisa para proteger seu rebanho. O autor afirma ainda que o *Scrum Master* é um líder e não um gerente. Ele deve ganhar respeito do *Team* por cumprir os deveres de sua função e não simplesmente porque lhe foi atribuído esse papel. Na verdade, um *Scrum Master* que age como um gerente de projeto provavelmente não está cumprindo todas as suas funções como *Scrum Master*.

Contrariando em parte a definição de Schwaber (2004), Mariz (2009) descreve que o *Scrum Master* é gerente do processo *Scrum*, assegurando que a equipe respeite e siga seus valores e práticas.

Apesar de alguns autores divergirem parcialmente em suas definições a respeito do papel do *Scrum Master*, todos concordam que, embora o *Scrum* enfatize que não devem existir hierarquias dentro da equipe, o *Scrum Master* acaba assumindo o papel de um líder que está sempre pronto a remover os impedimentos do projeto e proteger o *Team* de impedimentos internos e externos que possam prejudicar o alcance das metas a serem alcançadas.

Para exercer todas essas responsabilidades, o *Scrum Master* deve apresentar inúmeras habilidades. Cohn (*apud* Mariz, 2009) afirma que as principais qualidades esperadas de um *Scrum Master* são: responsabilidade, colaboração, humildade, comprometimento, influência e conhecimento.

Magno (2008) enfatiza que grande parte destas qualidades citadas por Cohn é ligada diretamente ao lado humano, e não técnico, de um profissional. Resumidamente, é possível verificar que é necessária uma mudança de atitude e comportamento do que novos processos e ferramentas.

Diante disso, pode-se resumir as responsabilidades dos *Scrum Masters* segundo a literatura, conforme o Quadro 3 a seguir.

Quadro 3: Resumo das responsabilidades dos Scrum Masters

1. Permitir que o time seja auto-gerenciável;
2. Garantir que os caminhos para a comunicação do time estejam abertos permanentemente;
3. Remover qualquer impedimento que o time encontre;
4. Proteger o time de interferências externas para garantir que sua produtividade não seja afetada;
5. Facilitar as reuniões do projeto;
6. Garantir que o processo está sendo seguido;
7. Garantir que o time esteja totalmente funcional e produtivo;
8. Ajudar o *Product Owner* na priorização dos itens de *Backlog*;
9. Remover a barreira entre desenvolvimento e cliente;
10. Ensinar o *Product Owner* a maximizar o ROI e atingir seus objetivos através do Scrum;
11. Melhorar o dia-a-dia dos membros do time facilitando a criatividade e fortalecimento;
12. Combater a ilusão do comando-controle;
13. Priorizar os impedimentos e combatê-los;
14. Determinar onde Scrum está, comparado com onde poderia estar; e
15. Manter as informações sobre o progresso do time sempre atualizadas e visíveis a todos.

Fonte: adaptação própria

Em função de tantas responsabilidades que são atribuídas a esses profissionais, é fundamental que os mesmos estejam amparados por ferramentas capazes de auxiliá-los e de fornecerem as informações das quais necessitam o mais rápido possível. Neste sentido, destaca-se a importância de um estudo capaz de analisar ferramentas existentes no mercado, o qual será apresentado na seção 4.2.

Para a realização deste estudo, foi necessária a utilização de uma metodologia para alcançar os objetivos desta pesquisa. Dessa forma, a próxima seção apresenta a metodologia utilizada pela pesquisadora para realização do estudo sobre as principais dificuldades dos *Scrum Masters* entrevistados em relação ao uso de ferramentas para auxiliar suas atividades.

3 METODOLOGIA

Este capítulo tem como objetivo apresentar os aspectos metodológicos utilizados para alcançar o objetivo proposto por esta dissertação, pois de acordo com Gil (2002), uma “pesquisa é desenvolvida mediante o concurso dos conhecimentos disponíveis e a utilização cuidadosa de métodos, técnicas e outros procedimentos científicos”. E para Fachin (2003) pesquisa “é um procedimento intelectual em que o pesquisador tem como objetivo adquirir conhecimentos por meio da investigação de uma realidade e da busca de novas verdades sobre um fato ou problema estudado”. Além disso, dissertação conforme Marconi e Lakatos (2007) é “um estudo teórico, de natureza reflexiva, que consiste na ordenação de idéias sobre determinado tema. Requer sistematização, ordenação e interpretação dos dados e por ser um estudo formal existe metodologia própria de trabalho”.

Com base nesta concepção, este capítulo tem o objetivo de apresentar os métodos científicos utilizados pela pesquisadora, o detalhamento da pesquisa, a classificação da pesquisa, a elaboração do instrumento de coleta, a definição da população, e por fim como a coleta, tabulação, análise e interpretação dos dados foram feitos.

3.1 Métodos científicos

O método segundo Fachin (2006) é “um instrumento do conhecimento que proporciona aos pesquisadores, em qualquer área de sua formação, orientação geral que facilita planejar uma pesquisa, formular hipóteses, coordenar investigações, realizar experiências e interpretar os resultados”.

Os métodos, para Marconi e Lakatos (2007), em geral têm dois momentos diferentes: o primeiro é a Pesquisa, ou coleta de dados, e o segundo é a Análise e Interpretação dos dados coletados.

Segundo Fachin (2006), a pesquisa utiliza também o método científico comparativo que consiste em investigar coisas ou fatos e explicá-los segundo suas semelhanças e suas diferenças. Para Marconi e Lakatos (2007), esse método realiza comparações com a finalidade de verificar similitudes e explicar divergências. Ou seja, permite analisar os dados e deduzir dos mesmos os elementos constantes. Portanto, destaca-se que o método comparativo orientou a pesquisadora na comparação dos dados coletados para encontrar similaridades e explicar diferenças entre os dados respondidos pelos *Scrum Masters* pesquisados.

3.2 Detalhamento da pesquisa

O detalhamento da pesquisa, também chamado de roteiro da pesquisa, apresenta os passos realizados para alcançar os objetivos deste trabalho. No intuito de tornar clara a leitura desse roteiro, a Figura 3 apresenta todos os passos realizados, bem como, suas dependências em um formato gráfico.

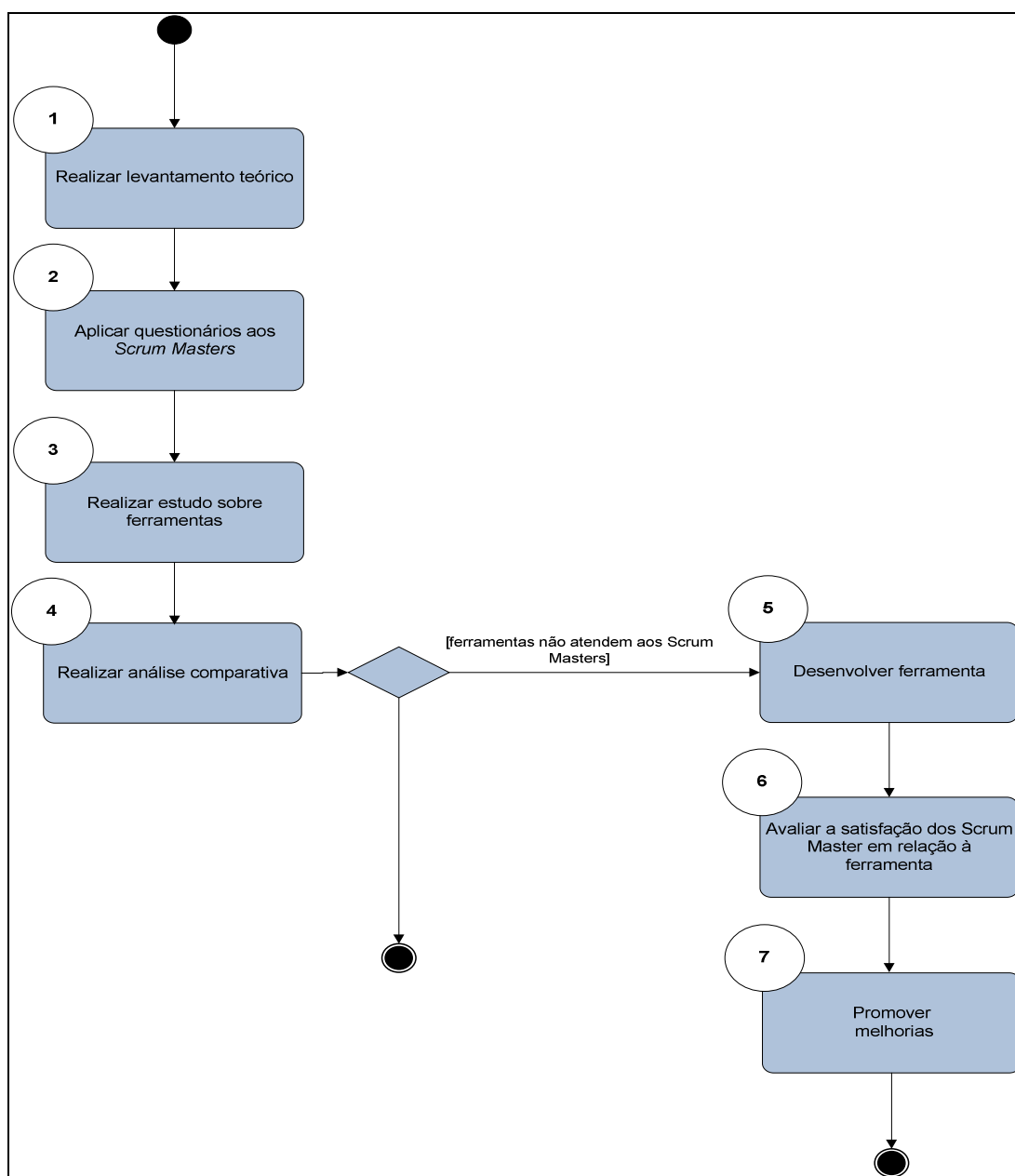


Figura 3: Fases da pesquisa
Fonte: elaboração própria.

Conforme a Figura 3, para atingir os objetivos desta pesquisa, realizou-se sete (07) passos:

Passo 01 - primeiramente foi realizado um levantamento teórico sobre a metodologia *Scrum* e, em especial, a respeito das responsabilidades e atividades do *Scrum Master*, com intuito de fundamentar conceitualmente esta pesquisa. Este levantamento foi realizado com base em livros, artigos, monografias e outros trabalhos acadêmicos;

Passo 02 - com base neste levantamento, foi criado um questionário, o qual foi aplicado com o intuito de investigar: 1) as atividades que estão sendo desempenhadas pelos *Scrum Masters*; 2) as principais ferramentas utilizadas pelos *Scrum Masters* para auxiliar na execução de suas atividades; e 3) verificar se estas atendem as suas necessidades. Concomitantemente, essa investigação também teve o propósito de identificar um conjunto mínimo de funcionalidades que esse tipo de ferramenta deve disponibilizar para que atenda as necessidades e expectativas desses profissionais. Os dados coletados por meio dos questionários foram tabulados, e os resultados estão apresentados na Seção 4.1;

Passo 03 - com base na lista de ferramentas utilizadas pelos *Scrum Masters* para auxiliar na execução de suas atividades mencionadas pelos entrevistados durante a aplicação do questionário, realizou-se um estudo a respeito das principais vantagens e desvantagens dessas ferramentas. Vale destacar que tais ferramentas podem ser tanto computacionais quanto não computacionais;

Passo 04 - com base neste levantamento, realizou-se uma análise comparativa, destacando as limitações, diferenciais e pontos comuns destas ferramentas. Por fim, criou-se uma proposta de solução aderente às reais necessidades dos *Scrum Masters* entrevistados, relacionadas com suas tarefas do dia-a-dia.;

Passo 05 - com base nessa proposta, foi implementada uma ferramenta *free*, desenvolvida com interface Web, chamada JustScrum;

Passo 06 - uma vez implementada, a ferramenta foi testada com alguns *Scrum Masters* e fez-se pesquisa de satisfação com o objetivo de avaliar a ferramenta e obtenção de sugestões de melhorias. Essa pesquisa de satisfação também foi feita por meio da aplicação de questionários e os resultados coletados foram apresentados e discutidos; e

Passo 07 - com base nas respostas dos *Scrum Masters* usuários da ferramenta, as sugestões de melhorias foram analisadas e as mais relevantes foram implementadas seguindo o ciclo apresentado na seção 5.

3.3 Classificação da pesquisa

Sobre a classificação de uma pesquisa, Gil (1999) preconiza que essa deve ser realizada de acordo com dois (02) aspectos principais: os objetivos (fins) e as formas de estudos (meios).

Quanto aos fins ou objetivos, este trabalho ficou classificado como pesquisa descritiva e exploratória.

Quanto aos meios ou procedimentos científicos, este trabalho utilizou das diretrizes da pesquisa de campo com aplicação de questionários, revisão bibliográfica e análise qualitativa e análise quantitativa dos resultados das entrevistas.

3.3.1 Pesquisa descritiva e exploratória

A pesquisa do tipo descritiva segundo Silva e Menezes (2001), visa descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis. Envolve o uso de técnicas padronizadas de coleta de dados como: questionário e observação sistemática. Para Vergara (2000), a pesquisa descritiva não tem o compromisso de explicar os fenômenos que descreve, embora sirva de base para tal explicação.

Nesse tipo de pesquisa os dados são registrados e analisados, com o intuito de descobrir a frequência com que um fato ocorre.

Neste contexto, a pesquisadora utilizou-se desta pesquisa para levantar a concordância dos *Scrum Masters* sobre as suas atividades diárias e as ferramentas utilizadas, através de questionário.

Já a pesquisa exploratória, utilizada pela pesquisadora, para Gil (2002) tem como “objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipóteses”, ou seja, pode-se afirmar que esse tipo de pesquisa tem como objetivo principal o aprimoramento de idéias e na maioria dos casos, utiliza técnicas de levantamento bibliográfico, entrevistas com pessoas que tiveram ou têm experiência com o assunto pesquisado e análise de exemplos. Esse tipo de pesquisa limita-se a definir o objetivo e buscar maiores informações sobre o assunto em questão.

Baseando-se nesta concepção, inicialmente, procurou-se compreender o cenário estudado através da pesquisa de campo e revisão bibliográfica sobre a metodologia *Scrum*, em

especial a respeito das responsabilidades e atividades do *Scrum Master*, assim como as ferramentas utilizadas por estes profissionais.

3.3.2 Pesquisa de campo

A pesquisa de campo envolve tanto as obras levantadas em diversos formatos, como também os participantes da pesquisa. Fachin (2003) define pesquisa de campo como a observação do contexto no qual é detectado um fato (problema), que a princípio passa a ser examinado e, posteriormente, é encaminhado para explicações por meio de métodos e das técnicas específicas.

Para Marconi e Lakatos (2002), a pesquisa de campo é utilizada com o objetivo de conseguir informações e/ou conhecimentos sobre um problema para o qual se procura uma solução/ resposta. Consiste na “observação de fatos e fenômenos tal como ocorrem espontaneamente, na coleta de dados a eles referentes e no registro de variáveis que se presumem relevantes, para analisá-los”.

Na pesquisa de campo se usou como instrumentos de trabalho o formulário e o questionário. O primeiro constitui-se segundo Fachin (2006), em “uma série de questões ordenadas sucessivamente e relacionadas com o objetivo do estudo”. Na aplicação desse instrumento é necessária a assistência do pesquisador ao pesquisado, havendo assim um intenso contato pessoal entre ambas as partes. O segundo instrumento, questionário, Fachin (2006) afirma que consiste em “um elenco de questões que são submetidas a certo número de pessoas com o intuito de se coletar informações”. Sua aplicação pode ser pessoalmente, via postal, por malote, telefone, email, entre outros.

Neste trabalho, optou-se por trabalhar com questionários que foram mandados via email para os participantes da pesquisa, conforme apresentado na seção 3.4 e Apêndice A.

3.3.3 Revisão bibliográfica

Para Manzo (*apud* Marconi e Lakatos, 2002), a bibliografia pertinente “oferece meios para definir, resolver, não somente problemas já conhecidos, como também explorar novas áreas onde os problemas não se cristalizaram suficientemente” e tem como objetivo, segundo Trujillo (*apud* Marconi e Lakatos, 2002), “permitir ao cientista o reforço paralelo na análise de suas pesquisas ou manipulação de suas informações”.

Barros e Lehfeld (2000) explicam que a pesquisa bibliográfica é aquela que “se

efetua tentando-se resolver um problema ou adquirir conhecimentos a partir do emprego predominante de informações advindas de material gráfico, sonoro e informatizado".

A pesquisa bibliográfica fundamentou a realização da investigação. Esta permitiu à investigadora se aprofundar no tema em estudo, através de estudiosos da área. Gressler (2004) diz que o pesquisador ao problematizar um assunto, provavelmente, já tenha conhecimento prévio sobre este assunto. Assim, o objetivo da pesquisa bibliográfica é a atualização e integração do pesquisador com esse conhecimento.

3.3.4 Análise quantitativa e qualitativa

Triviños (1987) diz que, na prática, toda pesquisa pode ser, ao mesmo tempo, quantitativa e qualitativa, desde que o pesquisador aproveite a informação quantitativa para avançar numa interpretação mais ampla da mesma.

O método quantitativo segundo Richardson *et al.* (*apud*, Marconi e Lakatos, 2007), “caracteriza-se pelo emprego da quantificação tanto nas modalidades de coleta de informações quanto no tratamento delas por meio de técnicas estatísticas, desde as mais simples como percentual, média, desvio-padrão, às mais complexas como coeficiente de correlação, análise de regressão etc”. Nesse método, os pesquisadores valem-se de amostras amplas e de informações numéricas.

O método qualitativo difere do quantitativo segundo Marconi e Lakatos (2007) não só por não empregar instrumentos estatísticos, mas também pela forma de coleta e análise de dados, pois esse método preocupa-se em analisar e interpretar aspectos mais profundos, descrevendo a complexidade do comportamento humano, fornecendo análise mais detalhada sobre as investigações, hábitos, atitudes, tendências de comportamento, entre outros. No método qualitativo, os pesquisadores valem-se de amostras reduzidas, os dados são analisados em seu conteúdo psicossocial e os instrumentos de coleta não são estruturados.

Assim, esta pesquisa segue o direcionamento da pesquisa quantitativa e qualitativa. Uma vez que se objetiva investigar, por meio de questionários, as atividades desempenhadas pelos *Scrum Masters* e as principais ferramentas utilizadas por eles para auxiliar na execução de suas atividades, isto caracteriza a pesquisa em quantitativa. A pesquisa é qualitativa quando se propõe verificar se essas ferramentas atendem às necessidades, identificando as vantagens e desvantagens das ferramentas existentes que têm o propósito de auxiliar os *Scrum Masters* na execução de suas responsabilidades.

3.4 Elaboração de Instrumentos de Coleta de Dados

Para o levantamento de dados, Gil (2006) afirma que são utilizadas técnicas de interrogação que podem ser o questionário, a entrevista e o formulário, onde:

- Entrevista - refere-se à técnica que envolve duas pessoas numa situação face-a-face, onde no decorrer do processo de coleta de dados, o pesquisador formula questões e o pesquisado responde;
- Questionário - refere-se à técnica de coleta de dados em que o pesquisador formula questões previamente, as quais são respondidas por escrito pelo próprio respondente sem ajuda ou participação do pesquisador; e
- Formulário - refere-se à técnica de coleta de dados em que o pesquisador formula questões previamente e anota as respostas do respondente.

Quanto aos tipos de questões, Gil (1999) apresenta três (03) tipos: abertas, fechadas e dependentes. As questões abertas não possuem valores pré-definidos, admitindo quaisquer respostas. As questões fechadas possuem valores pré-definidos, forçando o respondente a enquadrar sua percepção em uma das respostas. As questões dependentes são aquelas que são questionadas ou não em decorrência da resposta de uma questão anterior.

Diante do que foi exposto, destaca-se que este trabalho utilizou como instrumento de coleta de dados o questionário estruturado com quatorze (14) questões o qual pode ser visualizado no Apêndice A. Estas questões foram divididas em quatro (04) questões abertas e dez (10) questões fechadas.

Além disto, o questionário foi dividido em quatro (04) partes agrupadas por assuntos relacionados:

- Parte I - Identificação do sujeito da pesquisa (informações gerais sobre o profissional entrevistado);
- Parte II - Identificação das atividades desempenhadas pelos *Scrum Masters* (informações sobre as atividades cotidianas desempenhada pelos *Scrum Masters* em um projeto *Scrum*);
- Parte III - Identificação das ferramentas utilizadas pelos *Scrum Masters* (informações relacionadas às ferramentas usualmente utilizadas pelos *Scrum Masters* para auxiliar a execução de suas atividades, bem como verificar se estas atendem às expectativas desses profissionais); e
- Parte IV - Identificação das funcionalidades que uma ferramenta computacional deve disponibilizar para auxiliar os *Scrum Masters* (Se na parte

III, for identificado que as ferramentas atualmente utilizadas não atendem às necessidades desses profissionais, será identificado um conjunto mínimo de funcionalidades que tais ferramentas devem apresentar).

3.5 Definição da População para Coleta de Dados

Entende-se por população um conjunto de seres que apresentam pelo menos uma característica em comum. Portanto, a pesquisa considerou como população ou universo o conjunto de *Scrum Masters* (pessoas que participaram do curso *Certified Scrum Master*) cadastrados nas listas “Scrum Master Recife” e “Scrum Master Amazônia”. Entretanto, como não foi possível descobrir a quantidade de participantes em cada lista, não foi possível definir o tamanho da amostra, utilizando o método probabilístico. Segundo Gil (1999), os tipos de amostragem podem ser classificados em por acessibilidades, por tipicidade e por cotas. Nesta pesquisa, foi utilizado o tipo de amostragem não probabilista por acessibilidade ou conveniência, a qual segundo Rea & Parker (2000) é feita quando os entrevistados são selecionados com base na sua semelhança presumida com a população útil e na sua disponibilidade imediata. Apesar de não ter um rigor estatístico, Dantas (2005) afirma que isso não invalida a pesquisa.

Com o intuito de embasar a escolha de entrevistados para compor a amostra, definiu-se as seguintes características para os participantes da pesquisa:

- Estar envolvido em projetos que utilizem *Scrum* como metodologia de gestão; e
- Responder como *Scrum Master* desses projetos.

Antes de selecionar a amostra, foi enviado um convite, via email, para as duas listas mencionadas.

Uma vez que as características estavam definidas, foi enviado um convite, via e-mail, para as listas *Scrum Recife* (recife@scrum.org.br) e *Scrum Amazônia* (scrum-amazonia@yahoogrupos.com.br) solicitando voluntários para participar da pesquisa, apresentando a pesquisadora e explicando o objetivo da pesquisa.

