## MARVIN FERREIRA DA SILVA

ANÁLISE DA ADERÊNCIA DE PRÁTICAS ÁGEIS NA CULTURA DE STARTUPS DE SOFTWARE: O MAPEAMENTO DE PRÁTICAS ATRAVÉS DO SEMAT KERNEL

## MARVIN FERREIRA DA SILVA

# ANÁLISE DA ADERÊNCIA DE PRÁTICAS ÁGEIS NA CULTURA DE STARTUPS DE SOFTWARE: O MAPEAMENTO DE PRÁTICAS ATRAVÉS DO SEMAT KERNEL

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para a obtenção do título de Mestre em Engenharia Elétrica.

#### MARVIN FERREIRA DA SILVA

# ANÁLISE DA ADERÊNCIA DE PRÁTICAS ÁGEIS NA CULTURA DE STARTUPS DE SOFTWARE: O MAPEAMENTO DE PRÁTICAS ATRAVÉS DO SEMAT KERNEL

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia

Área de Concentração: Engenharia da Computação

Orientador: Prof. Dr. Edison Spina

#### Catalogação-na-publicação

Silva, Marvin Ferreira da

Análise da aderência de práticas ágeis na cultura de startups de software: o mapeamento de práticas através do SEMAT Kernel / M. F. Silva - São Paulo, 2016.

133 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Engenharia de software 2.Métodos ágeis 3.Startup 4.SEMAT I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

# **DEDICATÓRIA**

À minha mãe, por seu exemplo de luta.

#### **AGRADECIMENTOS**

À minha mãe, Dolores, por todo apoio e carinho.

Ao meu orientador Edison Spina, pela oportunidade, orientação e seu exemplo acadêmico e profissional.

Aos colegas de laboratório Lucas Lago, Marcel J. Simonette e Rodrigo Filev Maia por suas opiniões e contribuições ao trabalho.

Ao professor Luiz Barco, por seus conselhos e inspirações.

À todos que ao longo dessa trajetória contribuíram para que eu me tornasse não apenas um profissional melhor, mas um humano melhor.

Ao Centro de Estudos Sociedade e Tecnologia (CEST) da Universidade de São Paulo por ter patrocinado parcialmente este trabalho.

É melhor tentar e falhar, que preocupar-se e ver a vida passar. É melhor tentar ainda em vão, que sentar-se fazendo nada até o final. Eu prefiro na chuva caminhar, que em dias tristes em casa me esconder. Prefiro ser feliz, embora louco, que em conformidade, viver. (Martin Luther King)

#### **RESUMO**

Este trabalho analisa os principais métodos ágeis utilizados em empresas startup, como scrum, extreme programming, kanban e lean, isolando suas práticas e mapeando-as no Kernel do SEMAT para escolher os elementos essenciais da engenharia de software que estão relacionados a cada prática de forma independente. Foram identificadas 34 práticas que foram reduzidas a um conjunto de 26 pelas similaridades. Um questionário foi desenvolvido e aplicado no ambiente de startups de software para a avaliação do grau de utilização de cada determinada prática. Através das respostas obtidas foi possível a identificação de um subconjunto de práticas com utilização acima de 60% onde todos os elementos essenciais da engenharia de software são atendidos, formando um conjunto mínimo de práticas capazes de sustentar este tipo específico de ambiente.

#### PALAVRAS CHAVE:

Engenharia de software, desenvolvimento ágil, startup, SEMAT.

#### **ABSTRACT**

This work analyzes the main agile methods used in startup companies such as scrum, extreme programming, kanban and lean, isolating their practices and mapping them on SEMAT Kernel to choose the essential elements of software engineering that are related to each practice independently. 34 practices were indentified and reduced to a set of 26 by similarities. A questionnaire was developed and implemented in software startups environment to evaluate the usage degree of each particular practice. Through the responses obtained it was possible to identify a subset adopted by more than 60% of respondents where all the essential elements of software engineering are met, forming a minimum set of practices capable of sustaining this specific type of environment.

#### **KEY WORDS:**

Software engineering, agile development, startup, SEMAT.

# Lista de Figuras

Figura 1 - Classificação de empresas por quantidade de colaboradores	21
Figura 2 - Modelo sequencial proposto por Royce que deu origem ao Waterfall	31
Figura 3 - Execução do processo scrum	42
Figura 4 – Práticas da metodologia XP	44
Figura 5 - Fluxo em um painel Kanban	48
Figura 6 – Arquitetura do método SEMAT	54
Figura 7 - Dimensões do SEMAT Kernel	55
Figura 8 – Alphas que constituem a Essência	57
Figura 9 - Estados do alpha Requisitos	58
Figura 10 - Mapa de Práticas Ágeis	60
Figura 11 - SEMAT como ferramenta de controle	
Figura 12 - Espaços de Atividades dos três domínios (Coisas que sempre fazemo	os)
Figura 13 - Mapeamento de produto de trabalho e atividade	69
Figura 14 - Tabela de adjacência para análise por grafo	

# Lista de Gráficos

Gráfico 1- Taxa de Sucesso, Falha e Desafio em projetos de Software	33
Gráfico 2 - Distribuição da adoção de processos ágeis ao longo dos anos	
Gráfico 3 - Distribuição da adoção de processos ágeis	39

## Lista de Quadros

Quadro 1 - Princípios Lean	53
Quadro 2 - Mapeamento de práticas no SEMAT Kernel	66
Quadro 3 - Mapeamento de práticas ágeis e espaços de atividades do domínio Clier	nte 72
Quadro 4 - Mapeamento de práticas ágeis e espaços de atividades do domínio Solução	73
Quadro 5 - Mapeamento de práticas ágeis e espaços de atividades do domínio Empreitada	74
Quadro 6 - Percentuais de utilização de práticas	
Quadro 7 - Divisão de práticas por seu percentual de utilização	82
Quadro 8 - Mapeamento da cobertura de práticas por utilização nas atividades do domínio Cliente	84
Quadro 9 - Mapeamento da cobertura de práticas por utilização nas atividades do domínio Solução	
Quadro 10 - Mapeamento da cobertura de práticas por utilização nas atividades do domínio Empreitada	
Quadro 11 - Relação entre práticas e suas questões para escala Likert 1	

# Lista de Tabelas

Tabela 1 - Taxa de mortalidade de startups por tempo de vida	24
Tabela 2 - Fatores que contribuem para a sobrevivência de uma startup	
Tabela 3 - Fatores que contribuem para a descontinuidade de uma startup	26
Tabela 4 - Adoção de Práticas Ágeis	60
Tabela 5 - Descrição de práticas contidas nos métodos eleitos	64
Tabela 6 - Experiência na função desempenhada	77

# SUMÁRIO

DEI	DICATÓRIA	]
AGI	RADECIMENTOS	I
RES	SUMO	IV
ABS	STRACT	V
Lista	a de Figuras	V
Lista	a de Gráficos	VI
Lista	a de Quadros	VII
Lista	a de Tabelas	IX
SUN	MÁRIO	X
1	INTRODUÇÃO	
1.1	OBJETIVOS	15
	1 Objetivo Geral	
	2 Objetivos Específicos	
	ORGANIZAÇÃO	
1.3	TERMINOLOGIA	
2	STARTUP	
	3	
	1 Características de uma Startup	
	DESCONTINUIDADE OU FIM DE UMA STARTUP	
2.2.	1 Aspectos Significativos para a Descontinuidade de uma Startup	24
2.3	PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE EM START	
3	PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE	
3.1	CONTEXTO HISTÓRICO E A CRISE DO SOFTWARE	
	PROCESSOS DE DESENVOLVIMENTO ÁGIL DE SOFTWARE	
	1 Adoção dos Processos Ágeis	
	2 Scrum	
	3 Extreme Programming	
	4 Sistema Kanban	
	5 Desenvolvimento de Software Lean	
	Software Engineering Method and Theory (SEMAT)	
	1 Organização do Kernel	
3.3.2	2 Os Alphas	56
4	SELEÇÃO DE PRÁTICAS ÁGEIS E MAPEAMENTO ATRAVÉS DA	
	ESSÊNCIATAXONOMIA DE PRÁTICAS ÁGEIS	59
4.1	TAXONOMIA DE PRATICAS AGEIS	59
4.2	MAPEAMENTO DE PRÁTICAS NO SEMAT KERNEL	65
	MAPEAMENTO DE PRÁTICAS ATRAVÉS DA ESSÊNCIA	
	1 SEMAT COMO UMA FERRAMENTA DE CONTROLE PARA PROJE	
4.0	DE SOFTWARE2 COMPOSIÇÃO DE PRÁTICAS COM SEMAT KERNEL	66
4.3.2	2 COMPOSIÇÃO DE PRÁTICAS COM SEMAT KERNEL	67
	3 ELEIÇÃO DE PRÁTICAS PARA STARTUPS 4 PRÁTICAS VS "COISAS A SE FAZER"	
4 3 4	4 PRATICAS VS "COISAS A SEFAZER"	- 71

5 AVALIAÇÃO DE ADERÊNCIA DAS PRÁTICAS ÁGEIS	75
5.1 ESTUDO DE CASO	75
5.2 MÉTODO	75
5.3 ANÁLISE DOS RESULTADOS	
5.3.1 ANÁLISE DESCRITIVA	77
5.3.2 DIVISÃO POR UTILIZAÇÃO	82
5.4 DISCUSSÃO	87
6 CONSIDERAÇÕES FINAIS	90
6.1 CONTRIBUIÇÕES	90
6.2 TRABALHOS FUTUROS	91
REFERÊNCIAS	93
APÊNDICE A – QUESTIONÁRIO	100
APÊNDICE B – QUESTÕES E SUAS RESPECTIVAS PRÁTICAS	108
APÊNDICE C – RESPOSTAS DOS CANDIDATOS	111
APÊNDICE D – ANÁLISE GRÁFICA DAS RESPOSTAS	120

## 1 INTRODUÇÃO

O crescimento da indústria de software desencadeou o surgimento e de novas empresas e novos formatos de negócios. Muitas dessas empresas são pequenas empresas de alta tecnologia com grandes restrições de recursos financeiros e de pessoal e estão em busca de um modelo de negócio escalável e sustentável. Estas empresas são denominadas *startups*, que, apesar de tais restrições e riscos, geram grande impacto econômico devido ao seu alto nível de inovação. As *startups* são empresas diferenciadas e não foram consideradas pelas atuais práticas de engenharia de software (FAYAD, ET AL., 2000) (GIARDINO; ET AL, 2014).

Estas *startups* representam um seguimento da indústria de software que vem sendo negligenciado pelos estudos sobre processos e que possíveis lições aprendidas neste segmento podem ser aplicadas para outros tipos de organizações (SUTTON, 2000) (GIARDINO; ET AL, 2014).

Apenas nos Estados Unidos, 476.000 novas empresas são criadas a cada mês (FAIRLIE, 2013), respondendo por quase 20% da criação de novos postos de trabalho (FAIRLIE, 2014) e segundo Nobel (2011), 65% das Startups não sobrevivem aos primeiros cinco anos de existência e 75% das *startups* que recebem investimentos de *venture capital* falham.