Como resultado, obteve-se vinte e sete (27) voluntários, dentre os quais se fez necessário realizar um corte de oito (08) voluntários, pois não atendiam às características dos participantes da amostra proposta por este trabalho. Todavia, dentre os dezenove (19) participantes restantes, apenas treze (13) retornaram o questionário respondido. Portanto, o tamanho da amostra não probabilística por conveniência totalizou o número de treze (13) participantes.

3.6 Coleta, tabulação, análise e interpretação dos dados

A coleta de dados, segundo Marconi e Lakatos (2002), consiste na “etapa da pesquisa em que se inicia a aplicação dos instrumentos elaborados e das técnicas selecionadas, a fim de se efetuar a coleta dos dados previstos”. Para Barros e Lehfeld (2000, p. 89) “a coleta de dados significa a fase da pesquisa em que se indaga e se obtêm dados da realidade pela aplicação de técnicas”. A coleta de dados desta pesquisa realizou-se entre os dias 24/11/2009 a 15/12/2009.

Após a coleta dos dados, segundo Padua (2004) inicia-se o processo de análise, classificação e interpretação das informações coletadas. Para facilitar a compreensão, os dados coletados foram tabulados dispostos em tabelas e gráficos. Para isto, foi utilizado o Microsoft Office Excel 2007 [www.microsoft.com]. Segundo Abramo (*apud* Marconi e Lakatos, 2002), a tabulação “é a arrumação dos dados em tabelas, de maneira a permitir a verificação das relações que eles guardam entre si”. E para Teixeira (2003), tabulação trata-se de um processo de agrupar e contar os casos que estão nas várias categorias de análise.

Na tabulação, utilizaram-se as linhas para representar as respostas dos questionários e as colunas para as perguntas. Na medida em que os questionários iam sendo enviados, as respostas iam sendo tabuladas.

Após a tabulação dos dados foram criados gráficos de cada questão fechada para melhor visualização dos dados para análise e para as questões abertas foram agrupadas as respostas iguais para serem analisadas.

Em seguida à coleta e à tabulação dos dados, foram feitas a análise e interpretação dos mesmos a fim de se detectar os pontos de divergência e/ou convergência entre os dados coletados em relação à teoria e à prática apresentadas durante a pesquisa. Segundo Gil (1999), “a análise de dados tem como objetivo organizar e sumarizar os dados de tal forma que possibilitem o fornecimento de respostas ao problema proposta para investigação. Já a interpretação tem como objetivo a procura do sentido mais amplo das respostas, o que é feito mediante sua ligação a outros conhecimentos anteriormente obtidos”. De acordo com Marconi & Lakatos (2002), a interpretação é uma atividade intelectual que procura dar um significado mais amplo às respostas obtidas da coleta de dados, vinculando-as a outros conhecimentos e relacionando-as aos objetivos propostos e ao tema estudado na pesquisa.

4 RESULTADOS DA PESQUISA

Neste capítulo são apresentados os resultados da pesquisa realizada para avaliar as atividades desempenhadas pelos *Scrum Masters*, as principais ferramentas utilizadas por estes profissionais para auxiliar na execução de suas atividades, verificar se estas atendem as suas necessidades, e em caso contrário, identificar um conjunto mínimo de funcionalidades que os *Scrum Masters* entrevistados esperam de uma ferramenta. Os dados obtidos são apresentados conforme a sequência das questões do instrumento de coleta de dados e agrupados em quatro (4) partes, conforme o questionário apresentado no Apêndice A.

Na primeira parte, composta pelas questões de 1 a 5, foi analisada a identificação do participante da pesquisa. Na segunda parte, a qual envolve somente a questão 6, foram analisados os dados relativos às reais atividades desempenhadas pelos *Scrum Masters*. Na terceira parte, foram analisados os dados das questões 7 a 13, a respeito das ferramentas não computacionais e computacionais sugeridas pelos *Scrum Masters* entrevistados, com o objetivo de verificar a aderência das mesmas às expectativas desses profissionais. Na quarta e última parte, apenas com a questão 14, foram identificadas as funcionalidades computacionais que os *Scrum Masters* necessitam para a melhor execução de suas atividades.

Para a representação dos dados das questões fechadas (objetivas) de 01 a 08, 10 e 11 foram utilizados gráficos para facilitar ao leitor o entendimento dos resultados dos entrevistados. As respostas das perguntas 09, 12 e 13 são apresentadas em forma de texto, pois as mesmas representam as questões abertas (subjetivas) do instrumento da pesquisa. As respostas da questão 14 foram agrupadas de acordo com a frequência e apresentadas na forma de um quadro para melhor visualização das respostas.

Os participantes da pesquisa são identificados na representação dos dados como respondentes do questionário.

Na representação dos dados usa-se a seguinte identificação: % (N= número de participantes). Também na representação dos dados identificam-se os participantes como: N=13, que corresponde aos participantes respondentes das perguntas.

Nas questões com a opção de escolha de mais de uma alternativa foi calculado o percentual em cima da quantidade de respondentes, com a fórmula: $(\text{Quantidade de respondente por alternativa} * 100) / N$. Os resultados obtidos foram arredondados para não serem números quebrados.

Vale ressaltar que alguns desses 13 respondentes do questionário deixaram algumas questões em branco. Dessa forma, durante a análise dos dados, a quantidade dos respondentes que não responderam cada pergunta foi destacada.

4.1 Representação e análise dos dados

O tipo de gráfico utilizado na pesquisa é o circular, denominado também como pizza, que foram elaborados no Microsoft Excel. Os gráficos apresentados têm a finalidade de informar a situação atual e real do problema estudado, de forma clara e de fácil entendimento. Marconi e Lakatos (1999) afirmam que representar os resultados estatísticos através de elementos geométricos, permite uma descrição imediata do fenômeno estudado.

A análise feita pela pesquisadora visa verificar os dados respondidos pelos participantes da pesquisa a fim de averiguar quais as principais dificuldades dos *Scrum Masters* em suas atividades diárias e quanto ao uso de ferramentas que possam auxiliá-los.

4.1.1 Parte I - Identificação do sujeito da pesquisa

A parte I desta pesquisa serve para a identificação do participante da pesquisa, a fim de obter informações específicas de cada profissional entrevistado para haver uma previa seleção dos participantes mais simbólicos e que contribui para o resultado da pesquisa.

Questão 1 - Você possui alguma certificação? Qual? (Pode assinalar mais de uma opção).

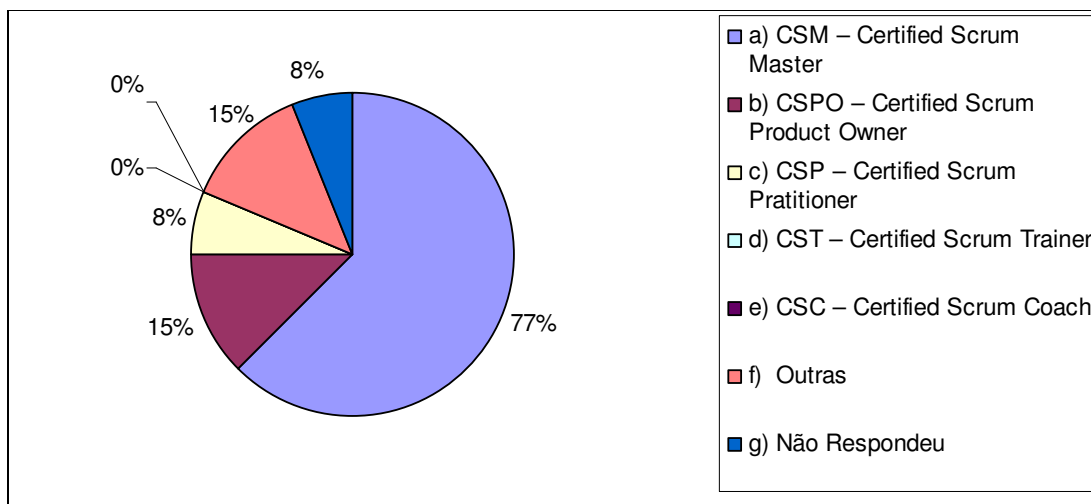


Gráfico 3: Certificação

Fonte: Elaboração própria, com dados da pesquisa de campo.

Conforme o Gráfico 03 dentre os respondentes, 77% (N=10) marcaram a opção *Certified Scrum Master* (CSM). Os demais ficaram distribuídos em 15% (N=2) com *Certified Scrum Product Owner* (CSPO), 15% (N=2) responderam Outras conforme opção disponível no questionário, porém identificaram que tinham certificação PMP. Além desses, 8% (N=1) responderam que possuíam certificação *Certified Scrum Practitioner* (CSP) e 8% (N=1) não responderam. Foram listadas no questionário também as certificações *Certified Scrum Trainer* (CST) e *Certified Scrum Coach* (CSC), porém, não tiveram respondentes.

Ao analisar os dados coletados na Questão 1, pode-se observar que mais da metade dos entrevistados tinham a certificação CSM, a qual é de extrema importância para a formação dos *Scrum Masters* e consequentemente, para a credibilidades dos participantes da pesquisa uma vez que se trata de um trabalho com o objetivo de investigar as principais dificuldades desses profissionais em relação ao uso de ferramentas voltadas para suas atividades.

Duas pessoas responderam que além de ter a certificação CSM tinham a CSP, o que torna muito interessante suas respostas, por estes terem não somente a visão do *Scrum Master* mas também a visão do *Product Owner*, peça fundamental para o desenvolvimento de projetos *Scrum*.

Dentre as pessoas que responderam que tinham outras certificações, uma afirmou que além do CSM e CSPO tinha o PMP e outra somente o PMO. O respondente com as três certificações é de enorme valor para esta pesquisa, pois além de ter as certificações de *Scrum Master* e *Product Owner*, também possui a certificação PMP, demonstrando seu conhecimento nas boas práticas de gerenciamento de projetos tanto ágil como tradicional baseado no PMBOK [www.pmi.org].

Vale ressaltar, também, que devido à transição dos projetos tradicionais para os ágeis é comum encontrar profissionais PMP se certificando em CSM, as quais são certificações equivalentes em se tratando em gerenciamento de projetos, porém de metodologias diferentes.

Um dos respondente marcou a opção *Certified Scrum Practitioner*. Essa certificação é atribuída a profissionais CSM que já praticam o *Scrum* por pelo menos um ano e que preenchem um questionário descrevendo suas experiências em um projeto e como se aplica a metodologia. Esse questionário é avaliado e acompanhado pela *ScrumAlliance*.

O respondente que disse não possuir nenhuma certificação foi aceito nesta pesquisa, por se tratar de uma pessoa iniciante nesta metodologia, o que oferece à pesquisadora uma visão importante das dificuldades encontradas pelas pessoas ao iniciar no *Scrum*.

Questão 2 - Há quanto tempo você trabalha com a metodologia de gerenciamento *Scrum*?

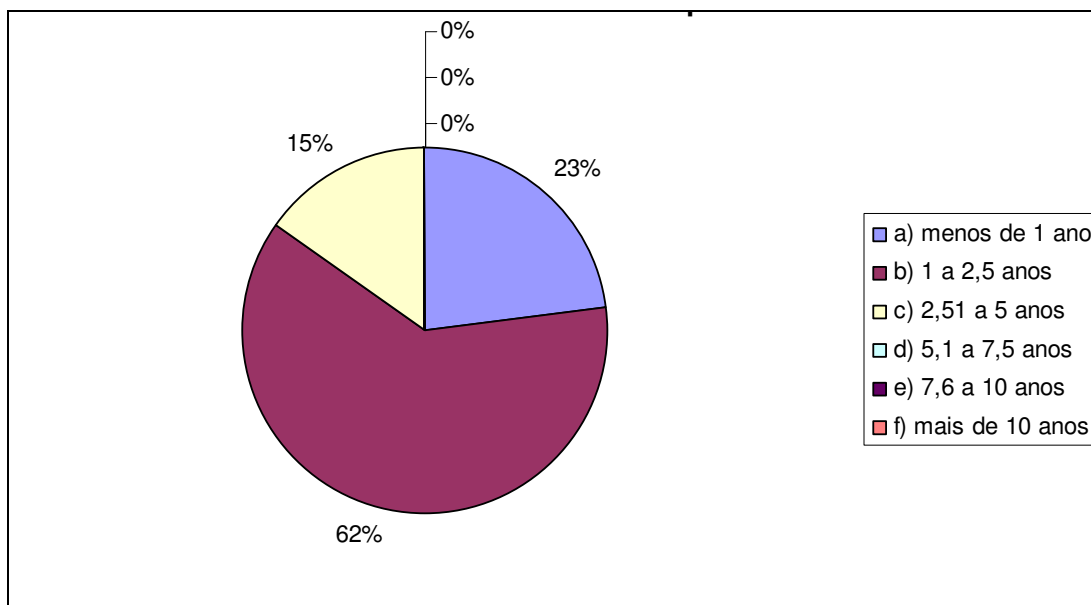


Gráfico 4: Tempo de trabalho com a metodologia de gerenciamento Scrum

Fonte: Elaboração própria, com dados da pesquisa de campo.

Os resultados obtidos relativos ao tempo de trabalho, demonstrados no Gráfico 4, mostram que 62% (N=8) trabalham de 1 a 2,5 anos com a metodologia *Scrum*, seguidos de 23% (N=3) que trabalham menos de 1 ano e 15% (N=2) que trabalham de 2,51 a 5 anos. No questionário também foram disponibilizadas as opções: 5,1 a 7,5 anos, 7,6 a 10 anos e mais de 10 anos, porém, as mesmas não tiveram respostas.

Dentre os 23% dos entrevistados que tem menos de um ano de experiência trabalhando com o *Scrum*, pode-se identificar que, mesmo tendo pouco tempo trabalhando com a metodologia, dois (2) responderam que têm a certificação CSM e somente um não tem nenhuma certificação.

Todos os 63% que responderam ter mais de um ano de experiência, possuem certificação CSM. Além disso, um (1) ainda tem a CSPO e somente um (1) não tem nenhuma certificação relacionada à metodologia *Scrum*, mas possui a certificação PMP. Em relação aos respondentes que marcaram a opção de mais de 02 anos e meio de experiência, um (1) tem as certificações de CSM, CSPO e PMP e outro tem somente a CSM.

Neste contexto, cruzando as respostas das questões 01 e 02, observou-se que quanto maior o tempo de conhecimento na metodologia, maior a busca por qualificações, o que pode ser comprovado com o número de certificações dos respondentes. Com as certificações é comprovado que o respondente tem profundos conhecimentos na metodologia, o que dá maior

credibilidade ao universo amostral utilizado nesta pesquisa, que se preocupou em entrevistar respondentes que possuísem não só bastante conhecimento na metodologia como também experiência.

Questão 3 - Há quanto tempo você trabalha como Scrum Master?

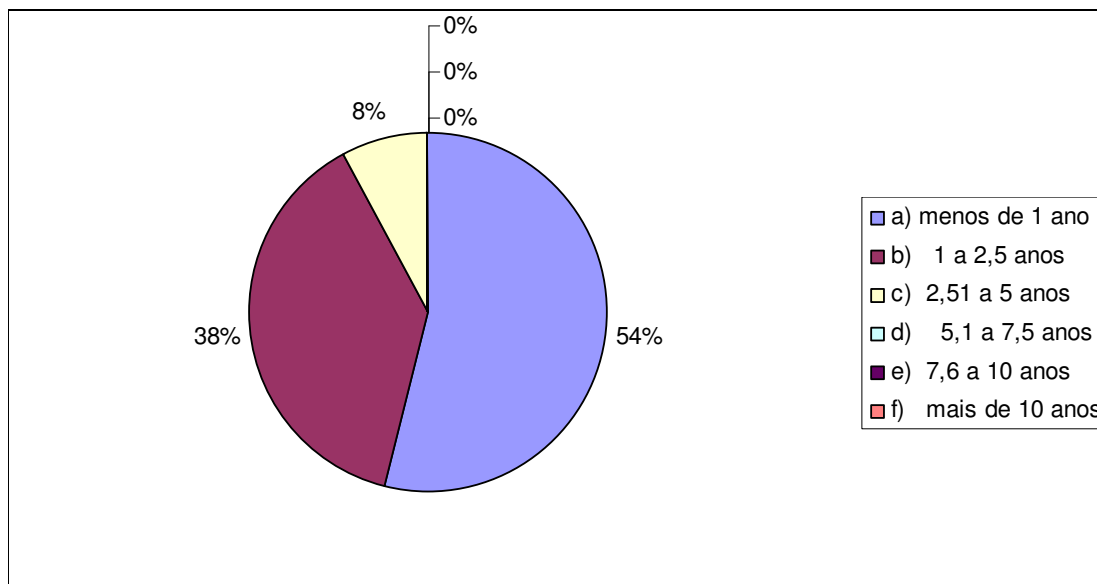


Gráfico 5: Tempo como *Scrum Master*

Fonte: Elaboração própria, com dados da pesquisa de campo

Os resultados conforme o Gráfico 05 foram: 54% (N=7) disseram trabalhar menos de 1 ano, 38% (N=5) trabalham de 1 a 2,5 anos e 8% (N=1) trabalham de 2,51 a 5 anos. Os itens do questionário de 5,1 a 7,5 de 7,6 a 10 e mais de 10 anos não obtiveram respostas.

Cruzando as respostas obtidas nas questões 02 e 03, constatou-se que dos 77% (N=10) que confirmam ter mais de um ano de experiência com a metodologia, 54% (N=7) dizem trabalhar a menos de um ano como *Scrum Master*. Assim, pode-se concluir que boa parte das profissionais exercem outros papéis em um projeto *Scrum* antes de se tornar um *Scrum Master* e isto também pode ser confirmado no Gráfico 6 da Questão 4.

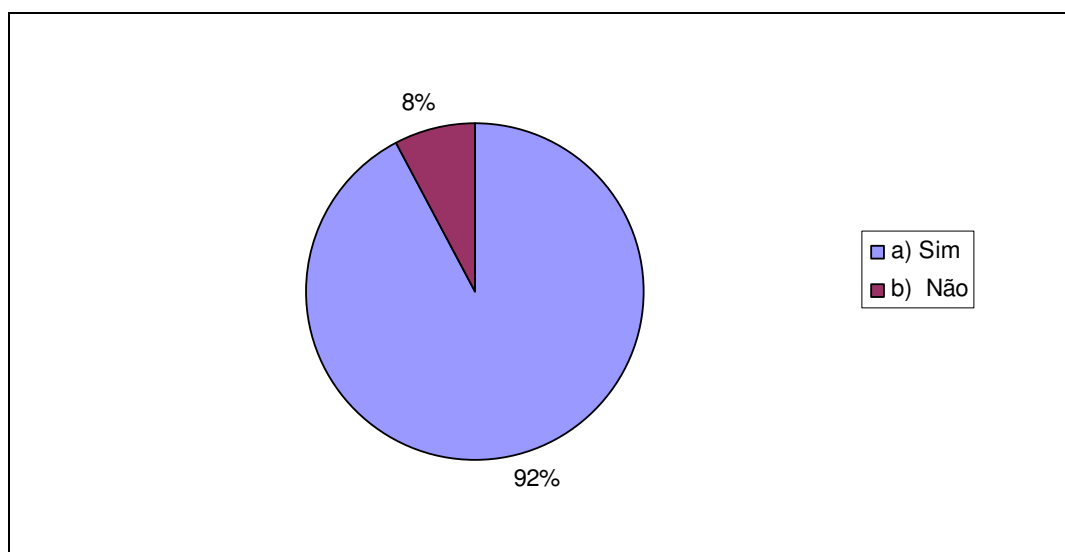
Questão 4 - Você atua em outra função além de *Scrum Master*?

Gráfico 6: Atuação em outra função além de *Scrum Master*

Fonte: Elaboração própria, com dados da pesquisa de campo.

Os resultados obtidos com referência à atuação em outra função, conforme o Gráfico 06 foi: 92% (N=12) responderam que sim e 8% (N=1) responderam que não atuavam em outra função, conforme observado no 4º Gráfico.

Na Questão 04, dentre os que afirmaram não trabalhar em nenhuma função além de *Scrum Master*, tem experiência de menos de um 1 ano nesse papel e com a metodologia *Scrum*, embora já possua a certificação CSM. Os demais respondentes atuam em várias outras funções além de *Scrum Master*, as quais são detalhadas na Questão 05.

Questão 5 - Se a resposta da questão anterior for “SIM”, responda: Quais as funções? (Pode assinalar mais de uma opção).

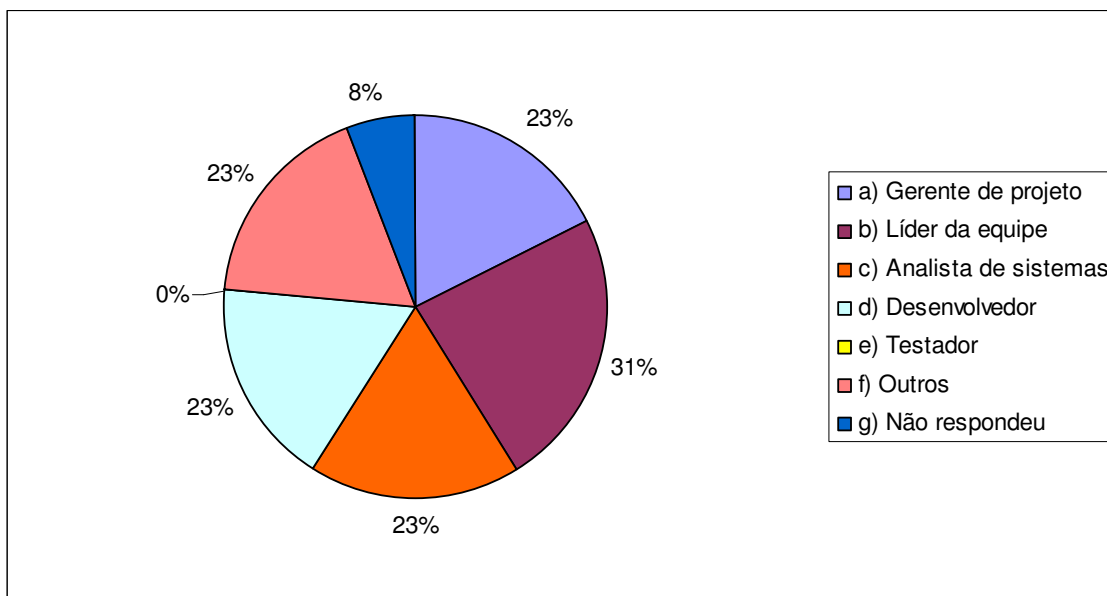


Gráfico 7: Quais funções que atua além de Scrum Master
 Fonte: Elaboração própria, com dados da pesquisa de campo.

Conforme o Gráfico 07, dos participantes respondentes, 31% (N=4) desenvolvem a função de líder da equipe; 23% (N=3) gerente de projetos, 23% (N=3) analista de sistemas; 23% (N=3) desenvolvedor; 23% (N=3) Outros. Dentre os que marcaram esta última opção, (N=1) respondeu que era Arquiteto de Soluções de SW e Interface; (N=1) participante respondeu que era Líder de Implantação e um (1) participante respondeu que era *Designer*. Além disso, 8% (N=1) não responderam e ninguém marcou a opção de testador.

Na Questão 05, nota-se que a maioria dos respondentes, 54%, atua além de *Scrum Master* como líderes de equipe e gerentes de projetos, o que reforça a idéia de que as funções desempenhadas pelos respondentes nas metodologias tradicionais quando passadas para as metodologias ágeis são equivalentes, ou seja, assim como a metodologias, as funções também são diferentes no projeto.

Comparando os gráficos das Questões 04 e 05, percebe-se que os *Scrum Masters* respondentes em sua maioria desempenham outro papel no *Time*, fato este que vai de encontro à literatura, a qual afirma que isto não deve ocorrer e, caso ocorra, os papéis devem ser desempenhados em projetos distintos.

4.1.2 Parte II - Identificação das atividades desempenhadas pelos Scrum Masters

A Parte II deste do questionário teve o intuito de identificar as atividades desempenhadas pelos *Scrum Masters*, ou seja, informações sobre as atividades cotidianas desempenhada pelos *Scrum Masters* em um projeto *Scrum*.

Questão 6 - No seu cotidiano de *Scrum Master*, quais as atividades desempenhadas nos projetos das quais participa? (pode escolher mais de uma opção)

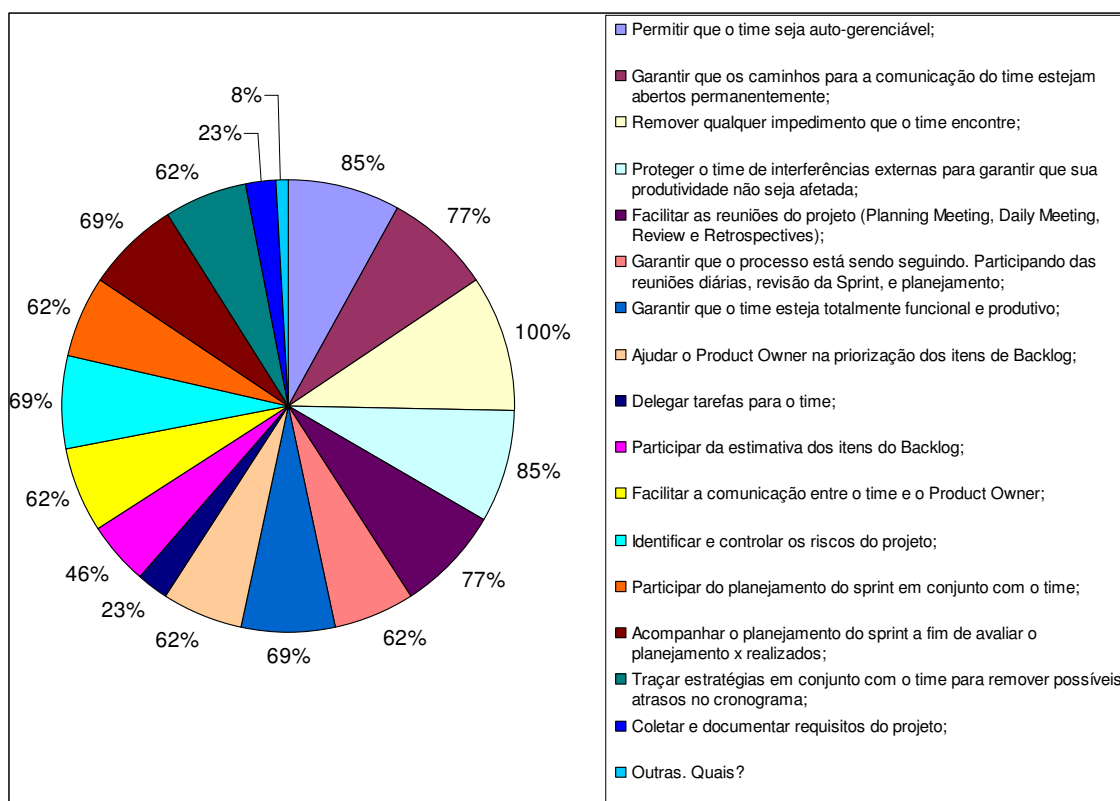


Gráfico 8: Atividades desempenhadas nos projetos
 Fonte: Elaboração própria, com dados da pesquisa de campo.

Esta pergunta foi dividida em 17 itens, os quais foram representados pelas letras de a/q. Nesta questão, foi permitido aos participantes que assinalassem mais de um item. Assim, obtiveram-se 135 respostas, divididas entre N=13 participantes da pesquisa.

As respostas representadas em percentual, conforme o Gráfico 8, foram alcançados os seguintes resultados: 100% (N=13) disseram que removem qualquer impedimento que o time encontre; 85% (N=11) protegem o time de interferências externas para garantir que sua produtividade não seja afetada; 85% (N=11) permitem que o time seja auto-gerenciável; 77% (N=10) disseram garantir que os caminhos para a comunicação do time estejam abertos

permanentemente; 77% (N=10) disseram facilitar as reuniões do projeto (*Planning Meeting*, *Daily Meeting*, *Review e Retrospectives*); 69% (N=9) disseram que identificam e controlam os riscos do projeto; 69% (N=9) acompanham o planejamento do *Sprint* com a finalidade de avaliar o planejamento x realizados; 69% (N=9) disseram garantir que o time esteja totalmente funcional e produtivo; 62% (N=8) responderam garantir que o processo está sendo seguindo. Participando das reuniões diárias, revisão do *Sprint*, e planejamento; 62% (N=8) ajudam o *Product Owner* na priorização dos itens de *Backlog*; 62% (N=8) Facilitam a comunicação entre o time e o *Product Owner*; 62% (N=8) Participam do planejamento do *sprint* em conjunto com o time; 62% (N=8) Traçam estratégias em conjunto com o time para remover possíveis atrasos no cronograma; 46% (N=6) Participam da estimativa dos itens do *Backlog*; 23% (N=3) Delegam tarefas para o time; 23% (N=3) Coletam e documentam requisitos do projeto e 8% (N=1) Outras atividades.

No item Outros, o participante respondente disse que desenvolve outras atividades como: esclarece dúvidas do time a respeito dos PBIs (itens do *product backlog*) que estão sendo desenvolvidos no *sprint*; que atualiza os documentos de requisitos e projetos de tela em função de sugestões reportadas pelos desenvolvedores, testadores e usuários, além de fazer essas atualizações em razão de novas funcionalidades a serem implementadas (desenvolvimento incremental); que comunica alterações de requisitos a toda a equipe; que mantém a base de dados do projeto e comunica as alterações feitas; mantém contato com o cliente, reportando: prazos de melhorias solicitadas, esclarece dúvidas e impedimentos e por fim acompanha o processo de implantação do software.

Na Questão 06 perguntaram-se aos *Scrum Masters* quais eram as atividades desempenhadas por eles no dia-a-dia em um projeto *Scrum*. A partir dos dados ilustrados no Gráfico 08, percebeu-se que na prática, os *Scrum Masters* entrevistados desempenham atividades que extrapolam as atividades que um *Scrum Masters* deve exercer segundo a revisão da literatura.

Como exemplo, pode-se destacar a opção: delegar tarefas para o Time. Esta atividade é abolida da metodologia ágil, pois defende que o estilo de liderança de um *Scrum Master* deve ser de facilitação e não de comando-controle, onde o gerente de projeto distribui as tarefas aos responsáveis por executá-las, de acordo com: habilidade técnica, conhecimento prévio da tarefa ou experiência.

Nesta opção teve três (03) respondentes, justificável, pois, os 03 atuam além da função de *Scrum Master* como Gerente de projeto e não perderam o hábito do comando-controle das metodologias tradicionais. O que reforça a idéia de como é difícil a adoção do

Scrum, pois as pessoas já estão acostumadas ao método antigo e têm medo do desconhecido. Em se tratando dos gerentes de projetos é comum não querer perder o poder de mandar na equipe e assim perder o controle da situação e do projeto.

Na opção coletar e documentar requisitos de projetos houve três (03) respondentes que afirmaram fazer esta atividade enquanto exerce a função de *Scrum Master*, porém esta também é uma atividade que é abolida para o *Scrum Master* que deve apenas facilitar o desenvolvimento correto do processo, além do que os requisitos devem ser descritos e documentados no *Backlog* que pode ser alterado em qualquer momento durante o projeto.

Um ponto positivo na pesquisa é que todos os respondentes do questionário marcaram a principal atividade do *Scrum Master* que é o de remover qualquer impedimento que o time encontre.

Porém, duas (02) pessoas não marcaram a opção de permitir que o time seja autogerenciável, que se pode dizer que é a principal atividade que um *Scrum Master* deve exercer. Dentre estes respondentes, um tem menos de um ano utilizando o *Scrum*, não possui nenhuma certificação e exerceu atividades de analista de sistemas e desenvolvedor, ou seja, esta pessoa faz mais parte do time do que é *Scrum Master* do projeto, e por ter pouca experiência ainda não tem a visão de time autogerenciável. O outro respondente que não marcou esta opção é *Scrum Master* há menos de um ano e exerce a função de *designer*. Vale ressaltar que estes dois respondentes não têm uma visão de liderança, logo, eles terão mais dificuldades em ter uma visão de *Scrum Master*.

4.1.3 Parte III - Identificação das ferramentas e funcionalidades utilizadas pelos Scrum Masters

Esta parte tem como objetivo a identificação das ferramentas utilizadas pelos *Scrum Masters*, obter informações relacionadas às ferramentas usualmente utilizadas por eles para auxiliar a execução de suas atividades, bem como verificar se essas atendem às expectativas desses profissionais.

Questão 7 - Para auxiliar no cumprimento de suas atividades, quais as ferramentas **não-computacionais** que VOCÊ REALMENTE UTILIZA? (Escolher mais de uma opção)

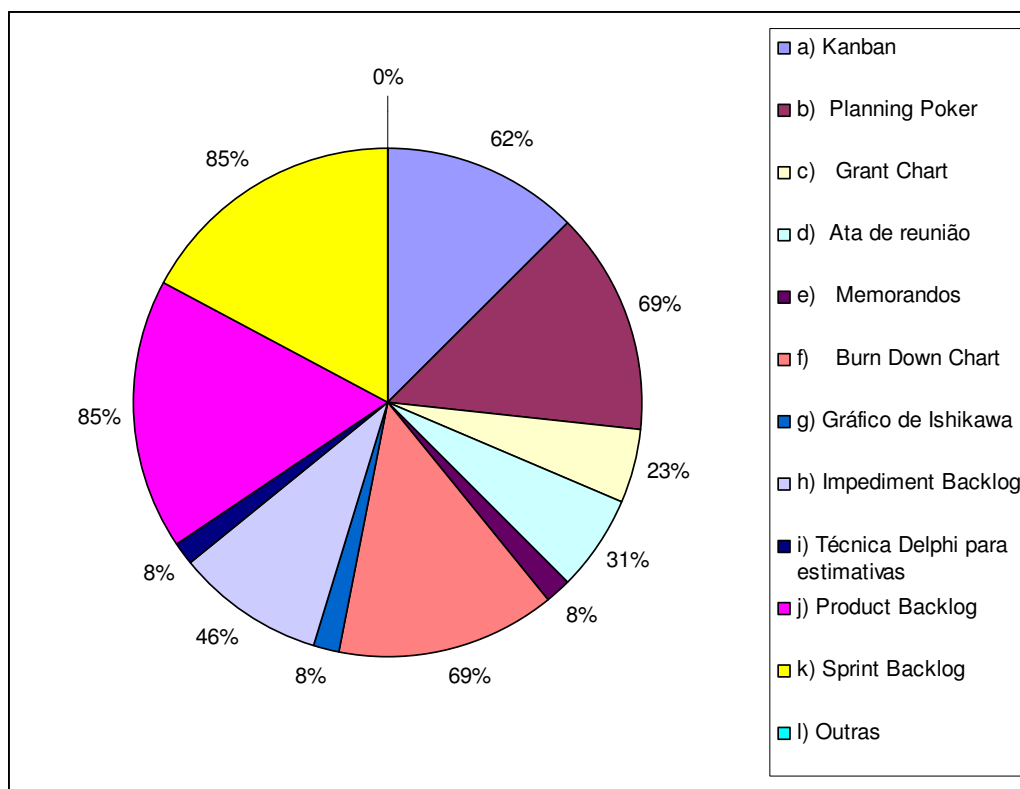


Gráfico 9: Ferramentas não-computacionais utilizadas
 Fonte: Elaboração própria, com dados da pesquisa de campo.

Visualiza-se no Gráfico 09 que 85% (N=11) dos respondentes utilizam *Product Backlog* como ferramenta não computacional; 85% (N=11) usam *Sprint Backlog*; 69 % (N=9) usam *Planning Poker*; 69% (N=9) utilizam *Burn Down Chart*; 62% (N=8) fazem uso do *Kanban*; 46% (N=6) utilizam *Impediment Backlog*; 31% (N=4) ata de reunião; 23% (N=3) usam *Grant Chart*; 8% (N=1) utilizam *Memorandos*; 8% (N=1) usam *Gráfico de Ishikawa*; 8% (N=1) utilizam a *Técnica Delphi* para estimativas e o item *Outras* não obteve respostas.

Analisando os dados coletados, se observou que os respondentes na sua maioria marcou as ferramentas *Product Backlog* e *Sprint Backlog*, as quais podem ser consideradas, segundo a literatura, essenciais para o desenvolvimento de um projeto utilizando *Scrum*. Destaca-se que apenas duas pessoas não acham necessárias essas duas ferramentas, sendo uma delas o respondente de menor experiência na pesquisa e a segunda exerce a função de líder de implantação, atuando muito mais nesse papel do que como *Scrum Master*.

Somando-se a isso, ressalta-se também o uso do *Burn Down Chart* por 69% dos respondentes, a qual também é uma ferramenta bem aceita pela metodologia *Scrum*.

Também se pode observar na pesquisa realizada, o emprego de ferramentas complementares, como a Técnica *Delphi* para estimativas e *Planning Poker*, as quais são bastante parecidas e empregadas para fazer estimativas. A Técnica *Delphi* trata-se da reunião do grupo do projeto que faz uma estimativa com base em sua experiência, o resultado é tabulado e discutido, depois é feita nova estimativa, tabulada e discutida, assim repete-se até não haver mais evolução e tira-se a média para a estimativa. O *Planning Poker* será explicado na Seção 4.2.1.2.

Questão 8 - Dentre as ferramentas não-computacionais que você utiliza, alguma PODERIA SER DESCARTADA? (Pode assinalar mais de uma opção).

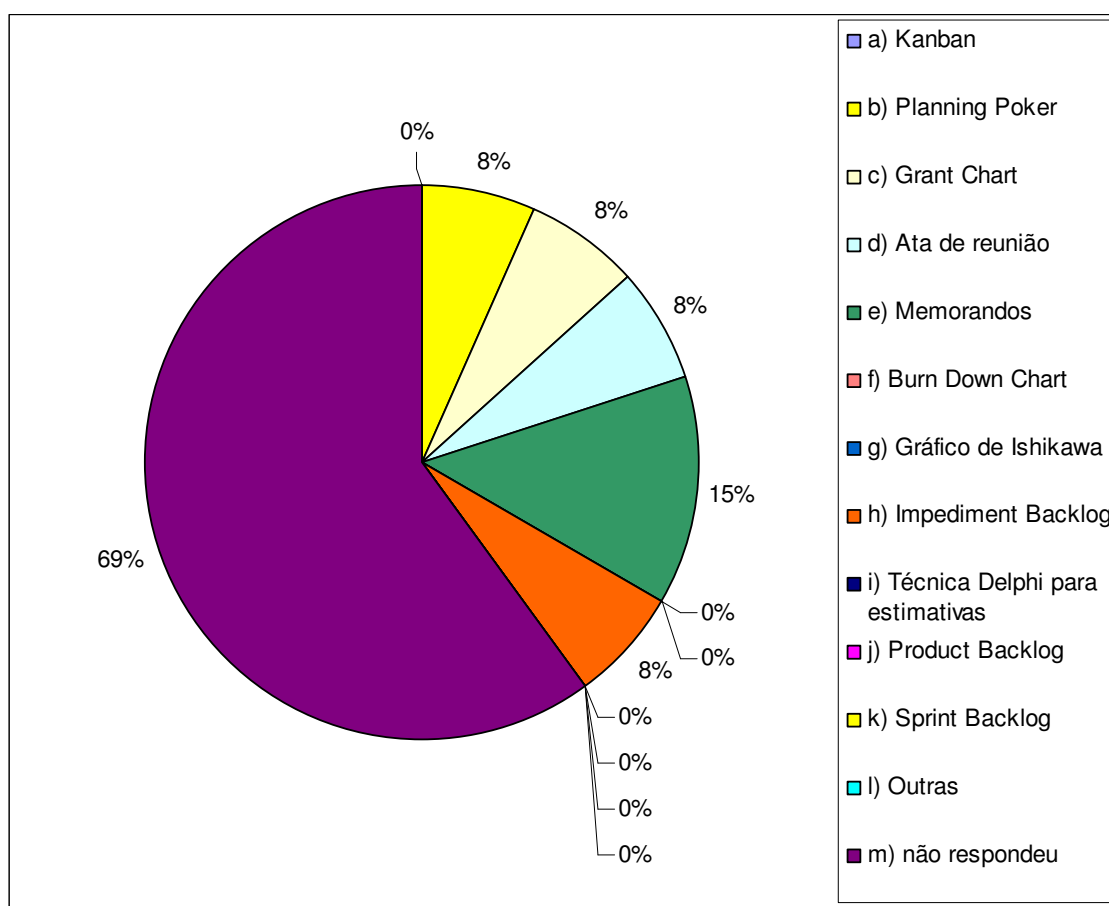


Gráfico 10: Ferramentas não-computacionais que poderiam ser descartadas

Fonte: Elaboração própria, com dados da pesquisa de campo.

Na questão concernente às ferramentas não computacionais que poderiam ser descartadas apesar de sua utilização pelos participantes da pesquisa, observa-se no Gráfico 10 que as respostas foram: 69% (N=9) dos questionados não responderam; 15% (N=2) disseram que poderiam descartar a ferramenta Memorando; 8% (N=1) *Grant Chart*; 8% (N=1)

descartariam Ata de reunião; 8% (N=1) descartariam *Impediment Backlog*; 8% (N=1) disseram que descartariam *Planning Poker*, porém este respondente na Questão 7 marcou que utiliza somente as ferramentas *Product Backlog* e *Sprint Backlog*. E um dos respondentes que afirmou descartar a ferramenta memorando, foi o mesmo que descartaria as ferramentas *Grant Chart* e Ata de reunião também na questão 07 afirmou não utilizar destas ferramentas no seu dia a dia como *Scrum Master*.

Dentre as ferramentas listadas como utilizadas pelos respondentes, o *Product Backlog*; *Sprint Backlog*; *Burn Down Chart*; *Kanban*, Gráfico de *Ishikawa* e Técnica *Delphi* na opinião dos respondentes não poderiam deixar de ser citadas, e a opção Outras não obteve respostas.

Analisando os dados da Questão 08, pode-se observar que a maior parte dos respondentes optou por ferramentas que não são tipicamente sugeridas pela metodologia *Scrum*. Talvez por este motivo fossem avaliadas como dispensáveis para a execução das atividades de *Scrum Master*.

Questão 9 - Existem ferramentas não-computacionais que você não utiliza atualmente, mas que gostaria de fazer uso? Qual (is)?

As respostas foram de acordo com a manifestação individual dos participantes da pesquisa. Desta forma, obteve-se o seguinte resultado: N=7 não responderam; N=1 respondeu que gostaria de usar Seis Chapéus do Pensamento; N=1 que gostaria de usar *Burn Down Chart*; N=1 Quadro de *Kanban* e N=1 Avaliação 360° com a equipe. Um dos respondentes N=1 informou não ter desejo de utilizar outra ferramenta. Outro respondente N=1 respondeu a esta questão afirmando que “as ferramentas utilizadas atualmente atendem às necessidades do porte dos nossos projetos”.

As pessoas que responderam que gostariam de utilizar *Burn Down Chart* e Quadro de *Kanban* e que não marcaram esta opção na questão 08, provavelmente trabalham em empresas que não aderiram a essas ferramentas.

A ferramenta Seis Chapéus do Pensamento, citada por um respondente como sendo uma ferramenta que gostaria de utilizar, trata-se de uma ferramenta de análise de idéias simples e eficiente, pois permite empregar um estilo de interação construtiva, onde simplifica o pensamento, enfocando o tema sob perspectivas específicas, com clareza e sem confrontos. Por exemplo, existe a sessão de identificação de riscos (chapéu preto), a de gerar idéias (chapéu verde), a de apurar informações (chapéu branco), a de expor emoções (chapéu

vermelho), a de buscar uma visão positiva (chapéu amarelo) e a de ordenar a própria reunião (chapéu azul), segundo Bono (2008). É uma ferramenta muito interessante para se aplicar nas reuniões do *Scrum*.

Em relação à Avaliação de 360° sugerida por um respondente, trata-se de um método de avaliação de pessoas, onde o profissional é avaliado não somente pelo seu superior, mas também pelos seus pares e eventuais subordinados. É uma técnica bastante interessante em equipes *Scrum*, pois força a interação entre os membros do projeto e forçando que cada um, dê o melhor de si.