Apesar da alta taxa de mortalidade entre empresas startups, incluindo as que recebem investimento, ainda não se sabe até que ponto as práticas da engenharia de software impactariam esta taxa, talvez em decorrência de pesquisas neste tema ainda serem prematuras (GIARDINO; ET AL, 2014).

Segundo Highsmith e Cockburn (2001), em muitos projetos de software não é mais possível encontrar seu plano original para ser analisado, pois, ser conforme a um plano não é mais um requisito principal de projeto. A satisfação do cliente e a entrega de valor são os objetivos de maior importância, em detrimento do gasto de esforço com a produção de artefatos de documentação desnecessários, que são exigidos em métodos pesados, e no passado eram utilizados para indicar progresso

no trabalho. Algo que não necessariamente é verdade pois um documento não traz retorno do investimento direto, algo que uma funcionalidade implementada e executável é capaz de trazer.

A priorização da satisfação do cliente na hora da entrega acompanhada da rápida resposta as mudanças é um dos principais fatores que diferem as novas abordagens de desenvolvimento de software das tradicionais que defendem a redução de custos através da antecipação de todos os requisitos visando à eliminação de mudanças no decorrer do projeto.

Desenvolver um projeto incapaz de atender às mudanças durante seu desenvolvimento significa negligenciar a mutabilidade inerente dos requisitos do negócio devido à alta competitividade de mercado, que é um fator considerado crucial para o sucesso de qualquer negócio nos tempos atuais.

Reconhecendo esta mudança e o que ela representa para as novas formas de desenvolvimento de software que surgiam na época, em 2001, um grupo formado pelos principais idealizadores das práticas que iam ao encontro destes ideais escreveram o Manifesto Ágil (BECK; BEEDLE; BENNEKUM, 2001). Este manifesto defende, acima de tudo, a colaboração, interação, transparência e cooperação entre todos os envolvidos no desenvolvimento de software, visando sempre o aumento do valor na entrega do que é produzido para os *stakeholders*.

Os problemas encontrados em *startups* diferem dos encontrados em outras empresas, por estarem em fases diferentes na construção do negócio. Alguns dos problemas eficientemente resolvidos pelos métodos ágeis podem nem existir em *startups* de tecnologia, tornando um método ágil aplicado de forma rígida a uma *startup* algo ineficiente e a utilização de todas as práticas contidas em um determinado método algo desnecessário (YAU; MURPHY, 2013).

Ainda segundo Yau e Murphy (2013), algumas práticas ágeis, como TDD (*Test Driven Development*) (BECK, 2002) e *Pair Programming* (BECK, 1999) são escolhidas na relação entre o custo e o tempo de trabalho para se obter aumento na qualidade de software. Algumas dessas práticas podem apresentar altos custos em

tempo de execução e em recursos utilizados o que pode não ser facilmente sustentado por empresas jovens.

Entre estes objetivos é possível citar a produção de um protótipo com maior número de funcionalidades para a conquista de mais usuários, ou um Mínimo Produto Viável (MVP) como é chamado pelo Método *Lean Startup* (RIES, 2011), para a busca de investidores que ajudem o crescimento dos negócios.

A definição do processo de desenvolvimento de *software* usado em uma *startup* é um ponto importante para a sua sobrevivência, pois o sua sobrevivência muitas vezes, depende de parcerias com organizações já consolidadas como fornecedores de hardware e software, empresas terceirizadas prestadoras de serviços e integradores de sistemas que alavancarão seu crescimento (SUTTON, 2000).

No entanto, para que essas parcerias ocorram é necessário que o processo de desenvolvimento de *software* seja conhecido e possa ser compartilhado com estes parceiros, para que ambas as partes conheçam suas capacidades e limitações e possam cooperar de forma clara e objetiva, de modo que cada uma alcance seus próprios objetivos.

O fato dos estudos relacionados aos processos de desenvolvimento de software com aplicação em startups ainda serem negligenciados, como citado por Sutton (2000) e Giardino, et al. (2014), mostra ser necessária uma maior exploração dos processos da engenharia de software aplicados a este tipo de empresa.

Segundo Yau e Murphy (2013), afirmam que a filosofia ágil é a que melhor se encaixa no contexto real destas empresas, mas que excedem em sua utilidade, não sendo necessários todo o ferramental fornecido neste tipo de ambiente.

#### 1.1 OBJETIVOS

Neste item serão apresentados o objetivo geral e os objetivos específicos desta pesquisa de mestrado.

Objetiva-se através desta pesquisa analisar os principais métodos ágeis a fim de identificar quais de suas práticas são melhor aplicáveis às empresas do tipo *startup*, propondo uma melhor utilização destas práticas de acordo com as restrições dessas empresas para que seja possível validar a sua aderência a ambientes reais e com o intuito de validar a hipótese de que existe um conjunto adequado de práticas que o melhor método ágil para uma *startup* é um método composto por diferentes práticas que respeitem suas restrições (YAU; MURPHY, 2013).

### 1.1.1 Objetivo Geral

O objetivo geral desta pesquisa é identificar quais práticas dos métodos ágeis são mais utilizadas no contexto de empresas *startups* através da validação da aderência de práticas independentes de métodos em ambientes reais de desenvolvimento.

Para que este objetivo geral seja alcançado foram definidos objetivos específicos.

## 1.1.2 Objetivos Específicos

Os objetivos específicos são:

- Analisar os principais métodos ágeis quanto à aplicabilidade de suas práticas em um ambiente com as restrições de uma empresa startup;
- Mapear as práticas mais aplicadas de forma que atendam minimamente as nuances de uma empreitada de desenvolvimento de software;
- Analisar a aplicabilidade das práticas elencadas em cenários reais de desenvolvimento através de um questionário;
- Propor uma forma de utilizar estas práticas de acordo com os resultados obtidos através deste estudo;

## 1.2 ORGANIZAÇÃO

Esta dissertação está organizada em seis capítulos:

- Capítulo 1: introdução, objetivos e organização da dissertação;
- Capítulo 2: definição de uma empresa startup, sua origem, restrições e as características que a diferem dos demais tipos de empresas existentes;
- Capítulo 3: conceitos sobre desenvolvimento ágil e seu contexto histórico, assim como os principais métodos utilizados;
- Capítulo 4: apresentar a taxonomia das práticas ágeis elencadas assim como seu mapeamento no SEMAT kernel;
- Capítulo 5: desenvolvimento e aplicação do questionário criado para a análise das práticas mais utilizadas e uma possível propostas de composição de práticas para este tipo de empresa;
- Capítulo 6: considerações finais, contribuições e as propostas de trabalhos futuros.

#### 1.3 TERMINOLOGIA

Abaixo estão os termos utilizados nesta pesquisa para que não hajam equívocos em sua interpretação.

Prática: Técnica bem definida, ou um conjunto de regras a serem seguidas.

**Método**: Um conjunto de práticas que descreve etapas a serem seguidas para que algum objetivo específico seja alcançado.

Alphas (do ESSENCE): Acrônimo de Abstract-Level Progress Heath Attribute, no português livre, Atributo de Progresso de Saúde de Nível Abstrato, são os elementos essenciais presentes em uma empreitada de software.

**Domínios (do ESSENCE):** São as três áreas específicas do *Kernel*, cada uma focando em um domínio específico do desenvolvimento de *software*.

**Venture Capital:** financiamento de capitais próprios de empresas que fazem investimento de alto risco em empresas jovens com grande potencial de crescimento para que se desenvolvam e ganhem mercado.

**Stakeholders**: uma pessoa ou grupo de interessados em uma empresa ou negócio que fez um investimento e espera um retorno por este investimento.

**Design First**: técnica onde todo o projeto é analisado e modelado antes de iniciarem qualquer etapa de codificação.

#### 2 STARTUP

Nesse capítulo será apresentado o conceito de empresa startup de software, sua origem, características que a diferem dos demais tipos de empresas e uma visão sobre a aplicação de processos de desenvolvimento de software em seu contexto.

## 2.1 DEFINIÇÃO

O termo *startup* ou *start-up*, expressão em inglês de origem estadunidense que será utilizado sem alterações em português nesse texto, significa começar ou iniciar algo e é o termo utilizado para se referir a pequenas empresas inovadoras que buscam, em um ambiente de extrema incerteza e alto risco, produzir um produto ou serviço que solucione um problema ou proponha uma nova solução para um problema já solucionado.

Este termo tornou-se muito conhecido durante a bolha da internet, período que cobriu aproximadamente os anos de 1997 e 2000, quando muitas pequenas empresas baseadas no mercado de internet surgiram e movimentaram grandes quantias em dinheiro de investimento. Durante este período, muitas empresas que apenas adicionavam o prefixo "e-" no início de seu nome ou apenas um ".com" ao final, conseguiam aumentar o preço de suas ações na bolsa de NASDAQ, devido às oportunidades de grandes negócios que o crescimento da internet causou na época. Neste mesmo período, muitas dessas empresas faliram por não conseguirem manter-se no mercado por conta de seus modelos de negócios insustentáveis (PERKINS, A; PERKINS, M, 1999).

Existem algumas definições para o significado de startup, no entanto, todas se baseiam pelos aspectos de risco, inovação e grandes desafios. Abaixo algumas definições para uma empresa startup:

"Uma pequena companhia explorando novas oportunidades de negócios, trabalhando para resolver problemas onde a solução não é bem conhecida e o mercado é altamente volátil" (GIARDINO; ET AL, 2014).

"Um grupo de pessoas iniciando uma empresa, trabalhando com uma ideia diferente, escalável e em condições de extrema incerteza." (SEBRAE, 2014).

"Uma instituição humana designada para criar um novo produto ou serviço sobre condições de extrema incerteza." (RIES, 2011).

Uma startup, resumidamente, atua como um pequeno grupo de pesquisa e desenvolvimento buscando novos produtos capazes de atingir muitos usuários buscando criar um modelo de negócio sustentável.

### 2.2.1 Características de uma Startup

Segundo Sutton (2000), uma *startup* é caracterizada por quatro aspectos principais referentes aos desafios enfrentados tanto por aspectos da engenharia como de negócio:

- Jovens e imaturas como característica mais básica, uma startup é uma empresa nova, ou ao menos relativamente nova e sem experiência em relação a outros tipos de empresa. Isso significa que elas têm pouca experiência acumulada, história ou know-how em sua área de atuação;
- Recursos limitados outra característica típica é a de que os recursos da empresa são limitados. Os primeiros recursos que este tipo de empresa investe são na produção, na promoção do produto e na construção de alianças
- Influências múltiplas em seus primeiros estágios, a startup também pode ser influenciada de diversas maneiras: investidores, clientes, parceiros e por outras empresas que competem no mesmo mercado, atuais ou potenciais;
- Tecnologias e mercados dinâmicos startups geralmente surgem com as mudanças na área de tecnologia, seja com novas tecnologias de rede, aumento na variedade de dispositivos computacionais, novas linguagens de

programação ou qualquer outra mudança que ocorra e proporcione novas possibilidades de negócios. Startups de software geralmente são criadas para desenvolverem tecnologias inovadoras e isso requer o que há de mais atual quanto a ferramentas e técnicas de desenvolvimento.

Dentre as características citadas, duas se sobressaem com relação aos desafios enfrentados: a imaturidade e os recursos limitados das *startups*. Com relação à imaturidade, o desafio de uma *startup* inicia-se com uma ideia e pela definição do que será feito e qual utilidade terá o produto ou serviço que será oferecido. Neste ponto o desafio recai sobre a avaliação da ideia proposta para o surgimento da *startup* e o principal risco envolvido é da aceitação da ideia pelos clientes alvo, pois ainda que inovador, um produto ou serviço deve ser principalmente atrativo. Autores da área sugerem que esta etapa se inicie com a escrita de um plano de negócios que ajudará os envolvidos no projeto a terem uma visão melhor de seus objetivos e façam um refinamento da ideia.

Uma das ideias principais de uma *startup* possuir um bom plano de negócios é poder alinhá-lo com o desenvolvimento do produto e buscar a obtenção de recursos financeiros através de possíveis investidores para o projeto, impulsionando seu desenvolvimento e aumentando sua chance de sucesso ao entrar no mercado, o que também envolve parcerias com empresas já consolidadas que também impulsionarão seu crescimento.

Com relação a seus recursos, uma *startup* normalmente inicia sem recurso financeiro algum, limitando-se apenas aos recursos humanos envolvidos no desenvolvimento do projeto, compondo sua força inicial de trabalho. Não possuir recursos financeiros para o início do projeto faz com que os envolvidos busquem outras formas para compensar este aspecto, procurando alternativas que reduzam o custo de desenvolvimento como a utilização de ferramentas, plataformas e linguagens de programação *open source*.

De acordo com a classificação de empresas por quantidade de colaboradores do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE, 2013) Figura 1, uma startup se enquadra como uma microempresa de até nove colaboradores,

podendo em casos mais específicos e dependendo do nicho em que ela está envolvida ser uma empresa de pequeno porte, ou apenas considerada uma pequena empresa.

Porte	Comércio e Serviços	Indústria
Microempresa (ME)	Até 9 empregados	Até 19 empregados
Empresa de Pequeno Porte (EPP)	De 10 a 49 empregados	De 20 a 99 empregados
Empresa de médio porte	De 50 a 99 empregados	De 100 a 499 empregados
Grandes empresas	100 ou mais empregados	500 ou mais empregados

Figura 1 - Classificação de empresas por quantidade de colaboradores Fonte: SEBRAE, 2013

Apesar de uma *startup* poder ter o mesmo número de colaboradores de uma pequena empresa comum, elas são distintas em suas características: enquanto o foco de uma startup é encontrar um negócio escalável, que envolve alto risco, pouco ou nenhum conhecimento de um mercado normalmente não explorado por propostas altamente inovadoras; o outro tipo surge pelo aumento de demanda em um mercado já existente ou para concorrer com outras empresas por uma parte do mercado.

No caso de uma empresa que não se trata de uma *startup*, seu risco pode ser avaliado analisando outras empresas presentes no mercado, sendo possível realizar um estudo de viabilidade do empreendimento e principalmente buscar *know how* com empresas já situadas no mercado ou apoio de instituições como o SEBRAE para a criação da empresa.

#### 2.2 DESCONTINUIDADE OU FIM DE UMA STARTUP

Segundo dados apresentados, 65% das *startups* não sobrevivem aos primeiros cinco anos de existência e 75% das *startups* que recebem investimentos de venture capital falham (NOBEL, 2011).

Segundo Giardino (2014), grande parte desta taxa de falha se deve ao alto risco envolvido, à perda do momento certo de entrar no mercado e outras questões relacionadas aos negócios.

Segundo pesquisa recente do Núcleo de Inovação e Empreendedorismo da Fundação Dom Cabral (2014), foi realizada uma nova pesquisa na tentativa de compreensão das causas da mortalidade de startups brasileiras. Essa nova pesquisa foi motivada pela falta de acesso à informação sobre empreendedores que tiveram experiências de insucesso como mostrou uma pesquisa similar em 2012, que visava mapear os atores que compõem o ecossistema de empreendedorismo de startups no país e suas características.

Ainda segundo a mesma pesquisa realizada em 2014, pela Fundação Dom Cabral, um dos fatores que indicam a falta de informação quanto a este ponto é a não aceitação do fracasso; fator que atua como barreira cultural bloqueando o empreendedorismo em série; diferente da cultura estadunidense, onde o fracasso em uma empresa startup é visto como símbolo de prestígio e orgulho, significando que o empreendedor que fracassou terá mais experiência e bagagem para seguir com uma nova empreitada empreendedora, fazendo com que aumentem suas chances de sucesso.

O mesmo estudo citado anteriormente investigou as características das *startups* através de duas perspectivas, uma a partir da análise do empreendedor e outra partindo das características das empresas investigadas, correlacionando-as ao ambiente de negócios no momento de sua criação.

Com relação às características capazes de influenciar o sucesso ou fracasso, para o empreendedor obteve-se:

- Nível de Escolaridade
- Ser membro de uma família empreendedora
- Capacitação/experiência em gestão
- Experiência anterior no negócio
- Identificação pessoal dos empreendedores com seu negócio
- Dedicação em tempo integral
- Sintonia entre sócios
- Sintonia entre sócios e investidores
- Capital social (networking)

#### E para a empresa obteve-se:

- Utilização de consultorias externas
- Número de sócios envolvidos
- Qualificação da equipe de trabalho
- Mercado como principal fonte de financiamento
- Volume de capital investido
- Facilidade de produção
- Capacidade de adaptação do modelo de produção
- Capacidade de adaptação do modelo de negócio
- Aceitação do produto pelo mercado

A relação entre a taxa de mortalidade de startups e seu tempo de vida pode ser observada na Tabela 1, que apresenta os dados obtidos pela pesquisa citada:

Tempo de vida (menor ou igual à)	Taxa de mortalidade
1 ano	25%
4 anos	50%
13 anos	75%

Tabela 1 - Taxa de mortalidade de startups por tempo de vida Fonte: Fundação Dom Cabral, 2014

Dados desta pesquisa constataram que pelo menos 25% das *startups* sucumbem antes de completarem um ano, seguido de 50% da morte para até quatro anos de existência e havendo 75% da morte das startups com tempo de vida de até treze anos.

Ainda não se sabe até que ponto as boas práticas da engenharia de software impactariam esta alta taxa de falha, pois as pesquisas neste tema ainda são prematuras (GIARDINO, 2014).

# 2.2.1 Aspectos Significativos para a Descontinuidade de uma Startup

Devido à escassez de recursos e a falta de pessoal, considera-se para o sucesso de uma *startup* que seus integrantes devam desempenhar diversos papéis em sua estrutura, por exemplo, espera-se que uma pessoa cuide do atendimento ao cliente, do marketing e da prospecção de novos clientes e que esse acúmulo de habilidades gerais das pessoas envolvidas possam suprir o déficit de mão de obra.

No entanto, segundo dados da Fundação Dom Cabral (2014), quanto maior for o quadro societário da *startup* maior será a chance de descontinuidade, com um fator de aumento de 1,24 para cada sócio no momento de sua constituição.

Ainda segundo a mesma pesquisa:

- Há indícios de que este insucesso se deva à menor capacidade dos gestores se adaptarem às mudanças e necessidades do mercado além dos problemas de relacionamento entre os sócios;
- O volume de capital que é investido na startup antes dela possuir suas próprias vendas pode se apresentar como um fator de risco à sua sobrevivência. Sendo a chance de descontinuidade de uma startup que o capital investido cobre seus custos operacionais de dois meses a um ano ser 3,2 vezes maior do que as startups que se mantem por apenas um mês e de duas a 2,5 vezes maior do que as que podem se manter por mais de um ano;
- A influência do ambiente em que a startup se encontra também se mostrou um fator importante, mostrando que se a empresa estiver instalada em uma aceleradora, incubadora, ou parque tecnológico terá 3,45 vezes mais chance de sucesso.

Também foram analisadas as percepções entre os empreendedores que já tiveram alguma experiência com descontinuidade de uma *startup* e os que não tiveram. Esse resultado pode ser observado nas Tabelas 2 e 3. Pode-se perceber que, para os dois casos, as opiniões dão ênfase às questões relacionadas aos fundadores da empresa e isso reforça o fato da importância do alinhamento e bom relacionamento entre as pessoas que constituem uma *startup*.

## Startups em operação

Fatores mais importantes para a sobrevivência	Fatores menos importantes para a sobrevivência
Aceitação do produto/tecnologia/serviço comercializado pelo mercado	Facilidade na produção da tecnologia/produto inicialmente idealizado
Sintonia entre os fundadores	Bom relacionamento e entendimento entre fundadores e investidores
Capacidade de adaptação dos gestores às necessidades/mudanças do mercado	Disponibilidade de capital para investir no negócio

Tabela 2 - Fatores que contribuem para a sobrevivência de uma startup Fonte: Fundação Dom Cabral, 2014

## Startups descontinuadas

Fatores mais importantes para a descontinuidade	Fatores menos importantes para a
descontinuidade	descontinuidade
Falta de comprometimento do tempo integral dos fundadores exclusivamente para a startup	Não aceitação do produto/tecnologia/serviço comercializado pelo mercado
Não alinhamento dos interesses pessoais e ou profissionais dos fundadores	Falta de identificação pessoal dos fundadores com o negócio
Falta de capital para investir no negócio	Inviabilidade de produção da tecnologia produto inicialmente idealizado

Tabela 3 - Fatores que contribuem para a descontinuidade de uma startup Fonte: Fundação Dom Cabral, 2014

# 2.3 PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE EM STARTUPS

Fayad (2000) se referiu ao desenvolvimento de software em pequenas empresas ou startups como uma vertente de engenharia de software em pequena escala. Não no sentido de que elas fazem produtos de software menores ou diferentes das grandes empresas, mas se referindo a pequenos grupos que desenvolvem produtos de software normais. Também afirma que este é um indicativo de que estes pequenos grupos necessitam de métodos de software desenvolvidos para projetos de larga escala, mas que não precisam de todo o ferramental que oferecem, sendo a principal dificuldade enfrentada à aplicação destes métodos a estes pequenos grupos, ou seja, utilizando as ferramentas para resolver problemas de projetos grandes em projetos menores.