Questão 10 - Quais as ferramentas computacionais que VOCÊ REALMENTE UTILIZA em suas atividades de *Scrum Master*? (pode escolher mais de uma opção)

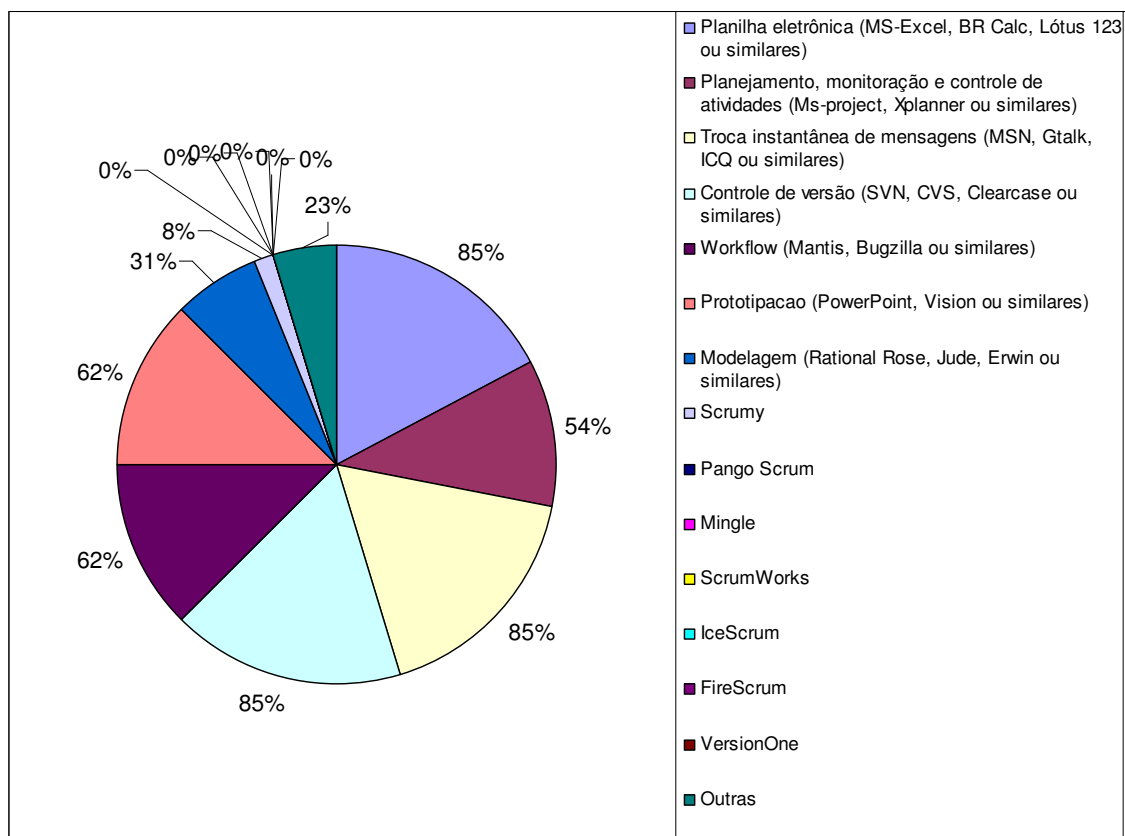


Gráfico 11: Ferramentas computacionais utilizadas
 Fonte: Elaboração própria, com dados da pesquisa de campo.

Em relação à frequência de uso das ferramentas computacionais, obteve-se o resultado, segundo observado no Gráfico 11: 85% (N=11) responderam que usam com frequência a Planilha eletrônica (*MS-Excel*, *BR Calc*, *Lótus* ou similares); 85% (N=11) disseram usar Troca instantânea de mensagens (*MSN*, *Gtalk*, *ICQ* ou similares); 85% (N=11)

utilizam o Controle de versão (SVN, CVS, *Clearcase* ou similares); 62% (N=8) utilizam *Workflow* (*Mantis*, *Bugzilla* ou similares); 62% (N=8) utilizam Prototipação (*Power Point*, *Vision* ou similares); 54% (N=7) utilizam Planejamento, monitoração e controle de atividades (*MS-Project*, *Xplanner* ou similares); 31% (N=4) utilizam Modelagem (*Rational Rose*, *Jude*, *Erwin* ou similares) e 8% (N=1) utilizam *Scrumy*. Dentre os 23% (N=3) que escolheram a opção Outras ferramentas, N=1 afirmou utilizar *ClockingIT*; N=1 afirma usar *Clockingit* e *PlanningPoker* e N=1 respondeu *Scrumability*. Os itens *Pango Scrum*, *Mingle*, *ScrumWorks*, *IceScrum*, *FireScrum* e *VersionOne* constantes no questionário não foram respondidos.

Dentre as ferramentas mais utilizadas está a planilha eletrônica, controle de versão e troca instantânea de mensagem. As duas primeiras são empregadas no controle do projeto e a última para comunicação constante.

Esses resultados demonstram a aderência dos mesmos com as respostas obtidas na Questão 06, em relação às atividades desempenhadas no dia-a-dia dos respondentes.

Além disso, planejamento, monitoramento e controle de atividades são essenciais para um *Scrum Master*, pois mesmo não devendo atuar como um gerente de projeto cabe a ele saber o que está acontecendo no projeto para poder ajudar a equipe no que for preciso a qualquer momento.

Em relação às ferramentas de *workflow* (*Mantis*, *Bugzilla* ou similares) são usadas pelos respondentes para cadastrar e acompanhar os erros encontrados pela equipe de testes, para registrar as melhorias solicitadas pelos clientes ou para registrar e acompanhar as não conformidades dos processos de auditoria interna.

Quanto às ferramentas de prototipação e modelagem (*Power Point*, *Vision* ou similares) servem para apoiar o levantamento e validação de requisitos junto ao cliente e *Product Owner*. Com estas técnicas, o *Scrum Master* pode auxiliar o PO a compreender as reais necessidades do cliente e, assim, tornar o software o mais funcional possível.

Por outro lado, as ferramentas *Scrumy*, *ClockingIT*, *Planningpoker* e *Scrumability*, embora sejam específicas para a metodologia *Scrum*, criadas para facilitar no desenvolvimento de projetos *Scrum*, não são muito empregadas pelos respondentes. Estas serão discutidas na Seção 4.2.3.

Questão 11 - Dentre as ferramentas computacionais que você utiliza, alguma PODERIA SER DESCARTADA?

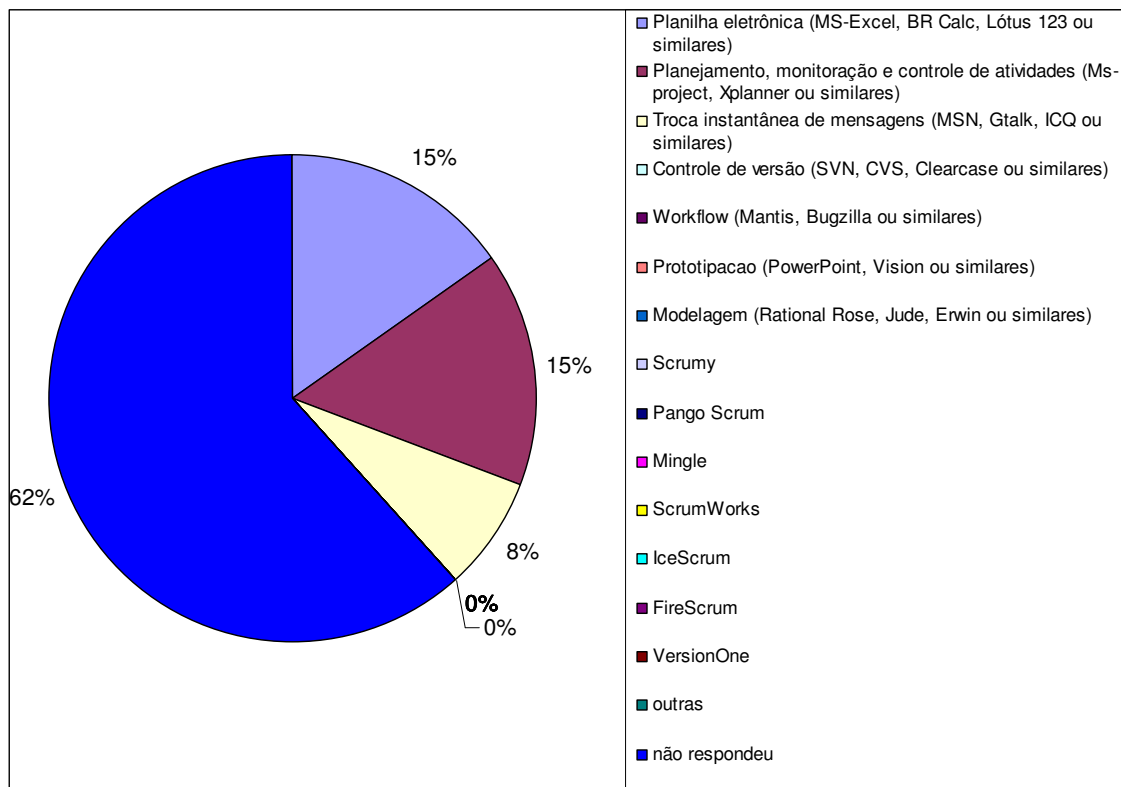


Gráfico 12: Ferramenta computacional que poderia ser descartada

Fonte: Elaboração própria, com dados da pesquisa de campo.

Neste item, conforme demonstrado no Gráfico 12, os respondentes disseram que descartariam as seguintes ferramentas: 15% (N=2) Planilha eletrônica (*MS-Excel*, *Br Cac*, *Lótus 123* ou similares); 15% (N=2) Planejamento, monitoração e controle de atividades (*MS-Project*, *Xplanner* ou similares); 8% (N=1) Troca instantânea de mensagens (*MSN*, *Gtalk*, *ICQ* ou similares). Os demais itens listados pela investigadora: Controle de versão (*SVN*, *CVS*, *Clearcase* ou similares); *Workflow* (*Mantis*, *Bugzilla* ou similares); *Prototipacao* (*PowerPoint*, *Vision* ou similares); *Modelagem* (*Rational Rose*, *Jude*, *Erwin* ou similares); *Scrumy*; *Pango Scrum*; *VersionOne* e *Outras* não foram assinalados. Destaca-se que 62% (N=8) dos questionados não responderam esta questão.

Dentre os respondentes que descartariam a planilha eletrônica, um utiliza a ferramenta *Scrumability*, que é uma ferramenta desenvolvida pela própria empresa e não comercializada. E o outro utiliza uma ferramenta de planejamento, monitoração e controle de atividades as quais são usadas de forma similar à planilha. Ou seja, ele descarta a ferramenta, pois utiliza outras que fazem a mesma coisa, porém de maneira mais organizada.

Questão 12 - Existem ferramentas computacionais que você não utiliza atualmente, mas que gostaria de fazer uso? Qual (is)?

Neste item, as perguntas foram respondidas de acordo com a preferência dos participantes da pesquisa. Assim, a preferência dos participantes ficou em: 31% N=4 para *FireScrum*; 8% N=1 respondeu *Scrum Works*; 8% N=1 *Office* e *MS-Project*; 8% N=1 gostaria de usar *Target Process* e *Rally* e justificou o seu possível uso por serem ferramentas comerciais; 8% N=1 respondeu que não utilizaria nenhuma outra ferramenta; 37% N=5 não responderam, um (01) destes respondentes justificou dizendo que “algumas ferramentas não estão sendo utilizadas por desconhecemos suas funcionalidades”.

As ferramentas *Pago Scrum*, *Mingle*, *Scrum Works*, *IceScrum*, *FireScrum* e *VersionOne* não foram citadas nas respostas da Questão 10, embora sejam aderentes à metodologia *Scrum*, assim como as ferramentas *Target Process* e *Rally*.

Além disso, embora o *Office* e o *MS-Project* não tenham sido desenvolvidos para atender às propostas do *Scrum*, foram mencionadas nessa questão por um dos respondentes.

As ferramentas *FireScrum*, *Scrum Works*, *Target Process* e *Rally* serão abordadas na Seção 4.2.3

Questão 13 - Dentre as funcionalidades que você utiliza nas ferramentas escolhidas na questão 10, qual (is) você gostaria de ter numa única ferramenta integrada?

Com relação a esta pergunta, N=1 respondeu que gostaria de ter planejamento, monitoração e controle de atividades (*Ms-project*), *Scrum Works*; N=1 respondeu planilha, comunicação, *kanban*, controle de versão, gráficos de acompanhamento; N=1 respondeu que gostaria que a ferramenta *Project* tivesse alguma funcionalidade do quadro de *Kanban*; N=1 todas as ferramentas de gestão, comunicação, documentação e planejamento; N=1 respondeu Planilha + *MS-Project* com controle de versão (SVN); N=1 *Product backlog*; *Sprint backlog*; Controle das estimativas para cada item do *sprint backlog*; *Burn down do sprint* e do projeto como um todo, cronograma do *sprint*, com acompanhamento do planejado x realizado (destacando quantas horas o time está atrasado para que o time possa tomar ações estratégicas para recuperação); Lista de impedimentos: de forma que o time possa indicar qual o impedimento das tarefas planejadas; Lista de riscos. Além dos dados elencados no questionário, os participantes da pesquisa contribuíram com: N=1 disse que nenhuma ferramenta, pois o *FireScrum* fornece o que ele precisa; N=1 respondeu que gostaria o

ClockingIt, “pois é uma ferramenta *openscore* fácil de ser customizada”; N=1 respondeu que gostaria do *FireScrum*; N=4 não responderam, mas um (01) dos respondentes justificou dizendo que “uso a planilha Excel para gerar um burn down, pois a ferramenta *Scrumability* irá gerar burn down e poderei descartar a planilha *Excel*”; outro respondente que também não respondeu à pergunta, justificou que gostaria de ter “um modo integrado às ferramentas não computacionais sem perder o aspecto visual e do todo”.

4.1.4 Parte IV - Identificação das funcionalidades de uma ferramenta computacional para auxiliar os Scrum Masters

O principal objetivo dessa parte é identificar as funcionalidades desejadas pelos respondentes em uma única ferramenta.

QUESTÃO 14 – Para auxiliar na análise das respostas assinale o nível de prioridades destas funcionalidades em: ES – Essencial; IM - Importante e DS – Desejável.

Neste item foi pedido aos respondentes que assinalassem nos títulos elencados ES para essencial; IM para importante e DS para desejável, conforme o grau de importância de cada um.

As funcionalidades sugeridas que eram equivalentes foram agrupadas, para que a análise dos dados fosse mais significativa, assim foram considerados mais vinte e três (23) títulos como sugestão que são apresentados no Quadro 4. Dos 13 participantes da pesquisa, N=3 não responderam esta questão.

Quadro 4: Funcionalidades

<i>Funcionalidade</i>	<i>Nome Funcionalidade</i>	<i>ES</i>	<i>IM</i>	<i>DS</i>
Func.1	Ambiente de simulação do planning poker		2	5
Func.2	Cálculo do ritmo sugerido (em pontos) para o time fechar o projeto com sucesso	1		
Func.3	Calculo do tempo total do projeto	2	2	2
Func.4	Cálculo e Gráfico do desvio do projeto	2		
Func.5	Controle do status dos requisitos e das user stories (não iniciado, em desenvolvimento, desenvolvido)		1	
Func.6	Cronograma do sprint (com as tarefas listadas, estimativa para cada tarefa, distribuição das tarefas por data, status de cada tarefa e possibilidade de identificação das tarefas)	1		
Func.7	Distribuição de atividades para os componentes da equipe	3	3	2
Func.8	Geração do burn down chart	6	2	
Func.9	Gerenciamento de risco (do projeto e do sprint)	2	2	
Func.10	Impediment Backlog	2	2	1
Func.11	Integração com ferramenta de registro de bugs/problemas			1
Func.12	Kanban/ taskboard	2	2	3
Func.13	Matriz de relacionamento Requisito x User story		1	
Func.14	Planejamento e controle dos Release (Roadmap dos sprints)	1		1
Func.15	Product Backlog	7		
Func.16	Registro das lições aprendidas (acompanhamento)			1
Func.17	Registro do tempo gasto e restante para cada tarefa		1	
Func.18	Registros de eventos do sprint (cancelamento do sprint, mudança de membros, ...)			1
Func.19	Relatórios de desempenho do time (por membro, por sprint,...)			1
Func.20	Relatórios do projeto e atas		1	
Func.21	Sprint Backlog	5		
Func.22	Troca instantânea de mensagens (ferramenta de comunicação)			2
Func.23	Velocidade de equipe (Estimado x Trabalhado)		1	1

Fonte: Elaboração própria, com dados da pesquisa de campo.

Houve um respondente que colocou como funcionalidade: estar com o *product owner* mais presente, reuniões diárias, reunião de retrospectiva, reunião de revisão e ter uma arquitetura já bem definida. Outro respondente colocou repositório de materiais para desenvolvimento. Estas “funcionalidades” foram descartadas, pois não se trata de uma

funcionalidade que possa ser inserida em um sistema, mas sim, em ações que devem ser adotadas na metodologia de um projeto *Scrum*.

Houve respostas com mais de uma funcionalidade em um único item, as quais foram desmembradas, por exemplo: Planejamento, monitoração e controle de atividades (*product backlog*, *impediment backlog*, *sprint backlog*, etc...). Esta resposta foi dividida em: *product backlog*, *impediment backlog*, *sprint backlog*.

Por outro lado, funcionalidades muito específicas foram agrupadas em itens macros, por exemplo, menciona-se: geração e documentação do *product backlog*; ferramenta de pesos dos itens para o *product backlog*; manutenção e priorização do *product backlog* e gerenciamento PBI (priorizados, entregues, não priorizados) foram agrupadas na funcionalidade macro *product backlog*.

A geração do *burn down chart* refere-se tanto ao do projeto como do *sprint*, e esses gráficos devem ser atualizados diariamente, medindo as funcionalidades planejadas para o projeto/ *sprint* para verificar quantas funcionalidades ainda precisam ser implementadas.

4.2 Análise comparativa das ferramentas

Para o desenvolvimento de seu papel, os *Scrum Masters* podem utilizar várias ferramentas para auxiliá-los na realização e controle de suas atividades. Neste trabalho, essas ferramentas serão divididas em dois grupos: ferramentas não computacionais e ferramentas computacionais. Nas seções seguintes são abordadas algumas dessas ferramentas e é feita uma análise comparativa entre as mesmas.

Para a realização dessa análise, foi traçado um comparativo entre as funcionalidades descritas nas questões 13 e 14 do questionário (Apêndice A) com as funcionalidades das ferramentas existentes no mercado e as que foram citadas pelos respondentes nas questões 7, 9, 10 e 12 do questionário (Apêndice A), com o objetivo de verificar no mercado se existem ou não alguma ferramenta que tenha as funcionalidades desejadas pelos respondentes.

4.2.1 Ferramentas não computacionais

Nas questões 7 e 9 do questionário foram citadas doze (12) ferramentas não computacionais que são: *Kanban*, *Planning Poker*, Ata de reunião, Memorandos, *Burn Down*

Chart, Gráfico de *Ishikawa*, *Impediment Backlog*, Técnica *Delphi* para estimativas, *Product Backlog*, *Sprint Backlog*, Seis Chapéis do Pensamento e Avaliação 360°.

Porém, nesta seção, serão descritas apenas as três (03) ferramentas de maior incidência entre os respondentes, que são o Quadro de *Kanban*, *Planning Poker* e *Burn Down Chart*. A escolha destas três (03) ferramentas também se dá por estas serem as ferramentas propostas pela metodologia para ajudar no desenvolvimento de projetos que utilizam o *Scrum*.

Em relação às ferramentas *Impediment Backlog*, *Product Backlog* e *Sprint Backlog*, já foram abordadas na seção de 2.2.3, logo não serão mencionadas nesta seção.

Vale ressaltar que estas ferramentas não são de uso obrigatório e sim complementar para o controle de um projeto *Scrum*, uma vez que auxiliam toda a equipe durante o desenvolvimento do projeto.

4.2.1.1 Quadro de Kanban

Kanban é uma palavra de origem japonesa, que significa registro ou placa visível. Para Moura (*apud* Almeida, 2007), *Kanban* é uma técnica de gestão de materiais e de produção, sendo controlada através do movimento de um cartão.

O *Scrum* propõe a utilização do quadro de *Kanban*, ao qual, segundo Farias Jr. (2008), ajuda no acompanhamento do andamento de uma iteração do projeto, chamada de *sprint*.

Rahal Junior (2008) afirma que o quadro *Kanban* é importante em um projeto *Scrum*, pois viabiliza a comunicação e a visibilidade entre os integrantes da equipe, além de permitir o acompanhamento da construção de um Item de *Backlog*, onde cada fase da construção de uma peça, software ou qualquer projeto, que faça parte do Item de *Backlog* pode ser rapidamente identificado pela equipe.

Para Ortiz (2009) é no *Kanban* que todas as histórias e tarefas são colocadas, assim toda equipe de desenvolvimento, o *Scrum Master* e o PO têm informações sobre o andamento do *sprint*. No quadro, também, são colocadas as informações como impedimentos e tarefas feitas fora do *sprint*.

Esse quadro tem a finalidade de ilustrar de forma clara e objetiva como está fluindo o trabalho da equipe, destacando, as atividades que ainda não iniciaram, aquelas que estão em execução, as que já foram concluídas, conforme a Figura 4. Com isto, é possível avaliar o que está sendo feito e o que falta para a meta ser concluída.

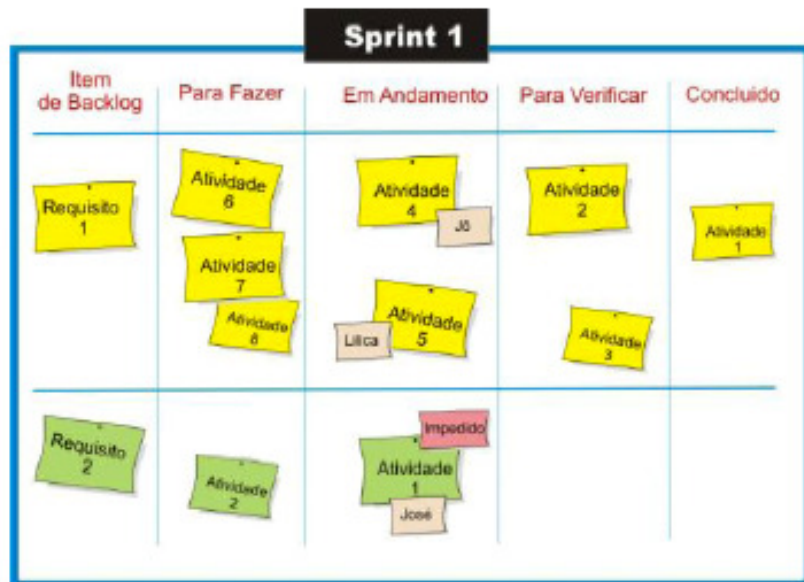


Figura 4: Quadro de Kanban

Fonte: Peres (2008)

4.2.1.2 Planning Poker

Antes de iniciar cada *sprint*, o time deve estimar os itens a serem desenvolvidos. Essa estimativa deve ser feita durante uma reunião, na qual o *Scrum Master* atua como um mediador.