Fayad cita como alguns dos problemas que esses métodos solucionam e que não resultam no valor esperado para esta vertente de desenvolvimento como:

- O desenvolvimento voltado para a reutilização de código. O projeto de soluções reutilizáveis consome muito mais tempo e os escassos recursos para priorizar o prazo;
- A falta de histórico para as estimativas do projeto. As estimativas de custo e de tempo de desenvolvimento não são simples de aplicar em projetos de inovação com um novo time;
- Requisitos claros e estáveis praticamente não existem. Nesse caso a empresa precisa criar seus próprios requisitos, ao invés de se basear em uma especificação inicial de um cliente, também levando em consideração que a competitividade do mercado pode forçar a empresa a fazer grandes mudanças nos requisitos durante o desenvolvimento do produto. O que vai contra a ideia de que os requisitos devem ser minuciosamente especificados para que sua qualidade seja mantida;
- A utilização de muitas práticas recomendadas para a garantia da qualidade, como ter um time exclusivo para testes apartado do time de desenvolvimento, uso componentes de testes e gerentes de projetos para este fim. Muitos dos

elementos para gestão de projetos não são necessários como também podem ser prejudiciais para estes projetos em pequenas empresas;

Para o cliente, o uso de protótipos ou versões incompletas é mais importante, Fayad (2000), sendo uma forma de obter realimentação e melhorar continuamente o produto. Algo que se alinha com formas de desenvolvimento iterativo e incremental, onde pequenas partes da análise e desenvolvimento são feitas em curtos espaços de tempo sempre com algo entregue ao final de cada iteração (MARTIN, 1999).

Segundo Sutton (2000), processos altamente sistemáticos e direcionados por métricas são coerentes para sistemas de software crítico, mas para produtos inovadores, reativos e altamente dinâmicos, principalmente para sistemas baseados em internet, não traz os mesmos benefícios chegando a prejudicar com relação a competitividade de mercado que este tipo de *software* precisa atender devido a mudanças constantes.

Sutton considera que ter um processo definido também é uma tarefa importante para uma *startup*, principalmente por ela, ao longo de seu desenvolvimento, muitas vezes depender de parcerias com empresas específicas para alavancar seu crescimento e isso significa que deve haver uma boa comunicação sobre qual a forma de trabalho seguida, por exemplo, para o desenvolvimento de uma integração entre ambas.

Apesar de Paulk (1995) reconhecer a importância de um processo, propostas como o *Capability Maturity Model*, ou CMM como é conhecido, não podem ser utilizadas neste contexto, pois as empresas nesse estágio não têm experiência acumulada, histórico de projetos, lhes faltam recursos e sofrem de múltiplas influências do ambiente que estão inseridas.

Apesar de existirem novas versões do CMM, inclusive voltadas para pequenas empresas, permanece verdadeiro que as premissas do modelo não viabilizam a sua adoção para *startups*.

Ainda que startups possam ser consideradas pequenas empresas não é verdadeiro que toda pequena empresa seja uma *startup*, então existem pequenas empresas, não *startups*, que o modelo se aplica.

Fatores já apresentados, como a alta taxa de mudanças, rápida entrega de versões para testes, realimentação com a coleta de informações, recursos escassos e pouca informação sobre os reais requisitos, são alguns dos motivos para que modelos de desenvolvimento que defendam um planejamento completo e antecipado do projeto a ser desenvolvido antes de qualquer fase de implantação ocorra, inviabilizando sua utilização nos projetos de produtos inovadores e altamente mutáveis como o das startups.

Por atender exatamente a este contexto inovador, mutável e que necessita de transparência na comunicação entre os interessados no projeto, é que, segundo Highsmith e Cockburn (2001), os métodos ágeis para desenvolvimento de *software* são os que melhor se enquadram para o desenvolvimento e criação de projetos inovadores.

Highsmith e Cockburn (2001) ainda afirmam que empresas que lidam com este contexto podem vir a falhar seus projetos caso não estejam preparadas para mudanças.

Os processos ágeis valorizam a realimentação entre os envolvidos, valendo tanto para o time de desenvolvimento como para os clientes, a receptividade à mudança e ao desenvolvimento contínuo, de modo que a cada nova etapa o produto seja melhorado, adicionando novas funcionalidades e atendendo a novos requisitos com a responsabilidade compartilhada, onde todos estejam conscientes de sua importância e do seu papel no sucesso do projeto.

## 3 PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Este capítulo apresenta um histórico sobre os processos de desenvolvimento de software, a transição do modelo em cascata até o que hoje é conhecido como movimento ágil, apresentando o manifesto que declarou os valores que norteiam esse movimento, um estudo sobre alguns dos principais métodos ágeis utilizados atualmente e qual sua contribuição para o caso das empresas startup.

## 3.1 CONTEXTO HISTÓRICO E A CRISE DO SOFTWARE

Com o grande crescimento da indústria de computadores e *software*, entre o fim da década de 60 e início da década de 70, surgiram diversos problemas acompanhando esse crescimento. Estes problemas, relacionados aos projetos de software, se davam principalmente por:

- Estouro de orçamento e prazo do projeto;
- Software de baixa qualidade e que n\u00e3o atingiam seus requisitos;
- Projetos ingerenciáveis e de baixa manutenibilidade;

Este período ficou marcado pelo surgimento do termo "crise do *software*" (NAUR; RANDELL; 1969) que faz uma alusão aos problemas anteriormente mencionados.

Outro marco deste período foi o surgimento do Modelo em Cascata de desenvolvimento de software, conhecido também como *Waterfall Model* (ROYCE; 1970).

Este modelo propõe que haja um sequenciamento das etapas de desenvolvimento de software, havendo dependência entre cada etapa seguinte por sua etapa anterior. Neste modelo de processo considerado *design first*, onde o desenvolvimento será orientado pelo projeto, todos os requisitos do sistema e do software a ser produzido são documentados em diagramas UML, listas de requisitos funcionais, requisitos de qualidade, casos de teste e todo artefato que for necessário e então se avança para

a etapa seguinte de produção de código, teste e operação como pode ser visto na Figura 2.

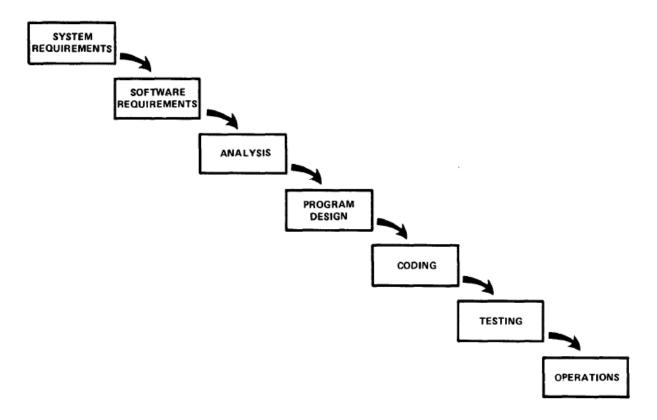


Figura 2 - Modelo sequencial proposto por Royce que deu origem ao Waterfall Fonte: ROYCE, 1970

No entanto, na visão do modelo de Royce que deu origem ao *Waterfall* era proposto que houvesse realimentação de informação entre as fases do modelo proposto, algo que os praticantes e gerentes ignoraram, transformando-o em um processo estritamente sequencial e sem troca de informações entre as fases (MARTIN; 1999), onde uma vez identificados todos os requisitos, seguido pelo projeto da solução, era esperado que tudo seguisse bem sem problema algum, já que tudo o que poderia acontecer relacionado ao *software* havia sido previsto durante a fase de projeto.

Esta visão onde tudo poderia ser previsto antes de iniciarem fases de produção do software pode ser considerada útil para sistemas críticos onde não há mudança nos requisitos do sistema, como em um sistema embarcado para aviões, onde após definidos seus requisitos não haverá alterações, mas para sistemas comerciais mostrou-se ineficaz, devido a mutabilidade nos requisitos por questões como a

concorrência de mercado, rapidez na entrega de uma versão para teste ou experimentação, necessidade especifica de grupos de usuários e etc (BROOKS; 1995).

A questão de mutabilidade nos requisitos se dá para Brooks (1995) como a única certeza que é possível ter durante o desenvolvimento de um sistema de *software*, devido aos fatores anteriormente já mencionados, o que indica que tentar prever completamente o futuro para um projeto leva ao erro e dá a falsa sensação de estar planejando corretamente o desenvolvimento do sistema.

Em 1995 o *The Standish Group* publicou um relatório que foi chamado de *Chaos Report*, ou Relatório do Caos, onde foram analisadas no ano de 1994 diversas empresas de *software* quanto ao seu sucesso nos projetos de *software* desenvolvidos e a análise dos dados obtidos chegou à seguinte conclusão, como pode ser vista no Gráfico 1:

- 16,2% de sucesso, onde os projetos foram finalizados no prazo correto e dentro do orçamento esperado com todas as funcionalidades inicialmente especificadas;
- 52,7% como desafio, o projeto foi finalizado com estouro de orçamento e com estouro no prazo estimado, oferecendo menos funcionalidades do que inicialmente especificado;
- 31,1% de fracasso, o projeto foi cancelado em algum ponto durante o seu ciclo de desenvolvimento;

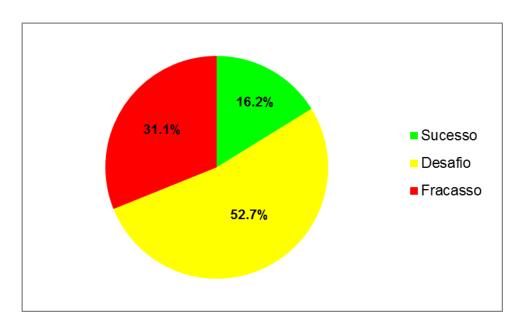


Gráfico 1- Taxa de Sucesso, Falha e Desafio em projetos de Software Fonte: THE STANDISH GROUP, 1995

Este relatório serviu como um alerta para que os processos para o desenvolvimento de *software* fossem repensados, principalmente quanto à utilização do modelo em cascata que era o processo mais utilizado principalmente nas empresas de grande porte.

A conclusão deste estudo identificou como um dos caminhos para o sucesso em projetos de *software* a utilização de abordagens que possibilitem o crescimento contínuo, baseadas em pequenos espaços de tempo que contemplem análise, desenho, implementação, teste e implantação, possibilitando que o *software* cresça através de iterações até que todas as funcionalidades necessárias sejam contempladas, abandonando a visão sequencial pregada pelo modelo em cascata praticado na maioria dos projetos da época.

Segundo Martin (1999), o modelo em cascata possuía resumidamente três grandes pontos de falha:

 Ausência de realimentação de informação entre as fases: ao concluir a análise segue-se para o projeto, ao concluir o projeto segue-se para a implementação sem permitir que haja mudança ou ajuste em algo relacionado

- a uma etapa anterior, trabalhando de forma estritamente sequencial, diferente do que foi proposto por Royce;
- Prazos equivocados para conclusão das fases: as datas para cumprimento dos prazos nas fases de análise e projeto são duas grandes metas, datas estimadas durante estas fases para conclusão das seguintes são utilizadas como indicadores para medir o trabalho dos desenvolvedores;
- Ausência de critério para finalização das etapas de análise e projeto: um grande equívoco desta abordagem é que não há como definir com precisão um estado de concluído para as fases de análise e projeto e que consequentemente criam prazos para as próximas etapas, ou seja, há estimativas fora da realidade, baseadas puramente na intuição do que fora produzido e que implica em uma altíssima probabilidade de falha e atraso por não levar a mudança em consideração ou a própria falta de informação;

Estes problemas levaram à utilização de novas abordagens que possibilitassem mudanças no percurso do projeto, rompendo o engessamento do processo anteriormente utilizado e levando a uma nova visão sobre como deve ser avaliado o bom andamento do projeto. Com isso passou-se a utilizar abordagens baseadas no modelo de processo iterativo e incremental, onde o desenvolvimento orienta-se por pequenas iterações representadas por espaços de tempo pré-determinados, como por exemplo, iterações de três semanas a um mês em que ao final de cada iteração será entregue alguma parte constituinte do *software* final esperado.