Segundo Magno (2009), existem diversas técnicas para auxiliar nas estimativas, tais como: *Kano*, *Theme Screening* e *Técnica Delphi*. Além desses, pode-se mencionar ainda o *Planning Poker*, o qual vem obtendo grande popularidade entre os Times *Scrum*.

Esta técnica, conforme Santos (2008), foi descrita inicialmente por James Grenning em 2002 e popularizada por Mike Cohn, no seu livro *Agile Estimating and Planning*.

Segundo Mueller (2007), *Planning Poker* é uma prática que auxilia na estimativa de histórias e tarefas a serem feitas no *sprint*. Para Branas (2009), ferramentas como o *Planning Poker* são alternativas muito interessantes para prender a atenção do time e motivar todos a pensarem e participarem ativamente do processo de estimativa de cada história.

Destaca-se que essa técnica coloca todos os participantes no mesmo nível, evitando que os menos experientes aceitem passivamente a decisão dos demais membros.

Ortiz (2009) comenta que o *Planning Poker* consiste em pontuar as histórias e tarefas do projeto através da utilização de um baralho especial. Essa técnica consiste em um jogo de cartas, no qual cada carta corresponde a uma numeração da série de Fibonacci (1, 2, 3, 5, 8, 13, 21...).

Assim, baseando nessa numeração, cada membro do Time, recebem suas cartas e todos juntos votam pela pontuação. Caso haja divergências, cada participante tem o direito de expor e defender sua opinião. Após exposto os pontos de vista, deverá haver uma nova votação. Este ciclo deve se repetir no máximo três vezes. Caso a equipe não chegue a um consenso, deverá ser atribuída para a estória e/ou tarefa em votação o valor da carta de maior pontuação na última rodada.

Para Magno (2009), o *Planning Poker* funciona adequadamente, pois durante o “jogo”, múltiplas opiniões de especialistas são apresentadas e discutidas e, como o *Scrum* trabalha com Times multi-funcionais, é sempre importante haver um consenso em relação às estimativas, conforme mostra a Figura 5.

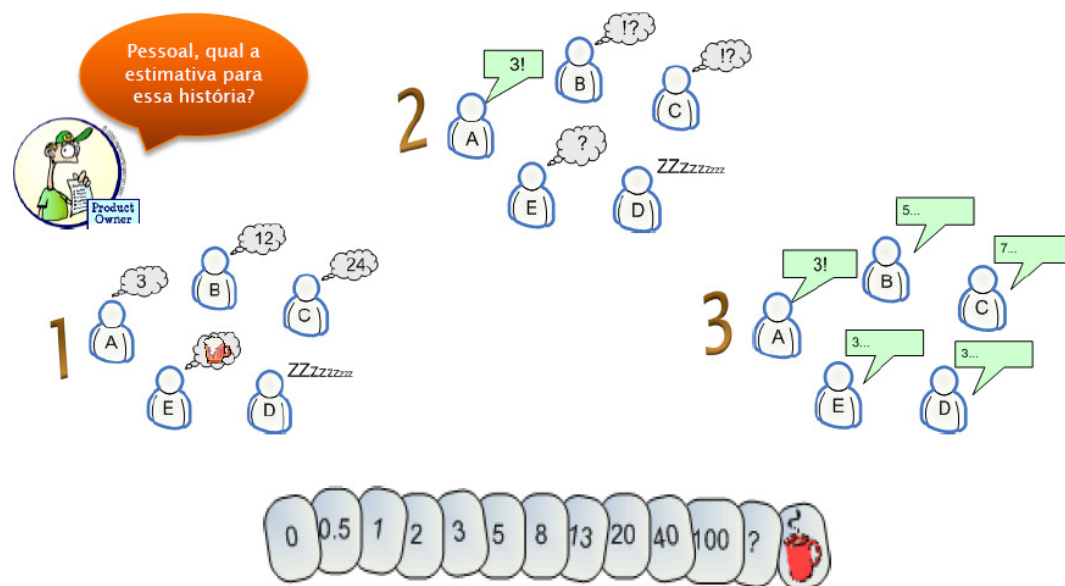


Figura 5: Estimando o Product Backlog com Planning Poker

Fonte: Pereira (2008)

4.2.1.3 Burn Down Chart

Segundo Brod (2007), um novo projeto começa com a estimativa das estórias e estas são inseridas no *Product Backlog* e no *Burn down Chart*. Ainda segundo o autor, o último é um gráfico que tem simplesmente a função de listar as horas a serem consumidas pelos recursos. Assim, o ideal é que as horas previstas para as atividades planejadas sejam iguais às horas realmente consumidas.

Velamakanni e Sathianarayanan (2009) acreditam que o gráfico *Burndown* é um presente valioso para as equipes do mundo ágil e que se trata atualmente de um artefato

amplamente utilizado nas iniciativas *Scrum*. O autor afirma que o *Burn down* representa graficamente a quantidade de esforço que falta para a conclusão das tarefas estipuladas. E que quando este gráfico é atualizado diariamente, fornece informações sobre as estimativas e sua precisão, produtividade, controle de processo e muito mais, bastando ser feita uma boa análise do mesmo.

Castro (2009) afirma que o gráfico de *Burn down* é uma forma visual e rápida de enxergar o status atual do projeto. Ele possui uma estrutura simples, onde:

- Eixo X - representa os dias do *sprint*;
- Eixo Y - representa o trabalho restante.

O trabalho restante pode ser definido de acordo com a sua necessidade. Algumas pessoas utilizam pontos, algumas pessoas utilizam horas, outros utilizam dias e assim por diante. O uso com pontos geralmente ocorre com a atualização do gráfico apenas quando uma *user story* é finalizada.

Conforme Rahal Junior (2007), para melhor acompanhamento do progresso de cada *sprint* e, conseqüentemente, do projeto, é comum a utilização de dois tipos de gráficos chamados *Sprint Burn down* e *Product Burn down*, conforme ilustrado na Figura 6.

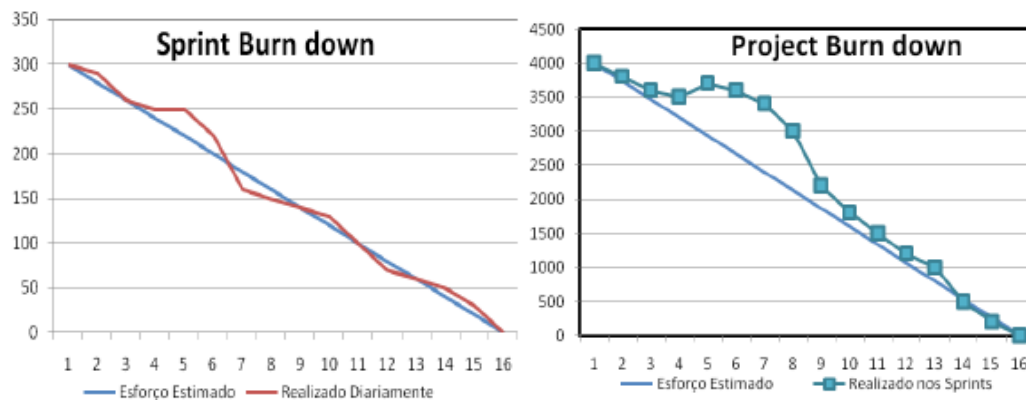


Figura 6: Sprint e Project Burn Down

Fonte: Magno (2009)

O *Sprint Burn down* tem por objetivo apresentar o acompanhamento diário das execuções das tarefas a serem realizadas em um *sprint*. Enquanto que o *Product Burn down* representa a execução das tarefas existentes em todos os *sprints*, com isso, obtém uma visão do projeto por inteiro.

Com base nesses dois gráficos, tanto o *Scrum Master* quanto o time podem avaliar facilmente se o *sprint* terá sucesso ou não com o ritmo atual e, assim, traçar estratégias para que o quadro seja revertido, caso haja necessidade.

4.2.2 Comparação entre as ferramentas não computacionais

Estas três (03) ferramentas abordadas anteriormente são as que mais foram mencionadas pelos entrevistados durante a pesquisa na questão 7 do questionário (Apêndice A). Vale ressaltar também que é desejo da maioria dos respondentes que essas ferramentas não computacionais fossem integradas a uma ferramenta computacional para melhor aplicá-la no ambiente dinâmico de um projeto *Scrum*.

Principalmente se tratando de um *Scrum Master*, duas dessas ferramentas (Quadro de *Kanban* e *Burn Down Chart*) são muito importantes para verificar o andamento do projeto. Além disso, o *Planning Poker* surge como uma alternativa divertida, porém bastante profissional, para o time estimar suas atividades.

Vale ressaltar que essas três (03) ferramentas oferecem recursos que não são excludentes entre si e que se utilizadas em conjunto, tendem a proporcionar melhorias no gerenciamento do projeto.

No mercado já existem versões computacionais destas ferramentas, que surgiram com o objetivo de aumentar a produtividade e consolidar informações que são perdidas quando feitas manualmente, além de tentar facilitar ainda mais o processo de gerenciamento ágil. Na próxima seção, serão abordadas algumas ferramentas computacionais mais conhecidas atualmente e que foram citadas pelos respondentes no questionário.

4.2.3 Ferramentas computacionais

Nesta seção serão abordadas ferramentas computacionais encontradas no mercado para auxiliar no desenvolvimento de projetos que utilizam *Scrum*.

Essas ferramentas computacionais ajudam as pessoas que trabalham com *Scrum* a planejar iterações, acompanhar suas atualizações e, de um modo geral, tornam a execução das atividades diárias de um *Scrum Master* menos complicadas e mais ágeis.

Stevens (2008) apresentou em seu blog *Scrum Breakfast*, uma pesquisa mostrando como se encontrava o uso de ferramentas dedicadas à gestão ágil. O resultado desta pesquisa é apresentado no Gráfico 13.

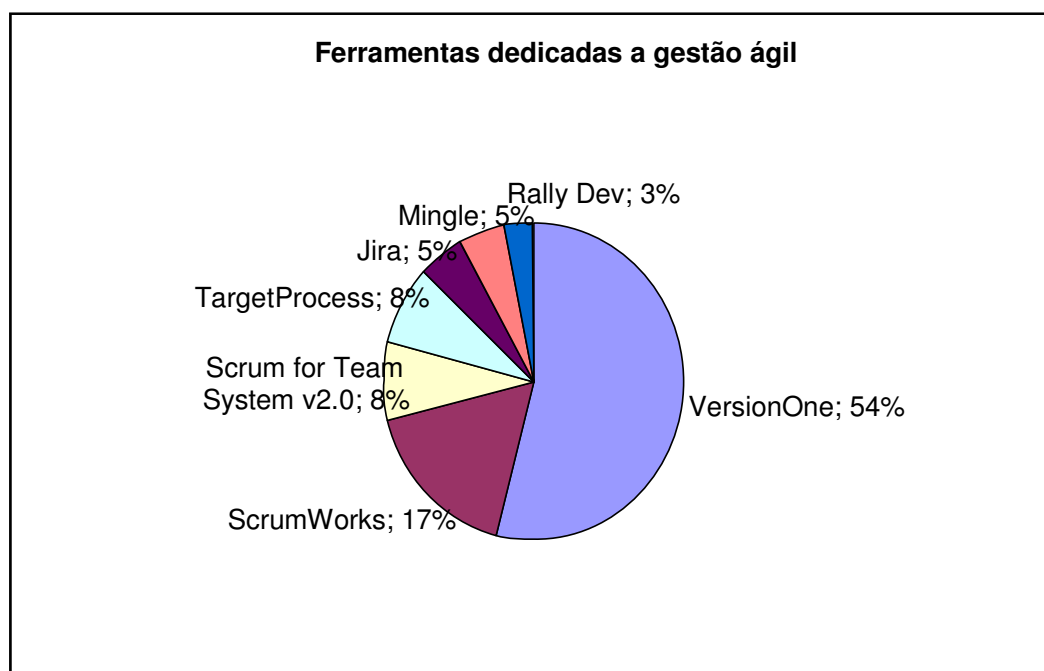


Gráfico 13: Ferramentas dedicadas a gestão ágil

Fonte Adaptada: Stevens (2008)

Contrariando os resultados de Stevens, *VersionOne* (2009) afirma que as ferramentas mais utilizadas em projetos ágeis são: o *Excel* (63%), *Microsoft Project* (33%), *Jira* (24%), *Próprias* (22%), *VersionOne* (18%), *Microsoft TFS* (16%), *Google Docs* (16%), *Bugzilla* (14%), *Rational* (10%), *Xplanner* (5%), *Mingle* (5%), *Primavera* (3%), *Target Process* (2%) e 24% responderam a opção *Outras*. Vale ressaltar que os respondentes podiam marcar múltiplas ferramentas.

Dentre as ferramentas mais utilizadas, conforme Stevens (2008), somente *ScrumWorks*, *Target Process* e *Rally* foram mencionadas entre os respondentes dessa pesquisa. E as ferramentas mais utilizadas conforme *VersionOne* (2009) também foram mencionadas pelos entrevistados na entrevista dessa pesquisa que são: o *Excel*, *Microsoft Project*, *Bugzilla*, *Rational*, e *Xplanner* que é muito parecido com o *Microsoft Project* porém ele é *on-line*.

A seguir será feita uma breve descrição de sete (07) ferramentas computacionais mencionadas pelos respondentes nas questões 10, 12 e 13 do questionário (Apêndice A), que são: *Scrumy*, *Clocking it*, *Planningpoker*, *ScrumWorks*, *FireScrum*, *TargetProcess* e *Rally*. E, adicionalmente, a *VersionOne* por ser a mais utilizada no mercado. Logo após, é apresentada a análise comparativa entre essas ferramentas.

A ferramenta *Scrumability* também foi mencionada por um respondente, porém, trata-se de uma ferramenta própria que está em processo de desenvolvimento. *Scrumability* não está disponível para análise, compra e nem uso, logo, não pôde ser feito a sua análise.

A avaliação dessas sete (07) ferramentas citadas pelos respondentes e mais a *VersionOne* que é a ferramenta mais utilizada no mercado foi feita através do uso das versões *free* e o levantamento dos recursos juntos aos fabricantes. Também foram levadas em consideração as declarações de alguns usuários destas ferramentas. Para um melhor detalhamento, foram estabelecidos critérios de avaliação, os quais são: características funcionais e licença.

4.2.3.1 Scrumy

Scrumy é uma ferramenta *web*, simples de usar e sem instalação, basta ter um navegador *web*, informar as credenciais e usar, segundo seu fabricante [<http://scrumy.com/about>] é uma maneira simples e intuitiva para fazer gestão de projetos *Scrum*. Basicamente provê a criação das *user stories* e o acompanhamento das mesmas por meio do painel de tarefas chamado de *taskboard* virtual. Existe a versão Pro que permite gerenciar múltiplos *sprints* e *backlogs*, possui ainda a opção de visualizar os projetos passados e criar *Burn down chats*. Além disso, o *Scrumy* é atualizado automaticamente e todos os membros da equipe têm uma visão real do projeto.

Existe duas licenças, a versão básica do *Scrumy* é gratuita e a versão *Scrumy* Pro custa US\$ 7 por mês ou US\$60 por ano.

4.2.3.2 ClockingIT

ClockingIT é uma ferramenta online bem intuitiva que possui um design gráfico simples, permite rastrear as tarefas, projetos e tempo gasto, permite ainda o cadastro de diversos projetos e colaboradores, onde em cada projeto é possível criar várias tarefas com diversos graus de prioridade, nível de dificuldade, tempo estimado de execução e data de entrega, além de poder designar um colaborador para a execução de tal tarefa. Todas as ações efetuadas por algum usuário são vistas em tempo real pelos outros usuários conectados ao mesmo projeto. A ferramenta possui lista de tarefas, o Gráfico de *Gantt*, relatórios flexíveis, chat embutido, oferece integração com *Google calender* ou o calendário da Apple, tem

sistema de RSS, um *wiki*, fórum e um *seedbar* o qual consiste em que o usuário marque o início e fim de uma tarefa que está realizando em tempo real.

Sua licença é *free*, basta apenas fazer um cadastro no site do fabricante [<http://www.clockingit.com/>], e é online.

4.2.3.3 Planningpoker

Trata de uma ferramenta *web* que tem por finalidade simular um *Planning Poker* para equipes que estão fisicamente distribuídas. Na ferramenta são apresentadas as *stories* para serem estimadas pelos participantes. Segue a mesma linha do pensamento do *planning poker* com cartas, porém online. Possui ainda um cronômetro de dois minutos para os debates não se alongarem. Além disso, é possível exportar os resultados das estimativas ao final do jogo. A desvantagem é que é necessário haver uma teleconferência ao mesmo tempo em que os participantes estão utilizando a ferramenta. Este problema poderia ser solucionado se houvesse um chat integrado à ferramenta. A licença de ferramenta é *free* e está disponível em <http://www.planningpoker.com/>.

4.2.3.4 ScrumWorks

ScrumWorks é uma ferramenta instalável e fornece funcionalidades como: geração do product backlog, sprint backlog, priorização de itens do product backlog, gerenciamento de times e usuários, geração de gráficos e relatórios, possui taskboard que permite o usuário monitorar o status das tarefas, permite visualizar a velocidade da equipe e mudanças de escopo, gerenciamento de ROI com valor de negócio, notificação através de email.

Segundo Filho (2009), a principal desvantagem do uso dessa ferramenta é a difícil configuração e instalação, além de instalar uma máquina servidora na máquina do usuário. Outra desvantagem é a demora para acessar os seus recursos.

A ferramenta possui duas versões: *Basic Edition* (gratuita) e *Pro Edition* (paga, \$289 usuário/ano). A ferramenta paga é mais completa [<http://danube.com/scrumworks>].

4.2.3.5 FireScrum

O *FireScrum* é uma ferramenta open source que foi desenvolvida especificamente para apoiar projetos que utilizam *Scrum*. *FireScrum* é uma aplicação *web*, ou seja, pode ser acessada através de um *browser*, podendo ser utilizada em um ambiente de Internet ou Intranet. Tem como funcionalidades principais: cadastro de projetos, criação de itens de *backlog*, priorização dos itens do *product backlog*, criação de *sprints*, associação de itens de *backlog* a *sprints*, criação de tarefas alocação de usuário em determinada tarefa, gráficos de *burndown* do projeto e do *sprint*, possui *taskboard*, cadastro de impedimentos, *planning poker*, compartilhamento de áudio e vídeo para a estimativa do *planning poker*, *chat*, gerenciamento de casos de teste, gerenciamento de *bugs*. A ferramenta também permite ser instalada na máquina do usuário ou invés de ser online.

A ferramenta foi criada por uma equipe de alunos do Mestrado Profissional do CESAR.EDU, no segundo semestre de 2007, voltada especificamente para projetos que utilizam o *Scrum*. E tem como objetivo suprir as necessidades de equipes remotas com foco em simplicidade e facilidade de uso e está disponível em <http://www.firescrum.com>.

4.2.3.6 TargetProcess

TargetProcess é uma ferramenta criada para gerenciar projetos que utilizam processos ágeis. Além de apoiar o *Scrum*, a ferramenta também apóia *Extreme Programming*, *Lean* e outros. A ferramenta possibilita personalização para se adequar à realidade da empresa. Permite ainda a integração com vários sistemas como *Visual Studio 2008*, Integração *Bugzilla*, *TestTrackPro*, *Jira*, *JUnit* e *CSV*. Tem as seguintes características funcionais: gerenciamento de bugs, gerenciamento de caso de testes, gerenciamento de planejado x realizado, integração com subversion, integração com help desk e e-mail, planejamento da iteração, planejamento de releases, visualização do esforço em cada tarefa, *taskboard*, geração de relatórios, burn down chart, gerenciamento de tempo dos usuários, quadro de *Kanban* da ferramenta permite visualizar o fluxo tanto das *stories* como também dos *bugs*.

A ferramenta possui uma versão gratuita *Community Edition*, limitada a 5 usuários, duas versões comerciais a primeira *On-Demand Trial*, que fica hospedada no *TargetProcess* por \$25 usuário/mês limitada e a segunda *On-Site Trail*, um pacote para download, hospedado no servidor do cliente por \$249/ usuário [<http://www.targetprocess.com/>].

4.2.3.7 Rally

Rally une o gerenciamento de projetos ágeis com o gerenciamento de requisitos, teste e falhas. Com esta ferramenta é possível gerenciamento dos requisitos, permite organizar as stories, tarefas, gerenciamento de testes e bugs do projeto, permite fazer o plano de iterações, relatórios, planejamento e acompanhamento das atividades da equipe, integração com outras ferramentas, controle de tempo e custo do projeto e gerenciamento de tempo da equipe. Com essa ferramenta, é possível ter uma imagem em tempo real das características do projeto e de qualidade, prioridades, obstáculos e riscos.

Essa ferramenta possui três versões: *Community edition (free)* disponível para dez (10) usuários e com limitações; *Enterprise edition* (US\$35 usuário/mês) número ilimitado de usuários, mas com limitações e *Unlimited edition* (US\$49 usuário/mês) a versão mais completa e está disponível em: <http://www.rallydev.com/>.

4.2.3.8 VersionOne

Essa ferramenta dá apoio não só ao *Scrum*, mas a diversas metodologias ágeis. Pode ser utilizada tanto em empresas de grande porte quanto de pequeno porte. Disponibiliza suporte a todo o ciclo de vida do projeto e permite que todos os usuários envolvidos no projeto atuem de maneira sincronizada, o que facilita o planejamento, a organização das prioridades e o processo de trabalho.

Tem como principais funcionalidades: planejamento e gerenciamento do backlog, planejamento e monitoramento do sprint, planejamento de releases, gerenciamento de bugs e casos de testes, taskboard, relatórios, burn down chat do projeto e da sprint, gestão de idéias, visualização do planejado x realizado e integração com outras ferramentas.

Possui três licenças: uma versão *Time edition* que é gratuita por um ano de uso com limitação de 10 usuário e funcionalidades, e é própria para uma única equipe ágil; duas versões comerciais, primeira *Enterprise Edition* \$29 usuário/mês para equipes mais experientes e a segunda *Ultimate* \$36 usuário/mês, essa última tem como diferencial apoio ao gerenciamento de testes, gerenciamento de idéias do cliente e relatórios personalizados, disponível em www.versionone.com/.

4.2.4 Comparação entre as ferramentas computacionais

Esta seção apresenta uma análise comparativa de funcionalidades citadas pelos respondentes entre as ferramentas analisadas, a fim de verificar se as funcionalidades desejadas pelos respondentes para facilitar suas atividades de *Scrum Master* estão presentes em uma única ferramenta.