As etapas deste processo iterativo e incremental são vistas como a execução do modelo em cascata em um curto espaço de tempo, onde em uma iteração é possível identificar todas as etapas contidas no modelo em cascata, havendo análise, projeto, desenvolvimento e teste do entregável que foi negociado para aquela iteração (MARTIN; 1999).

## 3.2 PROCESSOS DE DESENVOLVIMENTO ÁGIL DE SOFTWARE

Em fevereiro de 2001, dezessete pessoas, dentre eles os criadores dos processos baseados no modelo iterativo e incremental mais utilizado e outros influentes praticantes da indústria de *software*, se reuniram para discutir a necessidade de uma alternativa aos processos burocráticos e orientados à documentação, ou processos de desenvolvimento de *software* pesados, como também são conhecidos (BECK ET AL., 2001).

Apesar de não chegarem a um consenso sobre o que, exatamente, esta proposta significava para os processos de desenvolvimento de *software*, eles definiram as quatro ideias centrais que devem ser valorizadas por esta nova corrente de pensamento:

- Indivíduos e interação entre eles mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Ainda que os autores do manifesto admitam que haja valor nos itens à direita, eles enfatizam que valorizam mais os itens à esquerda. Estas quatro ideias estão alinhadas com as críticas e questionamentos sobre as práticas tradicionais quanto à interação e relacionamento entre as pessoas, entrega de valor ao cliente e receptividade a mudança.

Estas ideias são melhores compreendidas através dos 12 princípios criados pelo manifesto e que devem ser seguidos:

- (1) Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado;
- (2) Mudanças nos requisitos são benvindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;

- (3) Entregar *software* funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos;
- (4) Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
- (5) Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;
- (6) O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;
- (7) Software funcionando é a medida primária de progresso;
- (8) Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
- (9) Contínua atenção à excelência técnica e bom design aumenta a agilidade;
- (10) Simplicidade a arte de maximizar a quantidade de trabalho não realizado é essencial;
- (11) As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis;
- (12) Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo. Estes doze princípios aprofundam os conceitos por de trás das quatro ideias centrais que definem o manifesto. Através destes doze princípios é possível identificar três características básicas que devem estar presentes nestes tipos de processos:

- Transparência contínua, fundamental no relacionamento entre os indivíduos, sejam eles gerentes, desenvolvedores ou clientes;
- Receptividade a mudança, encarar a mudança como uma característica benéfica diretamente ligada ao sucesso do projeto;
- Aumento na entrega de valor, foco no que é importante ao cliente, àquilo que de fato aumentará sua competitividade e consequentemente aumente seu sucesso;

Estas três características estão alinhadas com o conceito de agilidade, que apesar de possuir diversas definições que variam da particularidade de cada autor, pode ser compreendida através das definições feitas respectivamente por Highsmith (2002) como "a capacidade de criar e responder a mudanças a fim de lucrar em um ambiente de negócios turbulento" e Krutchen (2010) como "a habilidade de uma organização de reagir à mudança em seu ambiente mais rapidamente que a taxa de mudanças".

## 3.2.1 Adoção dos Processos Ágeis

Segundo dados de uma pesquisa realizada pela VersionOne (2014), empresa especializada na gestão do ciclo de vida de projetos ágeis, referentes ao levantamento feito no ano de 2013 com 3501 participantes de diferentes empresas, foi constatado que 88% dos participantes afirmaram que suas empresas utilizam processos de desenvolvimento ágil, mostrando um aumento sobre o resultado damesma pesquisa realizada em 2012 (84%) e daquela de 2011 (com 80%).

Como é possível observar no Gráfico 2, onde são agrupadas as taxas de adoção de processos ágeis dos últimos oito anos através dos relatórios da VersionOne, apesar de existirem algumas variações ao longo dos anos a taxa de adoção mantém-se maior ou igual a 80% nos últimos 8 anos, o que indica a importância destes processos na indústria de *software*.

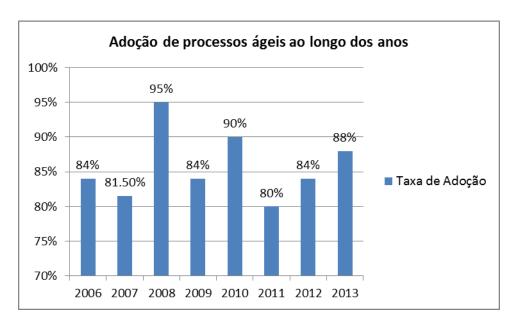


Gráfico 2 - Distribuição da adoção de processos ágeis ao longo dos anos Fonte: VersionOne, 2007-2014

Ao longo dos anos diferentes processos ágeis foram criados, inclusive antes da existência do manifesto e seus 12 princípios. Analisando os dados do relatório referente ao ano de 2013 criado pela VersionOne é possível identificar quais os processos ágeis mais utilizados.

A distribuição dos processos mais utilizados pode ser vista no Gráfico 3.

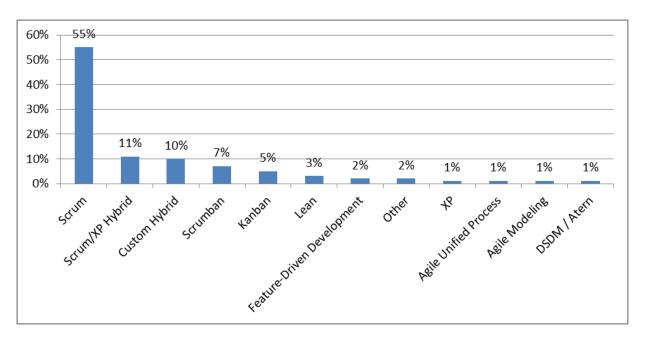


Gráfico 3 - Distribuição da adoção de processos ágeis Fonte: VersionOne, 2013

Ignorando-se os processos híbridos, os cinco processos ágeis mais utilizados, são:

- Scrum 55%
- Scrum/Xp Híbrido 11%
- Scrumban (Scrum + Kanban) 7%
- Kanban 5%
- Lean 3%

Os processos híbridos não foram considerados para esta classificação pela incapacidade de se definir quais os processos que serviram de base à sua constituição. De forma isolada tem-se Scrum, XP (*eXtreme Programming*), Kanban e Lean como processos ágeis mais utilizados em 2013, não sendo necessário levar em consideração suas versões combinadas como Scrum/XP Híbrido ou Scrumban. Através destes critérios é possível selecionar os processos ágeis utilizados para o escopo deste trabalho.

#### 3.2.2 Scrum

Desenvolvido de forma empírica e baseado em um processo incremental e iterativo, o scrum foi definido no livro "O Guia Scrum" em 2001 por seus criadores Ken Schwaber e Jeff Sutherland como um framework para pessoas lidarem com problemas complexos de forma criativa com foco na entrega de um produto com o maior retorno de valor possível.

O scrum é composto de eventos, papéis específicos desempenhados pelos envolvidos, regras e artefatos. Como papéis específicos tem-se:

- Product owner, ou dono do produto: pessoa indicada para representar os interesses dos stakeholders, definir prioridade no que será desenvolvido e tirar dúvidas do time quanto aos requisitos;
- Scrum máster: é a pessoa responsável por disseminar os valores do scrum e garantir que seus princípios e práticas sejam seguidos, assim como atuar como mentor, removendo impedimentos, e ajudando o time scrum e a empresa a obterem o melhor desempenho em seu processo;
- Time de desenvolvimento: basicamente o time que irá desenvolver o projeto, este time deve ser de no máximo 9 integrantes onde todos devem ser capazes de executar qualquer atividade, ou seja os integrantes do time devem ser pessoas multifuncionais;

#### Como suas atividades:

- Planejamento da sprint: reunião que antecede a execução da sprint onde o product owner irá apresentar os requisitos de maior prioridade para que sejam feitos durante a sprint, nesta etapa o time de desenvolvimento faz perguntas suficientes para que possam compreender o que deverá ser feito;
- Execução da sprint: execução de uma iteração de 2 semanas a um mês onde as atividades definidas para o projeto neste período serão executadas:
- Reunião diária: reunião de 15 minutos e em pé e que também são normalmente feitas no mesmo horário onde todos os integrantes do

time devem explicar em que estão trabalhando, o que fizeram anteriormente e se há algum impedimento no desenvolvimento do seu trabalho:

- Revisão da sprint: reunião ao final da sprint usada para apresentar ao product owner e aos stakeholders o que foi desenvolvido durante a sprint;
- Retrospectiva da sprint: reunião feita ao final da sprint apenas com os integrantes do time usada para a avaliação da última sprint executada com o intuído de identificar pontos de melhoria no processo e em sua execução;

#### E como seus artefatos:

- Product backlog: lista priorizada com uma breve descrição dos requisitos, esta priorização é feita pelo product owner junto aos stakeholders e é feita dos requisitos mais importantes para os menos importantes ou não tão bem definidos;
- Sprint backlog: lista de tarefas elencadas pelo time para serem executadas durante a sprint, estas tarefas são selecionadas pelo time durante o planejamento da sprint e são divididas em tarefas menores, sempre avaliando com uma estimativa o tempo necessário para completar cada tarefa;

O fluxo de desenvolvimento do scrum com seus papéis eventos e tarefas pode ser mais bem visualizado através da Figura 3.