Porém, nas ferramentas *Clocking it*, *Planningpoker* e *Scrumability* não será feita a análise com suas funcionalidades, pois: *Clocking it* é uma ferramenta que não é voltada para metodologias ágeis; *PlanningPoker* é uma ferramenta que tem somente a funcionalidade do jogo do *planning poker*; e a *Scrumability* não será mencionada no Quadro 5 abaixo, pois trata-se de uma ferramenta própria e não disponível no mercado, como comentado anteriormente.

Logo, no Quadro 5, apresenta a análise comparativa entre as funcionalidades citadas pelos respondentes da pesquisa e as funcionalidades das ferramentas analisadas.

Quadro 5: Análise comparativa

<i>Funcionalidades</i>	<i>Scrumy</i>	<i>ScrumWorks</i>	<i>FireScrum</i>	<i>TargetProcess</i>	<i>Rally</i>	<i>VersionOne</i>
Planning poker	-	-	X	-	-	-
Cálculo do ritmo sugerido (em pontos) para o time fechar o projeto com sucesso	-	X	-	-	X	X
Cálculo do tempo total do projeto	X	-	X	-	-	-
Cálculo e Gráfico do desvio do projeto	X	-	-	X	-	-
Controle do status dos requisitos e das user stories (não iniciado, em desenvolvimento, desenvolvido)	X	-	X	X	X	-
Cronograma do sprint (com as tarefas listadas, estimativa para cada tarefa, distribuição das tarefas por data, status de cada tarefa e possibilidade de identificação das tarefas)	X	X	X	-	-	X
Distribuição de atividades para os componentes da equipe	X	-	X	X	X	X
Geração do burn down chart	X	X	X	X	-	X
Gerenciamento de risco (do projeto e do sprint)	-	-	-	-	-	X
Impediment Backlog	-	X	X	X	-	X
Integração com ferramenta de registro de bugs/problemas	-	X	-	X	X	-
Kanban/taskboard	X	X	X	X	X	X
Matriz de relacionamento Requisito x User story	-	-	-	-	X	-
Planejamento e controle dos Release (Roadmap dos sprints)	-	X	X	X	-	-
Product Backlog	X	X	X	X	X	X
Registro das lições aprendidas (acompanhamento)	-	-	-	X	-	-
Registro do tempo gasto e restante para cada tarefa	-	-	-	-	-	-
Registros de eventos do sprint (cancelamento do sprint, mudança de membros)	-	X	-	-	-	X
Relatórios de desempenho do time (por membro, por sprint,..)	-	-	-	-	-	X
Relatórios do projeto e/ou atas	-	X	X	X	X	X
Sprint Backlog	X	X	X	X	X	X
Troca instantânea de mensagens (ferramenta de comunicação)	-	-	X	-	X	-
Velocidade de equipe (Estimado x Trabalhado)	-	-	-	X	X	X

Fonte: Elaboração própria, com dados da pesquisa de campo.

Conforme o Quadro 5 é possível verificar que apenas o *Kanban* ou *taskboard*, *product backlog* e *sprint backlog* são as funcionalidades que aparecem em todas as ferramentas analisadas. E dentre estas três (03) funcionalidades, o *product backlog* e o *sprint backlog* são consideradas essenciais pelos respondentes. Logo, as ferramentas satisfazem os respondentes nesses quesitos.

Porém, é notório que nenhuma ferramenta analisada possui todas as necessidades dos respondentes. Por este motivo, a ferramenta proposta neste trabalho visa atender estas necessidades diárias dos *Scrum Masters* com o diferencial de que foi feita uma pesquisa com os usuários dessa ferramenta, para assim conter somente o necessário e essencial para seu trabalho diário.

5 A FERRAMENTA JUSTSCRUM

Neste capítulo será apresentada a ferramenta JustScrum, cujas funcionalidades foram identificadas a partir dos questionários respondidos pelos Scrum Masters. Também serão discutidas suas principais características, arquitetura e as tecnologias utilizadas para seu desenvolvimento.

O JustScrum corresponde a um dos resultados desta pesquisa, sendo que seu principal objetivo é unificar em uma ferramenta as funcionalidades mencionadas pelos entrevistados, uma vez que durante a análise das ferramentas já disponíveis no mercado, não foi identificada nenhuma com todos os recursos elencados na Questão 14 do questionário (Apêndice A).

A criação da ferramenta JustScrum justifica-se pelo fato de que houve a preocupação prévia da realização de uma pesquisa com alguns profissionais Scrum Masters, a fim de identificar as funcionalidades ditas essenciais pelos mesmos para o desenvolvimento de uma ferramenta computacional capaz de auxiliá-los em suas atividades diárias.

O JustScrum foi desenvolvido em conjunto com um aluno de graduação da Faculdade Centro Universitário do Norte – Uninorte localizada em Manaus, o qual está utilizando os resultados obtidos neste trabalho para desenvolver seu trabalho de conclusão de curso.

Características

O JustScrum trata-se de uma ferramenta *free*, desenvolvida com interface Web, portanto pode ser utilizada por meio de um *browser*, como Firefox [www.mozilla.com/firefox].

Para o desenvolvimento do JustScrum foi escolhido o padrão arquitetural MVC (*Model View Controller*), o qual corresponde a um modelo com o objetivo de separar a lógica da aplicação (implementação das regras de negócio) da interface com o usuário. Para tanto, possui três componentes distintos que são: o modelo (*model*), a visão (*view*) e o controlador (*controller*), conforme ilustrado na Figura 7.

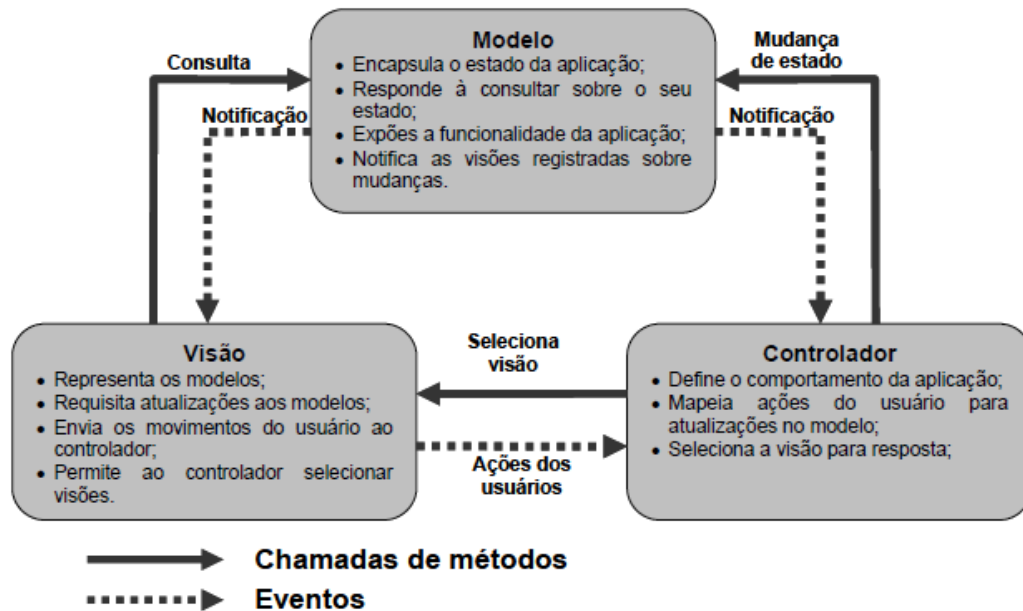


Figura 7: Arquitetura MVC

Fonte: Filho (2009).

Onde o componente *modelo* contém a lógica da programação; o componente *visão* é a camada de apresentação com usuário, exibe ao usuário os estados internos do modelo, na forma de interface, gráfica ou não; e o componente *controlador* atende às entradas do usuário na interface e as traduz para atualizações do modelo, funciona como intermediário entre a camada de apresentação e a camada de modelo.

Para o desenvolvimento, foram utilizadas tecnologias free, as quais são listadas a seguir:

- Java EE 5.0
[<http://www.oracle.com/technetwork/java/javaee/overview/index.html>] - usado para prover bibliotecas que forneçam funcionalidade para desenvolver, empacotar e implantar aplicações empresariais, tolerante a falhas e multicamada, baseado amplamente em componentes modulares executando em um servidor de aplicações;
- JBoss Seam [<http://seamframework.org/>] - é utilizado para eliminar a complexidade em níveis de arquitetura e API. Oferecendo um total controle sobre a implementação da lógica de negócio sem se preocupar com a exposição de informações ou configuração excessiva em arquivos XML, dispondo de anotações para classes Java e componentes bem definidos para a camada de apresentação;

- JBoss Server [<http://jboss.org/>] - usado como servidor onde a aplicação vai rodar;
- Java-Script [<http://www.javascript.com/>], Rich-faces [<http://jboss.org/richfaces>] e PrimeFaces [www.primefaces.org] - bibliotecas empregadas na implementação da camada de apresentação baseada em componentes, facilitando a integração com a camada de negócio;
- PostgreSQL 8.2 [www.postgresql.org] - usado como SGBD para gerenciamento do banco de dados da ferramenta; e
- Eclipse 3.4 [www.eclipse.org] - usado como ambiente de desenvolvimento do sistema.

Desenvolvimento

Para o desenvolvimento da ferramenta foi seguido um ciclo iterativo e incremental. Para melhor visualização de tal fluxo, a Figura 8 apresenta todos os passos executados:

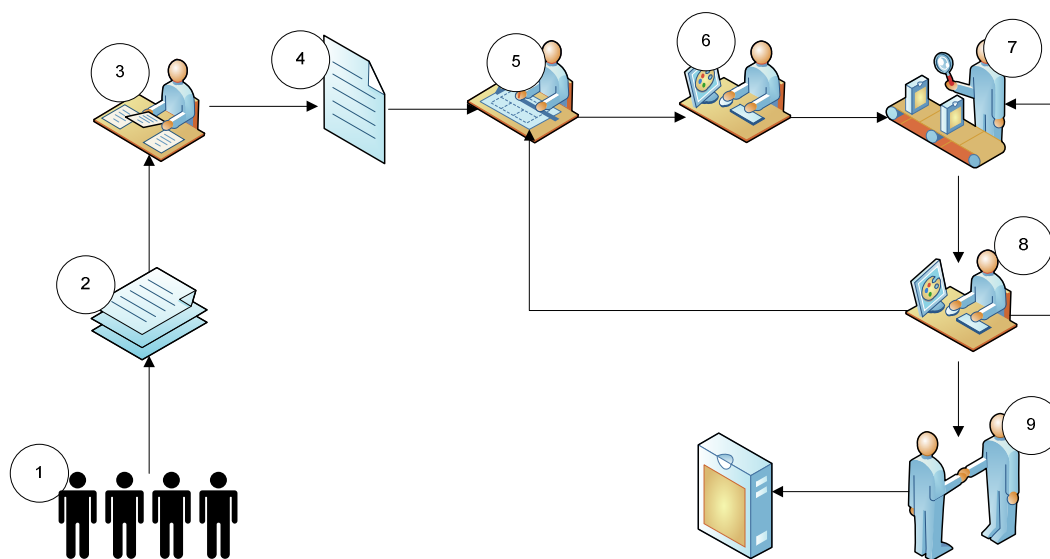


Figura 8: Ciclo de desenvolvimento

Fonte: elaboração própria.

Conforme a Figura 8, para a concepção da ferramenta JustScrum foram realizados nove (09) passos, que formaram o ciclo de desenvolvimento adotado pela pesquisadora deste trabalho e pelo aluno de graduação. Os passos são detalhados a seguir:

Passo 01 - Os *Scrum Masters* deram suas opiniões a respeito do que gostariam de ter em uma ferramenta para ajudar nas suas atividades diárias no questionário aplicado, Questão 14;

Passo 02 - As funcionalidades solicitadas pelos entrevistados foram agrupadas na forma de *user stories*, as quais são descrições simples de algo que o usuário possa efetivamente utilizar (entregável). As *user stories* são escritas sob o ponto de vista das necessidades dos usuários e são compostas por quatro tópicos:

- Solicitante – representa o papel (não uma pessoa) para o qual a user story será útil;
- Necessidade – descrição sucinta do que deve ser feito;
- Benefício – representa o que a user story trará de ganho para o usuário. Ressalta-se que o próprio solicitante deve responder qual o benefício;
- Critérios de aceitação – representam as regras que devem ser satisfeitas para que a user story seja aceita pelo usuário.

Um exemplo de *user story* pode ser visualizado na Figura 9.

US_1	O Product Owner poder identificar os itens do backlog de um projeto
Descrição da Entrega: • Como um Product Owner, ○ Gostaria de poder registrar minhas idéias para o projeto como itens de um <i>product backlog</i>. ○ Então, eu poderei manter o registro de qualquer idéia que eu tiver sobre o produto para posterior priorização. Critérios de aceitação: • Deve haver um cadastro de projetos com situação, data de inicio e fim do projeto; • Deve ser possível registrar um ou mais módulos de um projeto; • Deve haver o cadastro de item do <i>product backlog</i>; • Não deve ser permitido cadastrar itens no product backlog em um projeto concluído ou cancelado; • O item inserido deve ser posto em última prioridade para posterior priorização. • Projeto de tela: US_1_IdentificacaoItensProductBacklog.pdf	

Figura 9: Exemplo de user story

Fonte: elaboração própria.

Passo 03 - A pesquisadora priorizou as user stories, de acordo com a classificação estabelecida pelos Scrum Masters em: Essencial, Importante e Desejável. Nessa priorização também considerou a quantidade de solicitação das mesmas identificadas nos questionários, conforme Quadro 4 da seção 4.1.4.

Passo 04 - Foi criada a lista de entregas priorizadas das *user stories*, conforme o Quadro 6.

Quadro 6: Priorização das *user stories*

Prioridade	Nome da <i>user stories</i>
1	US_1 – O Product Owner poder identificar os itens do backlog de um projeto
2	US_2 – O Product Owner poder priorizar os itens do product backlog
3	US_5 – O time poder montar o sprint backlog
4	US_3 – O time poder fazer o planejamento do sprint
5	US_4 – O time pode fazer a execução do sprint
6	US_10 – O Scrum Master poder planejar e controlar os releases
7	US_6 – O time poder visualizar o burn down
8	US_7 – O time poder montar impediment backlog
9	US_8 – O time poder descrever as lições aprendidas do projeto
10	US_9 – O time poder cadastrar os riscos
11	US_11 – O time poder configurar os feriados para planejar o sprint

Fonte: Elaboração própria.

Passo 05 - Neste ponto foram elaborados projetos de tela para a entrega priorizada para facilitar a validação das funcionalidades antes da implementação das mesmas. Ressalta-se que dois dos *Scrum Masters* entrevistados participaram deste processo de validação.

Além dos projetos de tela, também foi elaborado o projeto de banco de dados para a entrega do ciclo. A Figura 10 e a Figura 11 ilustram um exemplo de projeto de tela e de banco de dados respectivamente, para a entrega US_2 – O *Product Owner* poder priorizar os itens do *product backlog*.

Para o desenvolvimento dos projetos de tela foi utilizado o programa Microsoft Office Visio 2007 [<http://office.microsoft.com/en-us/visio/>] e para o desenvolvimento do projeto de banco de dados foi utilizado o programa Erwin [www.erwin.com].

Página Inicial » Gestão de Projeto » Priorização do Product Backlog » Novo

Nova Priorização do Product Backlog

• Não é possível priorizar o backlog desse projeto, pois o mesmo está cancelado ou concluído.

Dados do projeto:

Nome Ver

Itens priorizados:

Prioridade	Item do Product Backlog	Módulo	Tamanho
1	PBI_1.1 - Entrega 1	Módulo 1	5
2	PBI_3.2 - Entrega 2	Módulo 3	2
3	PBI_2.2 - Entrega 3	Módulo 2	3

Voltar

Regra: Não deve permitir priorizar o backlog dos projetos cancelados ou concluídos

Regra: As prioridades devem ser maior ou igual a 1 e não pode ser superior a quantidade de entregas do product backlog. Ex.: Se a quantidade de entregas do Projeto 1 for 45, então a maior prioridade deverá ser 45.

Regra: Ao mudar a prioridade, deve reordenar as entregas. Ex.: Se mudar a entrega 3 para 1, então a entrega 1 deve ser atualizada para 2 e a entrega 2 para 3.

Regra: A medida que altera a prioridade e a estimativa do PBI, a base de dados já deve ser alterada

30 primeiros registros
20 primeiros registros
10 primeiros registros
Todos os registros

Deve ser um inplace input

Deve ser um inplace input

Navegação: Retorna para tela de pesquisa

Figura 10: Projeto de tela US_2 - O Product Owner poder priorizar os itens do product backlog

Fonte: elaboração própria.

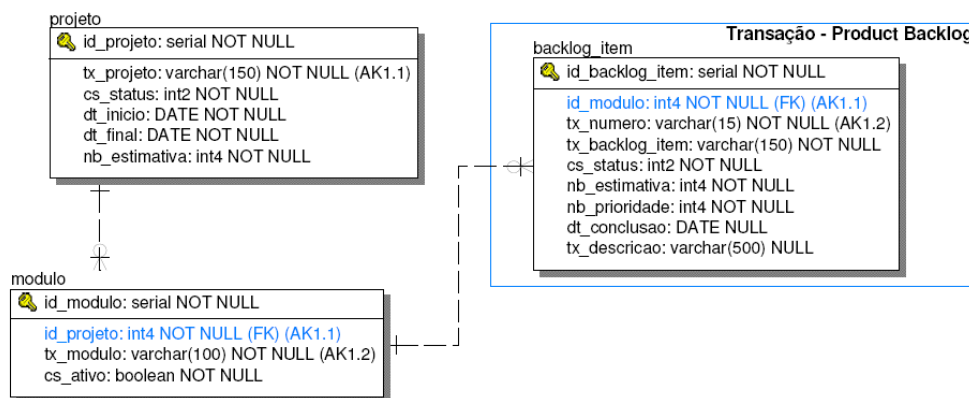


Figura 11: Banco de dados US_2 - O Product Owner poder priorizar os itens do product backlog

Fonte: elaboração própria.

Passo 06 - Após, elaborados os projetos de tela e banco de dados, foi feita a implementação da entrega.

Passo 07 - Neste passo, a entrega era submetida a testes por dois *Scrum Masters* que participaram da pesquisa e documentavam os problemas detectados conforme o modelo abaixo:

Problema: <descrição do problema detectado>

Pré-condições: <pré-condições para reproduzir o problema>

Passos: <passos para reproduzir o problema>

Resultado esperado: <descrição do resultado esperado pelo sistema>

Resultado atual: <descrição do resultado atual do sistema>

Passo 8 - Os problemas detectados eram corrigidos pelos desenvolvedores e retornavam para os testadores a fim de verificá-los. Caso novos bugs fossem encontrados, os desenvolvedores corrigiam novamente. Esse ciclo se repetia até não haver mais erros naquela entrega.

Passo 9 - Após não haver mais erros na entrega era dado o aceite dos testadores e o produto era incrementado com a próxima entrega priorizada, até terminar toda a lista de user stories. As figuras 12 e 13 ilustram duas telas implementadas.

Manaus, 10 de Agosto de 2010 Logout

Home » Pesquisa Sprint Backlog » Sprint Backlog

Sprint Backlog

Dados gerais:

Projeto * Sistema de gerenciamento industrial

Time * Time de desenvolvimento

Sprint * 1

☒ Incluir sábado no sprint
☐ Incluir domingo no sprint

Início * 16/08/2010

Final * 31/08/2010

Cancelado * Não

Itens do Sprint:

Estimativa total 360

[Listar Product Backlog](#)

Item do Product Backlog	Estimativa	Situação	Conclusão	Ação
PBI_1.2 - O almoxarife poder registrar saída de materias	90	Concluído	10/08/2010	
PBI_3.1 - Usuario poder realizar checklist de melhorias	20	Em desenvolvimento		
PBI_1.1 - O almoxarife poder realizar entrada de produto	30	Não iniciado		✖
PBI_2.1 - Usuario poder realizar controle de fluxo	120	Não iniciado		✖
PBI_4.1 - Usuario poder realizar faturamento mensal	100	Não iniciado		✖

5 Registros Encontrados

Salvar Cancelar

Figura 12:Tela implementada de Sprint backlog

Fonte: elaboração própria.

Manaus, 10 de Agosto de 2010 Logout

Geral ▾
Gerenciamento ▾

Home » Product Backlog » Novo Product Backlog

Novo Product Backlog

Dados do projeto:

Nome: Sistema de gerenciamento industrial
Situação: Em andamento

Item do Product Backlog:

Módulo*: almoxarifado ▾
Título*: O almoxarifado poder realizar entrada de produto
Código*: PBI_1.1
Estimativa*: 30 Pontos
Situação*: Não iniciado ▾
Descrição: O almoxarifado poder realizar entrada de produto.

Salvar Cancelar

Fechar

Figura 13: Tela implementada de Product backlog

Fonte: elaboração própria.

Ao final do desenvolvimento da ferramenta, a mesma foi submetida a uma avaliação de satisfação pelos *Scrum Masters*, os quais responderam um questionário, propondo melhorias ao software como, por exemplo, a implementação do módulo de cadastro de feriados. Esse questionário de avaliação pode ser visualizado no Apêndice B.

As instruções de como baixar e configurar a ferramenta JustScrum encontram-se no Apêndice C.

Funcionalidades

Por ser uma ferramenta voltada para atender às principais funcionalidades solicitadas pelos *Scrum Masters* entrevistados, deu-se prioridade àquelas classificadas como Essenciais: *product backlog*; geração do *burn down chart*; *sprint backlog*; distribuição de atividades para os componentes da equipe; *Kanban/ taskboard*; cálculo do tempo total do projeto; *impediment Backlog*; gerenciamento de risco (do projeto e do *sprint*); cálculo e gráfico do desvio do projeto; planejamento e controle dos *releases* (*Roadmap* dos *sprints*); cronograma do *sprint* (com as tarefas listadas, estimativa para cada tarefa, distribuição das tarefas por data, status de cada tarefa e possibilidade de identificação das tarefas) e cálculo do ritmo sugerido (em pontos) para o time fechar o projeto com sucesso.

O Quadro 7 apresenta como funcionalidades propostas pelos *Scrum Masters* foram agrupadas em *user stories*.