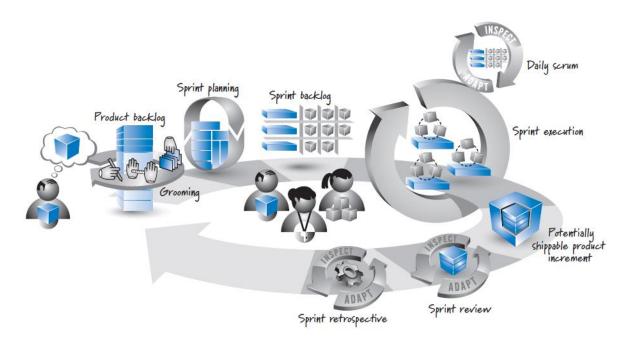


Figura 3 - Execução do processo scrum Fonte: RUBIN, 2012

## 3.2.3 Extreme Programming

O extreme programming é definido como uma disciplina leve para desenvolvimento de software com base em quatro valores principais, simplicidade, comunicação, feedback e coragem (BECK, 1996). Criado por Kent Beck, Ron Jeffries, e Ward Cunninghan o XP foca nos papéis de cliente, gerente e programador atribuindo direitos e responsabilidades especificas para cada um destes três papéis sendo:

- Cliente, aqueles que definem o que entregará valor ao negócio, que terá prioridade e o que poderá ser adiado e definindo quais testes serão necessários para que o sistema faça aquilo que é esperado;
- Programador, aqueles que analisam, projetam, testam, programam e integram o sistema. São responsáveis por estimar a dificuldade de todas as histórias e controlam o ritmo das histórias que são entregue aos clientes;
- Gerente, pessoa que une os clientes e os desenvolvedores, ajudando-os a atuarem em parceria como um time;

Considerados seus quatro mandamentos (JEFFRIES, 1999), seus valores podem ser definidos como:

- Simplicidade, que significa que o a solução desenvolvida será a mais simples possível, tanto em desenho como em sua construção eliminando ao máximo qualquer coisa desnecessária. Uma das regras do XP é conhecida como YAGNI do inglês "You aren't going to need it", ou "Você não irá precisar disso", que serve de alerta para que algo seja adicionado, ao código ou ao processo, somente quando for realmente necessário;
- Comunicação, considerado o fator chave para o desenvolvimento ágil e para a satisfação do cliente, tendo como principal forma de relacionamento a conversa face a face, ao invés de documentos, entre todos os envolvidos, sejam clientes, gerentes ou desenvolvedores, aumentando a confiança entre todos;
- Feedback, importante em qualquer processo de desenvolvimento e nevessário principalmente quando você tenta eliminar o máximo de coisas possível, o feedback contínuo é necessário para você saber se está no caminho certo;
- Coragem significa seguir o desenvolvimento utilizando a abordagem mais simples possível, sempre sendo honestos quanto as suas capacidades, sobre aquilo que você é capaz ou não de fazer;

O XP é composto de doze práticas que dão suporte a estes quatro mandamentos para que o time se mantenha no rumo certo, estas práticas podem ser vistas na Figura 4.

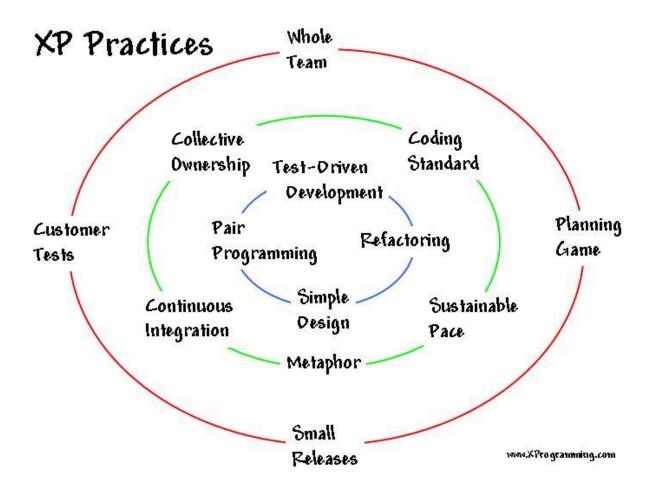


Figura 4 – Práticas da metodologia XP Fonte: JEFFRIES, 1999

Kent Beck (1996) define estas doze práticas como:

- Jogo do planejamento: estimar rapidamente o escopo da próxima entrega combinando prioridades de negócio com estimativas técnicas, sempre priorizando a comunicação entre os desenvolvedores e o cliente na definição e acordo dos prazos.
- Entregas pequenas: colocar um sistema simples rapidamente em produção e então entregar novas versões em pequenos ciclos (duas semanas);
- Metáfora: guiar todo o desenvolvimento com uma história simples compartilhada de como todo o sistema funciona;
- Design simples: O sistema deve ser projetado o mais simples possível em qualquer momento, toda complexidade desnecessária deve ser removida assim que descoberta;

- Teste: os programadores devem continuamente escrever testes unitários que devem ser executados de maneira transparente para o desenvolvimento continuar. Os clientes escrevem testes para demonstrar que as histórias foram concluídas.
- Refatoração: os programadores reestruturam o sistema sem mudar seu comportamento para remover duplicações, melhorar a comunicação, simplificar ou adicionar flexibilidade;
- Programação pareada: Todo o código produzido é escrito com dois programadores por máquina;
- Propriedade coletiva: qualquer um pode mudar qualquer trecho de código em qualquer lugar e a qualquer hora no sistema;
- Integração contínua: integrar novas mudanças no sistema em qualquer momento;
- 40 horas por semana: não trabalhar mais do que 40 horas por semana e caso isso ocorra nunca fazer horas extras por duas semanas seguidas;
- Cliente a bordo: inclua um usuário real no time, disponível a todo momento para responder possíveis dúvidas;
- Padronização de código: os programadores devem escrever todo o código de acordo com regras pré-determinadas para enfatizar a comunicação através do código;

As iterações no XP também ocorrem de 2 a 3 semanas, similar ao scrum, onde ao fim de cada iteração um conjunto de histórias de usuário são entregues. Histórias de usuários são requisitos escritas em linguagem comum, onde o cliente descreverá muito sucintamente a sua necessidade, informando o seu papel, o que precisa e qual benefício alcançará através desta necessidade.

Como processo que enfatiza o trabalho em equipe com relação à presença do usuário, programação pareada e propriedade coletiva o XP também preza pela qualidade através dos testes unitários, refatoração, design simples e integra contínua.

#### 3.2.4 Sistema Kanban

Desenvolvido pelo Vice Presidente da Toyota, Taiichi Ohno, como uma ferramenta para ser utilizado no Sistema Toyota de Produção desenvolvido pelo mesmo (SUGIMORI et al., 1977), o Sistema Kanban é uma ferramenta de apoio a produção de qualquer tipo de sistema, focando na gestão do trabalho efetuado e sendo muito útil em sua utilização junto aos métodos ágeis.

O termo Kanban, oriundo do japonês, significa cartão e representa exatamente sobre o que se trata este sistema: um sistema de cartões visualmente organizado conhecido como "sistema puxado" onde todo novo trabalho que adentre o sistema deverá ser puxado de uma esteira ou fila para sua realização, quando o sistema possuir capacidade para sua execução. Isso faz com que o sistema jamais seja sobrecarregado desde que a quantidade de cartões seja configurada de forma correta, o que significa que o inverso também é verdadeiro, na ausência de cartões não há execução de trabalho.

Sua versão aplicada ao desenvolvimento de *software* foi apresentada por David J. Anderson (ANDERSON, 2010) onde apresentou sua forma de trabalho conduzida através da utilização dos seus quatro princípios básicos:

- Comece com o processo existente: o Kanban não faz prescrição de papéis ou etapas de processo. Ele propõe uma mudança incremental e evolutiva no sistema iniciando com os processos e papéis existentes, sendo considerado um método para gestão de mudança;
- Buscar mudança incremental e evolucionária: todos, organização ou time, devem concordar que a mudança contínua, incremental e evolutiva é o caminho para que o sistema possa ser melhorado;
- Respeitar o processo, papéis, responsabilidades e títulos atuais: é comum haver coisas que são aceitáveis e valem a pena serem mantidas, a ideia é buscar direcionar o medo facilitando a mudança futura;
- Liderança em todos os níveis: promover o ato de liderança em todos os níveis, do individual ao coletivo, do junior ao sênior.

A utilização de painéis de fluxo de trabalho se tornou popular no desenvolvimento de software ágil devido a sua capacidade de limitar o trabalho-em-progresso para definir a capacidade e equilibrar a demanda em relação ao rendimento do trabalho entregue. Sua utilização também possibilita a visibilidade global do que está sendo feito e a identificação de problemas de qualidade e processos, tornando óbvio o impacto de defeitos, gargalos, variabilidade e custos no fluxo e vazão (ANDERSON, 2010).

Um painel Kanban em sua forma mais simples pode ser concebido através de três raias que dividem um fluxo de trabalho, a raia que contém o trabalho que deverá ser feito, a de trabalho em progresso e de trabalho concluído. Apesar desta ser uma estrutura simples, ela pode ser refinada para dar maior visibilidade as etapas do processo e especificar cada uma das etapas necessárias para a conclusão de um trabalho.

Um exemplo de painel Kanban pode ser visto na Figura 5, onde existem nove raias que representam um fluxo de desenvolvimento incluindo *buffers* para acumular o trabalho que antecede a etapa seguinte, a utilização de *buffers* é algo que varia da implementação dada e serve para dar visibilidade a possíveis gargalos, o que promove um de seus princípios que é a melhoria contínua.

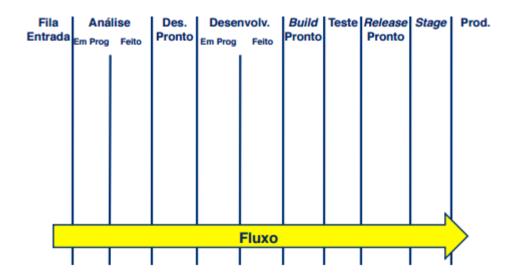


Figura 5 - Fluxo em um painel Kanban

Fonte: Anderson, 2010

#### 3.2.5 Desenvolvimento de Software Lean

O desenvolvimento de *software* Lean, ou apenas desenvolvimento Lean, é um método de desenvolvimento ágil adaptado do Sistema de Produção Toyota (SUGIMORI et al., 1977), para a produção de sistemas de *software*.

Este método foi apresentado por Mary Poppendieck e Tom Poppendieck (POPPENDIECK, Mary; et al., 2003) utilizando os princípios Lean tradicionais aplicados ao desenvolvimento de *software*, incluindo um conjunto de vinte e duas ferramentas que são comparadas as práticas ágeis e baseando-se nos sete princípios Lean.

Semelhante a produção de manufatura Lean o desenvolvimento Lean é resumido através de sete princípios:

 Eliminar desperdícios: desperdício é tudo aquilo que não adiciona valor ao produto, entendendo por valor como aquilo que é de fato percebido pelo cliente. Por exemplo, funcionalidades desenvolvidas que não serão utilizadas naquele momento;

- Ampliar o aprendizado: considerado um processo de descoberta, é esperado que o desenvolvimento de software explore opções, compreenda o domínio em que está envolvido e seja visto como um processo de aprendizado, ao invés de uma mera forma de produção;
- Decidir o mais tarde possível: tomar decisões definitivas que não permitam mudança em um ambiente volátil ou pouco compreendido irão causar problemas. Em ambientes de contínua mudança manter decisões de projeto em aberto até que se tenha informação suficiente para tomar a melhor decisão é melhor do que fazê-la precipitadamente;
- Entregar o mais rápido possível: uma das principais vantagens do desenvolvimento rápido é a capacidade de postergar decisões. Ser capaz de produzir algo com rapidez e qualidade possibilita que o cliente decida qual sua real necessidade no momento certo. A velocidade garante que o cliente receberá aquilo que ele precisa no exato momento e não o que precisava ontem. Esta abordagem é uma estratégia fundamental para a eliminação do desperdício;
- Emponderar a equipe: ninguém conhece melhor o trabalho do que as pessoas que o fazem. Envolver as pessoas com a expertise correta dentro do processo e guia-las com um líder faz com que melhores decisões técnicas e de processo sejam tomadas. Por conta da rápida entrega e da tomada de decisão tardia é impossível que decisões sejam centralizadas em uma única pessoa que orquestra as atividades desenvolvidas;
- Construir integridade: um software com integridade possui uma arquitetura coerente, alta pontuação em usabilidade, encaixa-se ao seu propósito, é manutenível, adaptável e extensível. A integridade vem de boa liderança, expertise relevante, comunicação efetiva e uma disciplina saudável;
- Visualizar o todo: especialistas tendem a otimizar o trabalho que envolve suas atividades, por exemplo, administradores de banco de dados focam na performance do tratamento das informações, programadores de interface de usuário em produzir telas com boa experiência para o usuário, programadores em ter um código limpo, manutenível e performático. Apesar de todas essas coisas serem boas para um sistema de software, elas são aplicadas separadamente, sem a visão do todo. Quando pessoas ou

organizações são medidas pela sua contribuição especializada, ao invés de sua contribuição para o desempenho global, o resultado mais comum é uma sub-otimização.