Quadro 7: Associação entre funcionalidades e user stores

Funcionalidade	User Story
Ambiente de simulação do planning poker	Não foi implementada
Cálculo do ritmo sugerido (em pontos) para o time fechar o projeto com sucesso	US_6 – O time poder visualizar o burn down
Calculo do tempo total do projeto	US_6 – O time poder visualizar o burn down
Cálculo e Gráfico do desvio do projeto	US_6 – O time poder visualizar o burn down
Controle do status dos requisitos e das user stories (não iniciado, em desenvolvimento, desenvolvido)	US_5 – O time poder montar o sprint backlog
Cronograma do sprint (com as tarefas listadas, estimativa para cada tarefa, distribuição das tarefas por data, status de cada tarefa e possibilidade de identificação das tarefas)	US_3 – O time poder fazer o planejamento do sprint
Distribuição de atividades para os componentes da equipe	US_4 – O time pode fazer a execução do sprint
Geração do burn down chart	US_6 – O time poder visualizar o burn down
Gerenciamento de risco (do projeto e do sprint)	US_9 – O time poder cadastrar os riscos
Impediment Backlog	US_7 – O time poder montar impediment backlog
Integração com ferramenta de registro de bugs/problemas	Não foi implementada
Kanban/ taskboard	US_4 – O time pode fazer a execução do sprint
Matriz de relacionamento Requisito x User story	Não foi implementada
Planejamento e controle dos Release (Roadmap dos sprints)	US_10 – O Scrum Master poder planejar e controlar os releases
Product Backlog	US_1 – O Product Owner poder identificar os itens do backlog de um projeto e US_2 – O Product Owner poder priorizar os itens do product backlog
Registro das lições aprendidas (acompanhamento)	US_8 – O time poder descrever as lições aprendidas do projeto
Registro do tempo gasto e restante para cada tarefa	US_4 – O time pode fazer a execução do sprint
Registros de eventos do sprint (cancelamento do sprint, mudança de membros, ...)	US_3 – O time poder fazer o planejamento do sprint
Relatórios de desempenho do time (por membro, por sprint,...)	Não foi implementado
Relatórios do projeto e atas	Não foi implementado
Sprint Backlog	US_5 – O time poder montar o sprint backlog
Troca instantânea de mensagens (ferramenta de comunicação)	Não foi implementado
Velocidade de equipe (Estimado x Trabalhado)	US_6 – O time poder visualizar o burn down

Fonte: Elaboração própria.

É possível visualizar no Quadro 7 que das vinte e três (23) funcionalidades solicitadas pelos Scrum Masters entrevistados, somente cinco (05) não foram implementadas, que são: ambiente de simulação do planning poker, integração com ferramenta de registro de bus/problemas, matriz de relacionamento requisitos x user story, relatórios de desempenho do time e relatórios do projeto e atas de reunião. Essas cinco funcionalidades não foram consideradas essenciais em uma ferramenta pelos Scrum Masters entrevistados e algumas delas foram ferramentas descartadas por esses profissionais nas questões 8 e 11 do questionário (Apêndice A).

6 CONSIDERAÇÕES FINAIS

Esta pesquisa procurou compreender as principais dificuldades dos *Scrum Masters* em relação ao uso de ferramentas para auxiliar suas atividades. Para isso, primeiramente foi realizado um levantamento teórico sobre a metodologia Scrum, visando principalmente as responsabilidades e atividades dos Scrum Masters. Após este levantamento foi aplicado um questionário com o intuito de coletar dados para identificar as atividades desempenhadas por estes profissionais, bem como as ferramentas de apoio utilizadas pelos mesmos. Com isso, pôde-se traçar um comparativo entre as funcionalidades existentes em tais ferramentas e as necessidades elencadas pelos Scrum Masters entrevistados.

Os resultados obtidos nessa pesquisa podem ser resumidos nos seguintes tópicos:

1. Com a coleta e análise dos dados das entrevistas, o estudo confirmou que as atividades e responsabilidades de um Scrum Master existentes na literatura são extrapoladas na prática pelos Scrum Masters da pesquisa realizada e que algumas atividades, até mesmo abolidas na literatura, como coletar e documentar requisitos do projeto, são exercidas por estes profissionais em suas funções diárias em um projeto;
2. Por meio desta pesquisa foi possível ainda verificar que os Scrum Masters entrevistados acabam tendo que exercer outras atividades que deveriam ser feitas por outros envolvidos no projeto, como manter a base de dados do projeto, para que este atinja o seu sucesso, mas nunca se esquecendo de suas atividades ditas obrigatórias pela literatura;
3. Com este estudo foi possível realizar uma compilação das principais ferramentas utilizadas pelos Scrum Masters entrevistados nas suas atividades diárias em um projeto e avaliar as funcionalidades existentes nas mesmas;
4. Através dos dados obtidos e da revisão bibliográfica foi possível realizar uma análise comparativa entre as funcionalidades das ferramentas pesquisadas e as funcionalidades listadas pelos Scrum Masters entrevistados para auxiliar em suas atividades profissionais;
5. Ao final da análise foi possível verificar que as ferramentas existentes não atendem completamente e sim parcialmente às necessidades e às expectativas dos Scrum Masters entrevistados uma vez que não integram em um único pacote todos os recursos necessários ao desempenho de suas atividades; e
6. Para resolver as limitações encontradas, foi desenvolvida uma ferramenta,

chamada JustScrum, que levou em consideração as atividades desempenhadas por estes profissionais, assim como as funcionalidades sugeridas pelos mesmos na entrevista.

O maior diferencial desta ferramenta encontra-se no fato de ter sido realizada uma pesquisa prévia com alguns profissionais, cujo objetivo foi identificar as atividades realizadas pelos Scrum Masters, compará-las com aquelas descritas na literatura e, assim, fundamentar o desenvolvimento da ferramenta nas reais necessidades desses profissionais.

Somando-se a isso, destaca-se que o JustScrum, diferentemente das ferramentas pesquisadas, integra em um único software as funcionalidades propostas pelos Scrum Masters entrevistados nesta pesquisa. Esse fato revelou-se fundamental ao longo da pesquisa para atender às expectativas dos profissionais, uma vez que de acordo com os dados coletados, os entrevistados relataram que usam diversas ferramentas para desempenhar suas atividades.

Complementado a descrição das principais inovações do JustScrum, o Quadro 8 apresenta um comparativo entre as ferramentas pesquisadas e aquela desenvolvida neste trabalho.

Quadro 8: Análise comparativa das ferramentas

<i>Funcionalidades</i>	<i>Scrumy</i>	<i>ScrumWorks</i>	<i>FireScrum</i>	<i>TargetProcess</i>	<i>Rally</i>	<i>VersionOne</i>	<i>JustScrum</i>
Planning poker	-	-	X	-	-	-	-
Cálculo do ritmo sugerido (em pontos) para o time fechar o projeto com sucesso	-	X	-	-	X	X	X
Cálculo do tempo total do projeto	X	-	X	-	-	-	X
Cálculo e Gráfico do desvio do projeto	X	-	-	X	-	-	X
Controle do status dos requisitos e das user stories (não iniciado, em desenvolvimento, desenvolvido)	X	-	X	X	X	-	X
Cronograma do sprint (com as tarefas listadas, estimativa para cada tarefa, distribuição das tarefas por data, status de cada tarefa e possibilidade de identificação das tarefas)	X	X	X	-	-	X	X
Distribuição de atividades para os componentes da equipe	X	-	X	X	X	X	X
Geração do burn down chart	X	X	X	X	-	X	X
Gerenciamento de risco (do projeto e do sprint)	-	-	-	-	-	X	X
Impediment Backlog	-	X	X	X	-	X	X
Integração com ferramenta de registro de bugs/problemas	-	X	-	X	X	-	-
Kanban/taskboard	X	X	X	X	X	X	X
Matriz de relacionamento Requisito x User story	-	-	-	-	X	-	-
Planejamento e controle dos Release (Roadmap dos sprints)	-	X	X	X	-	-	X
Product Backlog	X	X	X	X	X	X	X
Registro das lições aprendidas (acompanhamento)	-	-	-	X	-	-	X
Registro do tempo gasto e restante para cada tarefa	-	-	-	-	-	-	X
Registros de eventos do sprint (cancelamento do sprint, mudança de membros, ...)	-	X	-	-	-	X	X
Relatórios de desempenho do time (por membro, por sprint,...)	-	-	-	-	-	X	-
Relatórios do projeto e/ou atas	-	X	X	X	X	X	-
Sprint Backlog	X	X	X	X	X	X	X
Troca instantânea de mensagens (ferramenta de comunicação)	-	-	X	-	X	-	-
Velocidade de equipe (Estimado x Trabalhado)	-	-	-	X	X	X	X

Fonte: Elaboração da pesquisadora

Analisando os dados apresentados no Quadro 8, pode-se observar que dentre os recursos elencados pelos Scrum Masters entrevistados, o JustScrum atende 74% dos mesmos enquanto que a segunda ferramenta mais aderente às necessidades desses profissionais são as ferramentas FireScrum, TargetProcess e Rally atendendo 56% das funcionalidades.

Vale ressaltar que as funcionalidades que não foram implementadas nesta versão da JustScrum não eram consideradas essenciais em uma ferramentas pelos Scrum Masters entrevistados.

Diante do exposto, pode-se considerar que as principais contribuições dessa dissertação são:

1. Identificação das reais atividades dos Scrum Masters entrevistados em um projeto Scrum;
2. Identificação das principais ferramentas não-computacionais e computacionais utilizadas por estes profissionais em suas atividades diárias;
3. Apresentação dos recursos mais importantes para apoiar o trabalho desses Scrum Masters entrevistados;
4. Análise das ferramentas existentes no mercado para apoiar estes profissionais;
5. Análise comparativa entre as funcionalidades necessárias em uma ferramenta computacional e as ferramentas disponíveis no mercado; e
6. Disponibilização de uma ferramenta gratuita e integrada para apoiar as atividades dos Scrum Masters entrevistados. Destaca-se que o JustScrum já está sendo utilizado por cinco (05) dos treze (13) participantes desta pesquisa.

Alguns trabalhos futuros podem ser explorados a partir deste estudo:

1. Ampliar o número de Scrum Masters entrevistados, não apenas com profissionais do Brasil, mas também de outros países, com o intuito de identificar melhorias e novas funcionalidades para o JustScrum;
2. Expandir o quadro de ferramentas existentes no mercado para a comparação das funcionalidades;
3. Desenvolvimento das funcionalidades importantes e desejáveis propostas pelos Scrum Masters entrevistados; e
4. Ampliar a pesquisa para os demais membros do Team e para o Product Owner para o desenvolvimento de funcionalidades essenciais para o trabalho destes.

Finalmente, ressalta-se que a pesquisa mostrou que os Scrum Masters entrevistados atuam em atividades que ultrapassam aquelas listadas na literatura, e que as ferramentas estudadas neste trabalho não atendiam a todas as expectativas destes profissionais.

REFERÊNCIAS

AGILE MANIFESTO, 2004. Disponível em <<http://agilemanifesto.org/>>. Acesso em: 12 de maio 2009.

ALMEIDA, Carla Andréia de. **Implantação da manufatura enxuta em uma célula de produção**. 2007. Trabalho de Conclusão de Curso (Graduação) - Curso de Engenharia de Produção e Sistemas da Universidade do Estado de Santa Catarina, Joinville, 2007.

ARAUJO, Allan R.S.; SILVA, Juliana M.; MITTELBAACH, Artur F. **Scrum: Novas Regras do Jogo**. 2006. Disponível em:<<http://www.sbgames.org/sbgames06/proceedings/posters.htm>> Acesso em: 06 jun. 2009.

ASFORA, Diego Maciel. **Uma abordagem para a priorização de requisitos ágeis**. 2009. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Pernambuco, Recife, 2009.

BECK, K. et al. **Manifesto for Agile Software Development**. Dezembro 2001. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em: outubro, 2009.

BOEHM, B. **A View of 20th and 21st Century Software Engineering**. In: Proc. of the 28th ICSE, Shanghai, China. 2006. Disponível em: <http://portal.acm.org/citation.cfm?id=1134288>. Acessado em 23 de maio de 2009.

BONO, Edward de. **Os seis chapéus do pensamento: o método mundialmente consagrado de tornar o debate de ideias mais organizado, rápido e produtivo**. 2008. Ed. Sextante

BRANAS, Rodrigo. **Planning Poker Online!**. 2009. Acessado em 30 de Janeiro de 2010. Disponível em: <http://www.oncast.com.br/blog/?tag=planning-poker>

BROD, Cesar. **Scrum, perguntas e respostas**. 2007. Acessado em 30 de Janeiro de 2010. Disponível em: <http://brodtec.com/?q=node/223>

CAMARA, Fabio. **Uma metodologia ágil**. 2008. Disponível em:<<http://www.linhadecodigo.com.br/Artigo.aspx?id=2084>>. Acesso em: 23 maio 2009.

CASTRO, Flávio Steffens de. **Gráfico Burndown – Sugestão de uso**. 2009. Disponível em: <http://www.agileway.com.br/2009/08/18/grafico-burndown-sugestao-de-uso/>. Acesso em: 30 jan. 2010.

CAVALCANTI, Eric de Oliveira. **Firescrum: ferramenta de apoio à gestão ágil de projetos utilizando Scrum**, 2009. Dissertação (Mestrado em Engenharia de Software) - Centro de Estudos e Sistemas Avançados do Recife, Recife, 2009.

DANTAS, Edmundo Brandão. **Marketing descomplicado**. Brasília: Senac, 2005.

DUARTE, Ricardo. **Ferramentas para trabalhar com Scrum**. Disponível em:<<http://materialivre.wordpress.com/2009/09/18/ferramentas-para-trabalhar-com-scrum/>> Acesso em: 30 set. 2009.

FACHIN, Odília. **Fundamentos de Metodologia**. 4.ed. São Paulo: Saraiva, 2003.

FACHIN, Odília. **Fundamentos de metodologia**. 5 ed. São Paulo: Saraiva, 2006.

FARIAS JR., Luiz. **Como montamos o quadro do Scrum**. 2008. Disponível em:< <http://bluesoft.wordpress.com/2008/09/17/como-montamos-o-quadro-do-scrum/>.> Acesso em: 06 ago. 2009.

FERREIRA, Henrique Santos. **Um estudo da adequação de personalidades e papéis na metodologia Scrum de desenvolvimento de software**. 2007. Trabalho apresentado como requisito parcial para aprovação na disciplina Empreendedorismo, Centro de Informática da Universidade Federal de Pernambuco, Recife, 2007.

FERREIRA, Rodrigo Lopes. **Introdução Ao Desenvolvimento Ágil Com Scrum**. 2009. Disponível em:< <Http://Www.Scribd.Com/Doc/15692121/Introducao-Ao-Desenvolvimento-Agil-Com-Scrum>> Acesso em: 23 maio 2009

GADELHA FILHO, Adalberto de Castro. **Uma ferramenta web para gerenciamento de projetos SCRUM**. 2009. Trabalho de Conclusão de Curso (Graduação)-Curso de Sistemas de Informação do Centro Universitário Luterano de Manaus, Manaus, 2009.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2006.

GIL, A. C. **Métodos e Técnicas de pesquisa social**. 5. ed. São Paulo: Atlas, 1999.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GOMES, Franciene, D. **Processo Ágil de Desenvolvimento Baseado em Framework**. 2007. 163 f. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília da Fundação de Ensino Eurípides Soares da Rocha, Marília, 2007.

GONÇALVES. Eduardo dos Santos; REIS FILHO, Heitor Boeira dos. **Ferramenta para gerenciamento de requisitos em metodologias ágeis**. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Luterana do Brasil. 2008. Disponível em:< <http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/view/4593/3480>.> Acesso em: 25 maio 2009.

GRENNING, James. **Planning Poker or How to avoid analysis paralysis while release planning**. 2002. Acessado em 30 de Janeiro de 2010. Disponível em: www.renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf

GRESSLER, Lori Alice. **Introdução à pesquisa: projetos e relatórios**. 2.ed. São Paulo: Edições Loyola, 2004.

HIGHSMITH, J. Agile Project Management - **Creating Innovative Products**. Boston: Addison- Wesley, 2004.

MAGNO, Alexandre. **ScrumMaster por ele mesmo**. Revista Visão Ágil. 2008. Ano II, Edição 04. Disponível em: <http://www.visaoagil.com/>. Acessado em 23 de janeiro de 2009.

MAGNO, Alexandre. **The Zen of Scrum**. Certified Scrum Master Training. Curso ministrado em agosto de 2009 em Manaus. Manaus, 2009.

MANIFESTO ÁGIL. 2004. Disponível em:< <http://agilemanifesto.org/>> Acesso em: 24 jul. 2009.

MARCONI, Marina de Andrade & LAKATOS, Eva Maria. **Metodologia do trabalho científico**. 7 ed. São Paulo: Atlas, 2007.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Técnicas de pesquisa**. 4.ed. São Paulo: Atlas, 1999.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Técnicas de pesquisa**. 5.ed. São Paulo: Atlas, 2002.

MARIZ, Leila Maria Rodrigues de Souza. **Uma análise comparativa entre o discurso e a prática na gestão ágil de projetos de software com Scrum**, 2009. Dissertação (Mestrado em Ciência da Computação)- Centro de Informática da Universidade Federal de Pernambuco, Recife, 2009.

MUELLER, Rafael. **Planning Poker**. 2007. Disponível em:< <http://queroseragil.wordpress.com/2007/06/05/planning-poker/>> Acesso em: 30 jan. 2010.

NORTON, D. **Test Driven Development, Agile Scrum**. Disponível em:< <http://codebetter.com/blogs/darrell.norton/pages/50339.aspx>> Acesso em: 01 out. 2009.

OLIVEIRA, Ebenezer Silva de et al. **Uso de metodologias ágeis no desenvolvimento do sistema de embarque da CST - Companhia Siderúrgica de Tubarão**. 2003. Trabalho apresentado ao 7º Seminário de Automação de Processos na Associação Brasileira de Metalurgia e Materiais em Santos, SP, 2003. Disponível em:< <http://www.atan.com.br/dados/artigos/0682-274-00%20-%20Uso%20de%20Metodologias%20Ageis%20Proc.%20Desenv.pdf>> Acesso em: 15 jul. 2009.

ORTIZ, Rodrigo Lariante. **Metodologias ágeis com XP e Scrum**. 2009. Disponível em: <http://www.ortiz.blog.br/tag/metodologias-ageis/> Acesso em: 30 jan.2010.

PADUA, Elisabete Matallo Marchesini. **Metodologia da pesquisa: abordagem teórico-prática**. São Paulo: Editora Papirus, 2004.

PEREIRA, P. ; TORREAO, P. C. ; MARÇAL, A. S. C. Entendendo SCRUM para Gerenciar Projetos de Forma Ágil. **MUNDO PM**, Brasil, p. 64 - 71, abr. 2007. Disponível em: <http://www.cesar.org.br/files/file/SCRUM_MundoPM-Abril-Maio-2007.pdf> Acesso em: 15 maio 2009.

PEREIRA, Paulo. **Metodologias ágeis**. Recife, 2008. Disponível em: <<http://www.cesar.edu.br/docs/paulocaju.pdf>> Acesso em: 25 nov. 2009.

PERES, Eduardo Meira Peres. **Introdução ao Gerenciamento de Projetos utilizando Métodos Ágeis: SCRUM**. 2008. Acessado em 30 de janeiro de 2010. Disponível em: www.dbserver.com.br/2009/Junho/scrum_introducao.ppt

PRESSMAN, Roger S. **Engenharia de software**. 6.ed. São Paulo: McGraw-Hill, 2006.

RAHAL JR, Nelson Abu Samra. **Montar Product Burndown Chart**. 2007. Disponível em: <<http://blogdoabu.blogspot.com/2007/10/passo-7-montar-product-burndown-chart.html>> Acesso em: 06 ago. 2009.

RAHAL JR, Nelson Abu Samra. **Scrum e Kanban**. 2008. Disponível em: <<http://blogdoabu.blogspot.com/2008/10/scrum-e-kanban.html>>. Acesso em: 30 jan. 2010.

REA, L.M.; PARKER, R.A. **Metodologia de pesquisa: do planejamento a execução**. São Paulo: Pioneira, 2000.

RILDO, F. Santos. **Tutorial Scrum**. s/d. Disponível em: <<http://www.companyweb.com.br/Rildo/scrum/Tutorial-SCRUM-v9.pdf>> Acesso em: out. 2009.

RODRIGUES, Rafael; ROST, Rafael. **SCRUM (Metodologia para o Desenvolvimento Ágil de Software)**. 2007. Disponível em: <<http://rafaelrgi.wordpress.com/2007/11/23/scrum-metodologia-para-o-desenvolvimento-agil-de-software/>> Acesso em: 13 maio 2009.

ROMAN, Milton. **Processo de Certificação SCRUM**. 2009. Disponível em: <http://milton-roman.blogspot.com/>. Acessado em 06 de abril de 2010.

SANTANA, Eudes Raphael de Sá. **Engenharia de requisitos Ágil**. Monografia apresentado na disciplina de Engenharia de Requisitos do centro de Informática da UFPE, 2008.

SANTOS, Rildo F. **Tutorial Scrum**. 2009. Disponível em: <<http://rildosan.blogspot.com/2009/06/scrum-experience-o-tutorial-scrum.html>> Acesso em: 20 jun. 2009.

SANTOS, Wagner Roberto dos. **Estimativas Ágeis - Planning Poker**. 2008. Disponível em: <http://netfeijao.blogspot.com/2008/10/estimativas-geis-planning-poker.html>. Acesso em 30 jan. 2010.

SANTOS, Wagner. **SCRUM - uma abordagem prática**. 2008. Disponível em: <http://netfeijao.blogspot.com/2008/02/scrum-uma-abordagem-prtica.html>. Acesso em: nov. 2009.

SCHAWABER, K. & BEEDLE, M. **Agile software development with scrum**. NJ: Pretence Hall, 2001.

SCHWABER, K. “**Agile Project Management with Scrum**”, Microsoft Press, 2004.

SCHWABER, K. **Agile Project Management With Scrum**, Microsoft, 2004.

SCHWABER, K. **SCRUM Development Process**. 1995. Disponível em <http://jeffsutherland.com/oops/schwapub.pdf>. Acesso em: outubro de 2009.

Scrum Alliance. Acessado em 27 de junho de 2009. Disponível em: <<http://www.scrumalliance.org>>.

SERRA, Paulo Roberto Furtado. **Gerenciamento ágil de projetos com SCRUM**. 2009. Disponível em:< <http://www.slideshare.net/paulofurtado/curso-scrum>> Acesso em: 21 jun. 2010.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

SILVEIRA, Guilherme. **Gerenciamento de projetos com Scrum**. 2007. Disponível em:<<http://blog.caelum.com.br/2007/06/20/gerenciamento-de-projetos-com-scrum/>> Acesso em: nov. 2009.