Cada um destes sete princípios (POPPENDIECK, Mary; et al., 2003) estão diretamente ligados as vinte e duas ferramentas propostas para que os princípios sejam praticados estas ferramentas e princípios podem ser visualizados no quadro abaixo.

Princípios Lean e sua 22 Ferramentas						
Eliminar	Identificar o	Identificar quais pontos de sua				
desperdícios	desperdício	produção podem ser fonte de				
		desperdícios				
	Reduzir o	Reduzir burocracias, tarefas de gestão				
	desperdício	desnecessárias, reduzir ineficiências				
		mapeando sua cadeia de valor				
Ampliar o	Realimentação de	Mantenha e aumente o ciclo de				
aprendizado	informação	realimentação da informação, execute				
		testes o mais cedo possível, reduza a				
		documentação				
	Iterações	Tenha um ciclo curto de				
		desenvolvimento				
	Sincronização	Solicite um sistema de gestão de				
		configuração, build diário e testes				
		automatizados.				
	Desenvolvimento em	Desenvolva múltiplas opções,				
	conjunto	comunique as regras e deixe a				
		solução emergir. Trata-se de				
		comunicar regras, não escolhas ou				
		soluções.				
Decidir o mais tarde	Opções de decisão	Tenha opções, não obrigações de				
possível		fazer algo no futuro, não é possível				
		prever o futuro, portanto, mantenha				

		flexibilidade até que não haja mais				
		incerteza				
	Decidir no último	Desenvolva um senso de quando as				
	momento	decisões precisam ser tomadas,				
		decida apenas quando tiver				
		informações suficientes.				
	Tomada de decisões	Decidir entre solucionar problemas				
		através de decisões em largura ou				
		profundidade				
Entregar o mais	Estimular o sistema	Permita que as pessoas decidam o				
rápido possível		que precisa ser feito, estimulando a				
		auto gestão.				
	Teoria de	Reduza o tempo do ciclo de				
	enfileiramento	desenvolvimento, ou o tempo que leva				
		o processo ponta a ponta.				
	Custo de atraso	Saiba qual o impacto do atraso em				
		sua cadeia de valor, o				
		desenvolvimento rápido economiza				
		tempo e dinheiro, determine quando.				
Emponderar a	Auto determinação	Deixe o time planejar sua própria				
equipe		forma de trabalho				
	Motivação	Crie um senso de propósito no				
		trabalho, pessoas se importam com				
		propósito mais do que consigo				
		mesmas.				
	Liderança	Líderes lidam com a mudança,				
		definem o rumo, alinham pessoas e				
		possibilitam a motivação.				
	Perícia	Promova mentoria e atividades de				
		programação pareada, encoraje				
		treinamento e aprendizado continuo				
Construir	Integridade	Sistemas pequenos são				
integridade	percebida	desenvolvidos por um único time com				

		acesso direto as pessoas que julgarão
		a integridade do sistema, pequenas
		iterações devem ser usadas para
		promover ciclos de realimentação.
	Integridade	A efetividade da comunicação das
	conceitual	decisões é crítico, remover a
		complexidade diante do projeto como
		discussões sobre tecnologia em
		momento não apropriado.
	Refatoração	Sistemas complexos possuem efeitos
		que não são compreendidos durante o
		projeto, a arquitetura deve manter-se
		saudável durante a evolução do
		sistema e manter a integridade
		conceitual.
	Teste	Comunicar como as coisas devem
		funcionar, promover resposta em
		tempo real se o sistema funciona de
		acordo com seu projeto, testes de
		desenvolvimento e testes de cliente.
Visualizar o todo	Medições	Medições devem encorajar otimização
		do todo e o time deve colaborar para
		melhorar sua forma de trabalho .
	Contratos	Procure trabalhar com contratos com
		escopo opcional e peça ajuda ao seu
		time jurídico para encontrar a melhor
		forma de trabalhar isso e proteger
		todas as partes. Uma solução é utilizar
		contratos por estágios e de benefício
		compartilhado podem ajudar a
		organizar acordos entre empresas que
		utilizam desenvolvimento ágil.

#### Quadro 1 - Princípios Lean

## 3.3 Software Engineering Method and Theory (SEMAT)

Devido à grande diversidade de métodos existentes para o desenvolvimento de software e seu aumento contínuo, em 2009, Ivar Jacobson, Bertrand Meyer e Richard Soley iniciaram uma chamada a comunidade de software com o objetivo de buscar uma teoria para esta área, de modo que se torne uma disciplina com maior rigor e sólida base teórica.

As consequências desta grande quantidade de métodos sem uma sólida fundamentação são a prevalência de modismos mais típicos da indústria da moda que de uma disciplina de engenharia, a falta de uma base teórica sólida e largamente aceita, uma grande quantidade de métodos e variantes de métodos com pequenas diferenças compreendidas e artificialmente aumentadas, a falta de avaliação e validação experimental de credibilidade e a separação entre a pesquisa acadêmica e a prática da indústria (JACOBSON; MEYER; SOLEY, 2009).

A iniciativa pretendida deveria atender aos seguintes critérios:

- Definir um kernel de elementos amplamente aceitos, definidos, extensível para usos específicos;
- Abordar tanto questões de tecnologia quanto de pessoas;
- Ser suportada pela indústria, academia, pesquisadores e usuários; e
- Ter suporte a extensões em face da mudança dos requisitos e da tecnologia.

O resultado da inciativa SEMAT foi chamado de Essência e define um *kernel* escalável e extensível. O *kernel* fornece a base de elementos e vocabulário comuns para ajudar praticantes a compararem métodos e fazerem melhores decisões sobre suas práticas.

A Figura 5 mostra a arquitetura do método SEMAT, ela é composta por 4 componentes básicos:

- Método: uma composição de práticas dinâmicas e usadas para dar suporte as atividades diárias dos desenvolvedores;
- Prática: abordagem repetível de se fazer algo com um propósito específico em mente, uma prática proporciona uma forma sistemática e verificável de se alcançar algum objetivo específico do trabalho;
- Kernel: incluí os elementos essenciais da engenharia de software; e
- Linguagem: vocabulário de um domínio específico para definir métodos, práticas e os elementos essenciais do kernel.

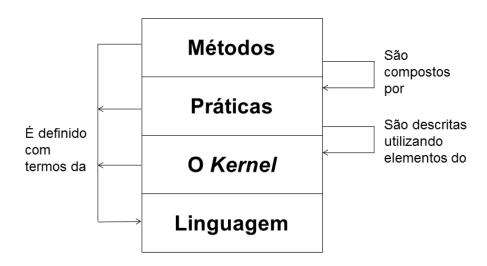


Figura 6 – Arquitetura do método SEMAT Fonte: SEMAT, 2012

#### 3.3.1 Organização do Kernel

O *kernel* proposto pelos autores é organizado em três domínios específicos, cada um focando em um domínio específico do desenvolvimento de *software*, (Figura 6) sendo estes:

 Cliente: todo desenvolvimento de software envolve pelo menos um cliente que demanda uma necessidade específica. A perspectiva do cliente deve ser considerada no dia-a-dia de trabalho para garantir que a solução correta seja desenvolvida, este domínio está relacionado ao uso e aquisição de valor do sistema de software desenvolvido;

- Solução: o objetivo principal do desenvolvimento de software é propor a solução de um problema através de sistema de software que funcione. O domínio de solução possui todos os aspectos relacionados a especificação e desenvolvimento de um sistema de software;
- Empreitada: o desenvolvimento de software trata-se de uma empreitada que leva tempo e trabalho para atingir a seus objetivos ela envolve um time de desenvolvimento e afeta diferentes pessoas. O domínio de empreitada está relacionado aos aspectos que envolvem o time de desenvolvimento e a forma com que efetuam seu trabalho (JACOBSON; et al, 2012).



Figura 7 - Dimensões do SEMAT Kernel Fonte: SEMAT, 2012

O *kernel* tem por função capturar a essência da engenharia de *software*, formar um mapa do contexto relacionado a engenharia de *software* e constituir uma base para a avaliação do trabalho em execução.

Cada um dos três domínios do SEMAT *Kernel* possue um conjunto de conceitos universais a toda empreitada de software cuja evolução deve ser compreendida, monitorada e controlada, estes conceitos são chamados de *alphas*, um acrônimo para *Abstract-Level Progress Health Attribute*, que traduzido para o português significa Atributo de Saúde de Progresso de Nível Abstrato.

### 3.3.2 Os Alphas

Os *alphas* representam elementos que devem ser monitorados através de seu progresso e saúde para que sua empreitada seja finalizada com sucesso, possuindo listas de checagem para que isso seja alcançado. Ao todo o *kernel* possui sete *alphas* básicos que são os itens minimamente necessários para uma empreitada de *software*, sendo estes, por domínio:

- Cliente: oportunidade e stakeholders;
- Solução: requisitos e sistema de software;
- Empreitada: trabalho, equipe e forma de trabalho.

Individualmente cada *alpha* possui uma definição específica que define qual o seu papel na estrutura do *kernel* a qual pertence (JACOBSON; et al, 2012):

- Oportunidade: o conjunto de circunstâncias que motivam o desenvolvimento ou uma mudança em um sistema de software;
- Stakeholders: as pessoas, grupos ou organizações que afetam ou são afetadas por um sistema de software;
- Requisitos: o que um sistema de software deve fazer para tratar a oportunidade e satisfazer os stakeholders;
- Sistema de Software: sistema constituído de software, hardware e dados que provem seu valor primário através da execução de software;
- Time: o grupo de pessoas ativamente engajadas no desenvolvimento, manutenção, desenvolvimento e suporte de um sistema de software específico;
- Trabalho: atividade envolvendo esforço físico ou mental feita para alcançar um objetivo;
- Forma de Trabalho: o conjunto sob medida de práticas e ferramentas usadas pelo time para guiar e suportar seu trabalho.

Os *alphas* não são elementos individuais, eles relacionam-se uns com os outros como pode ser visto na Figura 7, formando um mapeamento global de relacionamentos entre os elementos de cada domínio.