SOARES, Felipe S. Furtado et al. **Adoção de Scrum em uma fábrica de desenvolvimento distribuído de software**. 2008. Disponível em:< <http://200.17.137.109:8081/xiscanoe/publicacoes/conferences/i-wdds/adocao-de-scrum-em-uma-fabrica-de-desenvolvimento.pdf>> Acesso em: 20 fev. 2009.

SOARES, M. S. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. **INFOCOMP (UFLA)**, v. 3, p. 8-13, 2004. Disponível em:< <http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf>> Acesso em: 12 maio 2009.

SOMMERVILLE, I. **Engenharia de Software**. Tradução André Maurício de Andrade Ribeiro; revisão técnica kecki Hirama. São Paulo: Addison Wesley, 2003.

STANDISH GROUP INTERNATIONAL. **The CHAOS Report**, 1994. Disponível em:< http://www.standishgroup.com/sample_research/chaos_1994_1.php> Acesso em: 14 fev.2009.

STANDISH, The Standish Group International. **The CHAOS Report**. Standish Group. 2004. Disponível em: <http://secure.standishgroup.com/reports/reports.php?rid=500>.

TAKEUCHI H.; NONAKA I. **The new new Product Development Game**. Harvard Business Review, 1986.

TEIXEIRA, Enise Barth. **A análise de dados na pesquisa científica**. Importância e desafios em estudos organizacionais. Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2003. Disponível em:< redalyc.uaemex.mx/pdf/752/75210209.pdf> Acesso em: 21 jun. 2010.

TRIVIÑOS, Augusto Nibaldo Silva. **Introdução à pesquisa em ciências sociais: a pesquisa qualitativa em educação**. São Paulo: Atlas, 1987.

VELAMAKANNI, Bharadwaj; SATHIANARAYANAN, Ramanathan. **Análise Burndown no Gerenciamento de Cronograma e de Produtividade**. 2009. Traduzido por Samuel Carrijo. Disponível em:< <http://www.infoq.com/br/articles/burndown-analysis>>. Acesso em: 30 jan. 2010.

VERGARA, Sylvia Constant. **Projetos e Relatórios de Pesquisa em Administração**. 3. ed. São Paulo: Atlas, 2000

VERSIONONE. **State of Agile Development Survey**. 4th annual, 2009. Disponível em:<<http://pm.versionone.com/StateOfAgileSurvey.html>.> Acesso em: 19 maio 2010.

VERSIONONE. **The State of Agile Development Survey Results**, 2008. Disponível em:<http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.> Acesso em: 29 mar. 2009.

VERSIONONE. **The State of Agile Development**. 2007. Disponível em: <http://www.versionone.com/pdf/StateofAgileDevelopmentSurvey.pdf> Acesso em: 25 de junho de 2009.

VARGAS, Ricardo Viana. **Gerenciamento de projetos: estabelecendo diferenciais competitivos**. Rio de Janeiro: Brasport, 2005.

VIEIRA, Sérgio Roberto Costa. **O desenvolvimento de uma ferramenta web para auxiliar no gerenciamento de riscos em projetos de software**, 2007. Trabalho de conclusão de Curso (Graduação) - Instituto de Ensino Superior Fucapi de Manaus, Manaus, 2007.

VILLAS BÔAS, M. **Um Novo Enfoque para o Gerenciamento de Projeto de Desenvolvimento de Software**. 2005. Dissertação (Mestrado) - Departamento de Administração da Universidade de São Paulo, São Paulo, 2005.

ZANATTA, Alexandre Lazaretti; VILAIN, Patrícia. **Uma análise do método ágil scrum conforme abordagem nas áreas de processo**. 2005. Disponível em:< http://wer.inf.puc-rio.br/werpapers/artigos/artigos_wer05/alexandre_zanatta.pdf.> Acesso em: 14 jul. 2009.

APÊNDICE A – Questionário da pesquisa

Mestrado: Mestrado em Ciência da Computação

Tema da dissertação: Um estudo sobre as principais dificuldades dos Scrum Masters em relação ao uso de ferramentas para auxiliar suas atividades.

Universidade: Universidade Federal de Pernambuco

Orientador: Fernando da Fonseca de Souza, PhD

Aluna: Ana Carolina Oran Fonseca e Toledo

Sobre o Documento

Este questionário deve ser respondido por *Scrum Masters* e seu principal objetivo é coletar informações de forma estruturada a respeito das ferramentas utilizadas por tais profissionais para auxiliar na execução de suas responsabilidades. Além disso, também tem o propósito de levantar as funcionalidades mais relevantes que esse tipo de ferramenta deve disponibilizar.

O questionário é estruturado em quatro partes, as quais são:

Parte I - Identificação do sujeito da pesquisa (informações gerais sobre o profissional entrevistado);

Parte II - Identificação das atividades desempenhadas pelos *Scrum Masters* (informações sobre as atividades cotidianas desempenhada pelos *Scrum Masters* em um projeto *Scrum*);

Parte III - Identificação das ferramentas utilizadas pelos *Scrum Masters* (informações relacionadas às ferramentas usualmente utilizadas pelos *Scrum Masters* para auxiliar a execução de suas atividades, bem como verificar se estas atendem às expectativas desses profissionais); e

Parte IV - Identificação das funcionalidades que uma ferramenta computacional deve disponibilizar para auxiliar os *Scrum Masters* (Se na parte III, for identificado que as ferramentas atualmente utilizadas não atendem às necessidades desses profissionais, será identificado um conjunto mínimo de funcionalidades que tais ferramentas devem apresentar).

SIGILO DAS INFORMAÇÕES

As informações relacionadas à identificação do entrevistado serão tratadas como sigilosas e não serão reveladas sem autorização prévia e por escrito do entrevistado. As informações coletadas serão processadas e apresentadas no trabalho de dissertação e em artigos científicos de forma agregada ou isoladamente, sem a identificação da fonte.

AGRADECIMENTOS

O orientador da pesquisa, Prof. Fernando da Fonseca de Souza, e a aluna de mestrado Ana Carolina Oran Fonseca e Toledo agradecem antecipadamente a sua participação. Os resultados desta atividade da pesquisa serão divulgados a todos os entrevistados.

INSTRUÇÕES DE RESPOSTA

Deve-se assinalar a letra correspondente à opção desejada como resposta.

Parte I - Identificação do sujeito da pesquisa

Nome Completo (opcional):

Empresa (opcional):

Email (desejável):

1. Você possui alguma certificação? Qual? (Pode assinalar mais de uma opção)
 - a) CSM – Certified Scrum Master
 - b) CSPO – Certified Scrum Product Owner
 - c) CSP – Certified Scrum Pratitioner
 - d) CST – Certified Scrum Trainer
 - e) CSC – Certified Scrum Coach
 - f) Outras: _____

2. A quanto tempo você trabalha com a metodologia de gerenciamento SCRUM?
 - a) menos de 1 ano
 - b) 1 a 2,5 anos
 - c) 2,51 a 5 anos
 - d) 5,1 a 7,5 anos
 - e) 7,6 a 10 anos
 - f) Mais de 10 anos

3. A quanto tempo você trabalha como SCRUM MASTER?
 - a) menos de 1 ano
 - b) 1 a 2,5 anos
 - c) 2,51 a 5 anos
 - d) 5,1 a 7,5 anos
 - e) 7,6 a 10 anos
 - f) Mais de 10 anos

4. Você atua em outra função além de *Scrum Master*?
 - a) Sim
 - b) Não

5. Se a resposta da questão anterior for “SIM”, responda: Quais as funções? (Pode assinalar mais de uma opção).
 - a) Gerente de projeto
 - b) Líder da equipe
 - c) Analista de sistemas
 - d) Desenvolvedor

- e) Testador
f) Outros. Qual? _____

Parte II - Identificação das atividades desempenhadas pelos Scrum Masters

As principais atribuições do *Scrum Masters* são listadas a seguir, de acordo com Schwaber (2004):

- Permitir que o *time* seja auto-gerenciável;
- Garantir que os caminhos para a comunicação do *time* estejam abertos permanentemente;
- Remover qualquer impedimento que o *time* encontre;
- Proteger o *time* de interferências externas para garantir que sua produtividade não seja afetada;
- Facilitar as reuniões do projeto (*Planning Meeting*, *Daily Meeting*, *Review e Retrospectives*);
- Garantir que o processo está sendo seguindo - Participando das reuniões diárias, revisão da *Sprint*, e planejamento;
- Garantir que o *time* esteja totalmente funcional e produtivo; e
- Ajudar o *Product Owner* na priorização dos itens de *Backlog*.

Entretanto, acredita-se que na prática, os *Scrum Masters* desempenham atividades que extrapolam as diretrizes citadas acima. Em vista disso, responda a seguinte questão levando em consideração as atividades desempenhadas por você no seu dia-a-dia.

6. No seu cotidiano de Scrum Master, quais as atividades desempenhadas nos projetos das quais participa? (pode escolher mais de uma opção)
- a) Permitir que o *time* seja auto-gerenciável;
 - b) Garantir que os caminhos para a comunicação do *time* estejam abertos permanentemente;
 - c) Remover qualquer impedimento que o *time* encontre;
 - d) Proteger o *time* de interferências externas para garantir que sua produtividade não seja afetada;
 - e) Facilitar as reuniões do projeto (*Planning Meeting*, *Daily Meeting*, *Review e Retrospectives*);
 - f) Garantir que o processo está sendo seguindo. Participando das reuniões diárias, revisão da *Sprint*, e planejamento;
 - g) Garantir que o *time* esteja totalmente funcional e produtivo;
 - h) Ajudar o *Product Owner* na priorização dos itens de *Backlog*;
 - i) Delegar tarefas para o *time*;
 - j) Participar da estimativa dos itens do *Backlog*;
 - k) Facilitar a comunicação entre o *time* e o *Product Owner*;
 - l) Identificar e controlar os riscos do projeto;
 - m) Participar do planejamento do *sprint* em conjunto com o *time*;
 - n) Acompanhar o planejamento do *sprint* a fim de avaliar o planejamento x realizados;
 - o) Traçar estratégias em conjunto com o *time* para remover possíveis atrasos no cronograma;
 - p) Coletar e documentar requisitos do projeto;
 - q) Outras. Quais?

Parte III - Identificação das ferramentas e funcionalidades utilizadas pelos Scrum Masters

O primeiro grupo é composto por ferramentas não-computacionais. Este tipo de ferramenta é caracterizado por representar o produto que está sendo desenvolvido com materiais de fácil aquisição, baixo custo e fácil manipulação. Porém, esta característica de baixo custo implicará em atividades mais longas e resultados não tão satisfatórios em termos de precisão. Ex.: Quadro de Kanban.

O segundo grupo é composto por ferramentas computacionais. Este tipo de ferramenta é utilizado para realizar uma avaliação prévia de sistemas homem-máquina em um tempo menor e com um custo, a médio e longo prazos, também menor em relação às ferramentas não computacionais. Ex.: Excel.

Com base nisto, responda as questões abaixo.

Ferramentas Não-Computacionais

7. Para auxiliar no cumprimento de suas atividades, quais as ferramentas **não-computacionais** que VOCÊ REALMENTE UTILIZA? (pode escolher mais de uma opção)

(OBSERVAÇÃO IMPORTANTE: não inclua ferramentas que você gostaria de usar nesta resposta. Isto será verificado na questão 9)

- a) Kanban
- b) Planning Poker
- c) Grant Chart
- d) Ata de reunião
- e) Memorandos
- f) Burn Down Chart
- g) Gráfico de Ishikawa
- h) Impediment Backlog
- i) Técnica Delphi para estimativas
- j) Product Backlog
- k) Sprint Backlog
- l) Outras: _____

8. Dentre as ferramentas não-computacionais que você utiliza, alguma PODERIA SER DESCARTADA? (Pode assinalar mais de uma opção)

- a) Kanban
- b) Planning Poker
- c) Grant Chart
- d) Ata de reunião

- e) Memorandos
- f) Burn Down Chart
- g) Gráfico de Ishikawa
- h) Impediment Backlog
- i) Técnica Delphi para estimativas
- j) Product Backlog
- k) Sprint Backlog
- l) Outras: _____

9. Existem ferramentas não-computacionais que você não utiliza atualmente, mas que gostaria de fazer uso? Qual(is)?

Ferramentas Computacionais

10. Quais as ferramentas computacionais que VOCÊ REALMENTE UTILIZA em suas atividades de *Scrum Master*? (pode escolher mais de uma opção)
(OBSERVAÇÃO IMPORTANTE: não inclua ferramentas que você gostaria de usar nesta resposta. Isto será verificado na questão 12)

- a) Planilha eletrônica (MS-Excel, BR Calc, Lótus 123 ou similares)
- b) Planejamento, monitoração e controle de atividades (Ms-project, Xplanner ou similares)
- c) Troca instantânea de mensagens (MSN, Gtalk, ICQ ou similares)
- d) Controle de versão (SVN, CVS, Clearcase ou similares)
- e) Workflow (Mantis, Bugzilla ou similares)
- f) Prototipação (PowerPoint, Vision ou similares)
- g) Modelagem (Rational Rose, Jude, Erwin ou similares)
- h) Scrumy
- i) Pango Scrum
- j) Mingle
- k) ScrumWorks
- l) IceScrum
- m) FireScrum
- n) VersionOne
- o) outras: _____

11. Dentre as ferramentas computacionais que você utiliza, alguma PODERIA SER DESCARTADA?

- a) Planilha eletrônica (MS-Excel, BR Calc, Lótus 123 ou similares)
- b) Planejamento, monitoração e controle de atividades (Ms-project, Xplanner ou similares)
- c) Troca instantânea de mensagens (MSN, Gtalk, ICQ ou similares)
- d) Controle de versão (SVN, CVS, Clearcase ou similares)
- e) Workflow (Mantis, Bugzilla ou similares)
- f) Prototipação (PowerPoint, Vision ou similares)
- g) Modelagem (Rational Rose, Jude, Erwin ou similares)

- h) Scrummy
- i) Pango Scrum
- j) Mingle
- k) ScrumWorks
- l) IceScrum
- m) FireScrum
- n) VersionOne
- o) outras: _____

(Existem ferramentas que você não utiliza em seus projetos por alguma restrição, seja de segurança, política da empresa, não é *free*, etc. Se você conhece outras ferramentas, mas não pode utilizar em seus projetos, responda a próxima pergunta).

12. Existem ferramentas computacionais que você não utiliza atualmente, mas que gostaria de fazer uso? Qual(is)?

13. Dentre as funcionalidades que você utiliza nas ferramentas escolhidas na questão 10, quais você gostaria de ter numa única ferramenta integrada?

Parte IV - Identificação das funcionalidades de uma ferramenta computacional para auxiliar os Scrum Masters

14. Baseado na sua experiência, elenque as funcionalidades que deveriam ser fornecidas por uma ferramenta computacional voltada para auxiliar os SM.

Para auxiliar na análise de suas respostas, responda a questão e assinale o nível de prioridades destas funcionalidades em: **ES**- Essencial, **IM** – Importante, **DS** – Desejável.

Exemplo:

Funcionalidade	ES	IM	DS
<i>Geração do burn down chart</i>	X		
<i>Calculo do tempo total do projeto</i>		X	
<i>Distribuição de atividades para os componentes da equipe</i>			X
<i>Ambiente de simulação do planning poker</i>			X
<i>Kanban</i>			X

Funcionalidade	ES	IM	DS

Comentários em geral:

APÊNDICE B – Questionário de satisfação

Sobre o Documento

Este questionário tem como principal objetivo avaliar a capacidade da ferramenta JustScrum de fornecer as funções as quais satisfazem as necessidades dos Scrum Masters em suas atividades diárias em um projeto Scrum.

1. Verifique as funcionalidades da ferramenta e marque se satisfaz ou não as suas necessidades.

FUNCIONALIDADES	SIM	NÃO
Permite o controle do status das user stories?		
Permite fazer o cronograma do sprint?		
Permite distribuir as atividades para os componentes da equipe?		
Possibilita a geração do burn down chart do projeto?		
Possibilita a geração do burn down chart da sprint?		
Possibilita o cadastro de risco?		
Permite o gerenciamento de risco?		
Possibilita o cadastro dos impedimento?		
Permite o gerenciamento do Impediment Backlog?		
Permite a visualização de um taskboard?		
Possibilita fazer o planejamento e controle dos Release?		
Permite o cadastro de itens no Product Backlog?		
Possibilita a priorização dos itens do Product backlog?		
Permite criar o registro das lições aprendidas?		
Permite criar o registro do tempo gasto e restante para cada tarefa?		
Permite criar o registro de eventos do sprint?		
Possibilita fazer o cadastro de Sprint Backlog?		
Permite fazer o planejamento da sprint?		

2. Verifique a capacidade da ferramenta com relação a usabilidade e portabilidade.

USABILIDADE E PORTABILIDADE	SIM	NÃO
A ferramenta é fácil de usar?		
Aprende-se facilmente a navegar na ferramenta?		
A ferramenta é demasiado lento a responder aos comandos?		
A ferramenta é fácil de instalar?		

3. Quais funcionalidades que você gostaria de ter adicionada a ferramenta JustScrum?

4. Atribua uma classificação de qualidade global da ferramenta JustScrum.

Excelente	Bom	Razoável	Fraco

Obrigada!

APÊNDICE C – Instrução de configuração da ferramenta JustScrum

Para o funcionamento adequado da ferramenta JustScrum é necessário instalar quatro (4) ferramentas, que são:

1. Postgresql

- Faça o download no endereço:

<ftp://ftp3.br.postgresql.org/pub7/postgresql/binary/v8.2.16/win32/postgresql-8.2.16-1.zip>

- Descompactar na maquina local;

- Passos para instalação:

Na seção *Service Configuration*, informe:

Account name: *postgres*

Account password: *Informe sua senha*

Na seção *Initialise database cluster*, informe:

Port number: *5432*

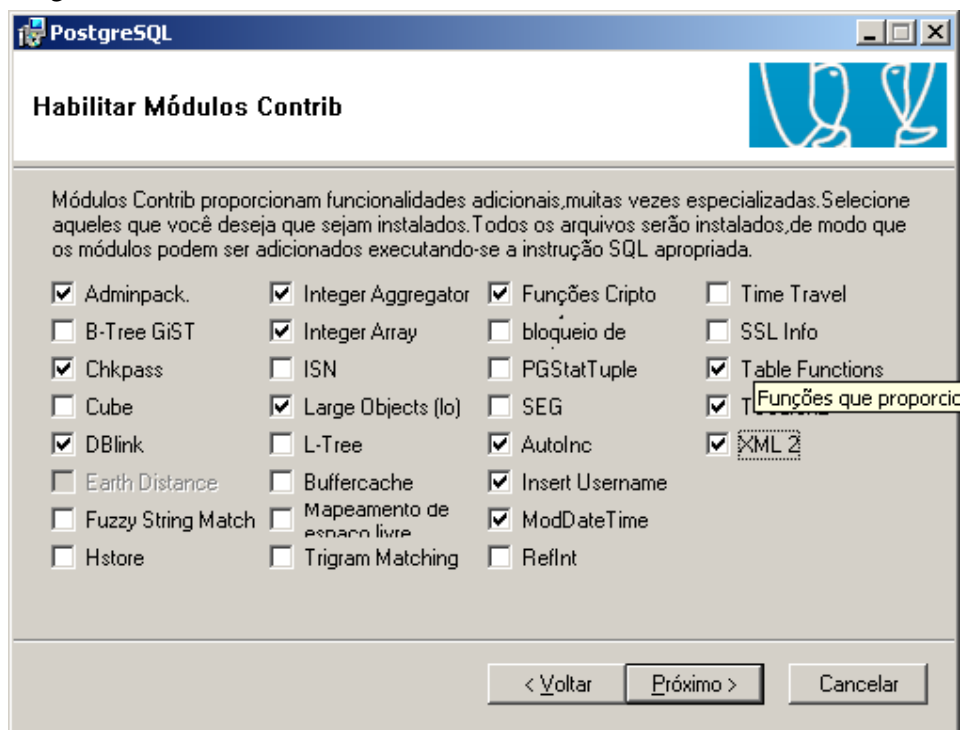
Locale: *Portuguese, Brazil*

Encoding: *LATIN1*

Superuser name: *root*

Password: *informe sua senha.*

Na seção *Enable contrib modules*, selecione as seguintes opções de acordo com a imagem abaixo:



- Prossiga com a instalação normalmente.

2. Jre (java Runtime Environment)

- Faça o download no endereço:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk6-jsp-136632.html>

- Passos para instalação:

Siga os passos de acordo com a ferramenta;

Na seção *custom setup*, selecione:

Opção: public jre.

3. JBoss-Server

- Faça o download no endereço:
<http://sourceforge.net/projects/jboss/files/JBoss/JBoss-4.2.2.GA/jboss-4.2.2.GA.zip/download>
- Descompactar no diretório raiz;

4. Firefox

- Faça o download no endereço:
<http://mozilla2.snt.utwente.nl/firefox/releases/3.6.8/win32/pt-BR/Firefox%20Setup%203.6.8.exe>
- Executar a instalação seguindo os passos normalmente.

Configuração do JustScrum

Para configurar e começar a utilizar o JustScrum siga os seguintes passos:

1. Configurando o banco de dados do JustScrum

Acesse o aplicativo de administração do postgres (pgAdminIII);
 Crie uma novo banco com o seguinte nome JustScrum;
 Rode os scripts para que sejam criadas as tabelas do sistema, encontrados no endereço:
<http://sourceforge.net/projects/justscrum/files/scripts>

2. Adicionando o aplicativo ao servidor de aplicação

Copiar a versão mais atual do JustScrum no endereço:
<http://sourceforge.net/projects/justscrum/files/versaoatual>
 Copiar a versão mais atual no diretório:
 C:\servidorAplicacao\server\default\deploy\easyScrum
 Adicione no mesmo diretório o arquivo de conexão com o banco de dados encontrado no
<http://sourceforge.net/projects/justscrum/files/arqbd>
 Em seguida edite o mesmo e adicione a url onde está instalado o respectivo banco de dados com senha e usuário de conexão.

3. Iniciando o servidor de aplicação

Acesse o prompt do DOS, acesse o diretório onde está instalado o jboss-server e acesse a pasta bin.

Logo após, execute o seguinte comando `run.bat -b 0.0.0.0` e aguarde que o servidor seja iniciado.

4. Acessando a aplicação

Execute o Firefox e informe o seguinte endereço:
<http://NomeDoServidor:8080/JustScrum/layout/login.seam>