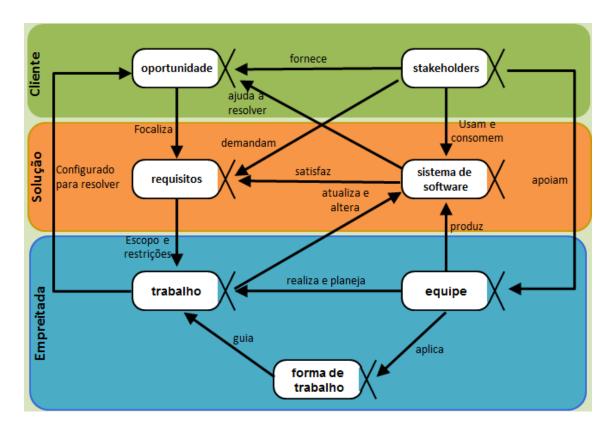


Figura 8 – Alphas que constituem a Essência Fonte: SEMAT, 2012

Os alphas são elementos que possuem estados definidos que atuam na forma de listas de checagem que serão acompanhadas e evoluídas durante toda a empreitada. Cada alpha é controlado através de cartões manipuláveis, Figura 8, que são movimentados da esquerda para direita, por exemplo, no início de uma empreitada todos os seis estados do alpha requisitos iniciam-se a esquerda e conforme cada um de seus estados é alcançado, de acordo com o que o respectivo estado exige, ele avança para a direita. Desta forma, é possível medir a evolução, analisando em que ponto encontram-se os alphas através de uma visão global do progresso da empreitada.

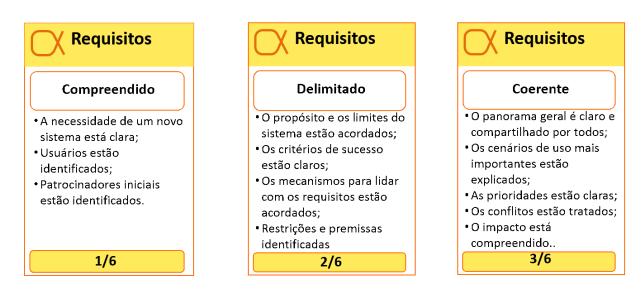


Figura 9 - Estados do alpha Requisitos Fonte: SEMAT, 2012

Apesar de existirem sete *alphas* como elementos essenciais a uma empreitada de *software*, é possível a criação de sub *alphas* para apoiar uma forma específica de trabalho, criando assim novas listas de checagem de acordo com um contexto específico.

# 4 SELEÇÃO DE PRÁTICAS ÁGEIS E MAPEAMENTO ATRAVÉS DA ESSÊNCIA

Neste capítulo serão apresentadas e descritas as práticas contidas nos métodos ágeis elegidos, assim como o seu mapeamento dentro do SEMAT *Kernel*.

## 4.1 TAXONOMIA DE PRÁTICAS ÁGEIS

De acordo com o levantamento feito pela Version One, foi possível analisar a variação da adoção das principais práticas ágeis nos últimos anos. Para este trabalho são considerados apenas os dois últimos relatórios. A variação pode ser analisada na Tabela 4.

Prática	2013	2012	Em relação ao ano anterior
Daily Standup	85%	85%	0%
Iteration Planning	75%	75%	0%
Unit Testing	72%	74%	-2%
Retrospectives	74%	72%	+2%
Release Planning	70%	69%	+1%
Burndown/Team-Based Estimation	69%	67%	+2%
Velocity	60%	58%	+2%
Coding Standards	55%	57%	-2%
Continuous Integration	58%	56%	+2%
Automated Builds	56%	55%	+1%
Dedicated Product Owner	55%	51%	+4%
Integrated Dev/QA	50%	49%	+1%
Refactoring	47%	48%	-1%
Open Workarea	44%	43%	+1%
TDD	38%	40%	-2%
Digital Taskboard	45%	39%	+6%
Story Mapping	41%	38%	+3%
Kanban	39%	32%	+7%
Collective Code Ownership	29%	32%	-3%
Pair Programming	30%	30%	0%
Automated Acceptance Testing	28%	27%	+1%
Analog Taskboard	22%	24%	-2%
Continuous Deployment	25%	23%	+2%
Agile Games	17%	17%	0%
Cycle Time	15%	13%	+2%
BDD	12%	10%	+2%

Tabela 4 - Adoção de Práticas Ágeis Fonte: VersionOne, 2012-2013

Muitas das práticas apresentadas estão simultaneamente presentes em outros métodos como é possível observar no mapa das práticas presentes nos principais métodos ágeis, Figura 10. Outro aspecto importante a ser citado é que as práticas mais utilizadas assim como as com maior crescimento são em sua maioria práticas presentes nos quatro métodos elegidos para a realização desta pesquisa.

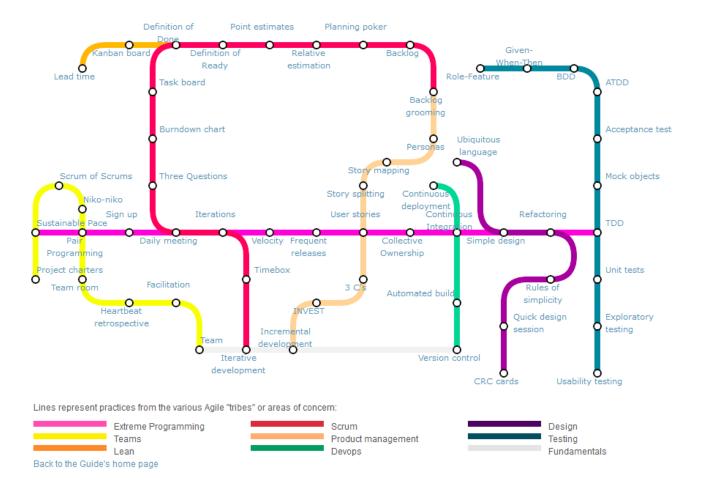


Figura 10 - Mapa de Práticas Ágeis Fonte: Agile Alliance, 2015 [site]

Lago (2014) descreveu 59 práticas presentes nos métodos ágeis, dentre elas estão algumas das práticas pertencentes aos métodos eleitos para o escopo deste trabalho. Foram utilizadas partes das descrições feitas por Lago (2014) para algumas das práticas dos métodos eleitos, o apoio da literatura e a experiência do autor. Também foram consideradas como práticas as 22 ferramentas do método Lean, já apresentadas. Suas descrições podem ser analisadas na Tabela 5.

Prática	Descrição				
Ritmo sustentável	também conhecido como semana de 40 horas, procurar fazer com que o trabalho seja realizado sem horas extras com qualidade em um ritmo sustentável (BARBOSA; LACERDA, [s.d.]);				
Programação pareada	é uma técnica na qual duas pessoas se sentam literalmente lado-a- lado e escrevem um programa no mesmo computador (WRAY, 2010);				
Seleção voluntária de tarefas	os próprios desenvolvedores selecionam as tarefas que realizarão (COHEN; LINDVALL; COSTA, 2004);				
Velocity	ao final de cada iteração o time calcula o total de estimativas das histórias de usuário completadas. Com isso realiza-se a estimativa de quantas iterações serão necessárias para completar o projeto (ALLEMAN, 2003);				
Integração contínua	é o processo automatizado – normalmente realizado diariamente – de unificar os esforços de atores independentes para entregar um sistema completo (MCCONNEL, 1996);				
Histórias de usuários	são descrições curtas e abstratas do que deve ser implementado que servem primariamente como âncoras para maiores discussões com os clientes. Os desenvolvedores discutem os requisites em detalhe com o cliente durante o desenvolvimento (MAURER; MARTEL, 2002);				
Propriedade coletiva	a equipe possui propriedade coletiva do software desenvolvido. Qualquer pessoa pode alterar qualquer parte do código. Porém códigos desenvolvidos só podem ser integrados se todos os testes unitários forem atendidos (MACKENZIE; MONK, 2004);				
Implantação contínua	é uma extensão da integração contínua que consiste em automatizar o processo de implantação do software (HUMBLE; READ; NORTH, 2006);				
Design simples	define que todo o código desenvolvido deve fazer parte de alguma das tarefas do backlog, em outras palavras – não programe hoje o que será necessário amanhã (FOWLER, 2001);				
Refatoração	nova escrita do código sem adição de funcionalidades (Wood, 2013);				
Desenvolvimento orientado à testes	é considerado a união de duas práticas 'test-first development' que consiste em escrever os testes de unidade antes que o código necessário para cumprir tais testes seja escrito e refatoração (Rafique, 2013);				

Padrões de Codificação	Padrões de codificação permitem implementações mais fáceis para todos os programadores, evitando que sejam perpetuados diferentes estilos de programação pelo código (PAULK, 2001);			
Visão do trabalho, fluxo de trabalho e riscos	visualização de todo o fluxo de trabalho de modo que seja possívelidentificar gargalos e a situação de cada um de seus estágios, a manifestação clara dessa prática é o quadro Kanban (SUGIMORI et al., 1977);			
Limitação do trabalho em progresso	limitar o trabalho em progresso gera foco na execução de tarefas e não excede a capacidade de execução de um time, participantes em cada etapa executam suas atividades somente no momento em que uma atividade de etapa anterior é finalizada (SUGIMORI et al., 1977);			
Gestão de fluxo	conhecido como gerir e mensurar o fluxo, essa prática recomenda que haja medições em seu fluxo de trabalho, no Kanban medir basicamente significa contar coisas (SUGIMORI et al., 1977);			
Tornar o processo explícito	manter as regras e a forma de trabalho de um fluxo da forma mais clara possível (SUGIMORI et al., 1977);			
Implementar ciclos de feedback	promover melhoria contínua através da avaliação do fluxo e do trabalho executado de forma explicita (ANDERSON, 2010);			
Aumentar colaborativamente e evoluir experimentalmente	experimente diferentes modelos capazes de promover melhorias de forma colaborativa (ANDERSON, 2010);			
Desenvolvimento incremental	o projeto é dividido em fatias 'verticais' que adicionam a cada versão funcionalidades visíveis para o usuário (MARTIN, 1999);			
Desenvolvimento iterativo	o desenvolvimento é incremental no sentido que atividades podem ser revisitadas em diversos momentos, e é organizado em iterações com objetivos determinados (MARTIN, 1999);			
Timebox	a equipe possui uma quantidade de tempo fixa na qual o sistema deve ser construído, ela possui como entrada uma lista de funções do sistema e como saída um sistema que será avaliado para decidir se está pronto para ser colocado em produção (ABBAS; GRAVELL; WILLS, 2008);			
Iterações	no contexto ágil é utilizado para definir uma <i>timebox</i> , usualmente com duração entre 1 e 4 semanas e com duração fixa durante todo o projeto (ABBAS; GRAVELL; WILLS, 2008);			
Reunião diária	essa práticas também conhecida como <i>daily scrum</i> é definida como uma reunião curta (cerca de 15 minutos) realizada todos os dias para atualização do andamento do projeto (PAASIVAARA; DURASIEWICZ; LASSENIUS, 2009);			
Três perguntas	na reunião diária, todos os desenvolvedores têm que responder a três perguntas: "O que você realizou ontem?", "O que você fará hoje?" e "O que está te atrapalhando?" (SUTHERLAND; VIKTOROV, 2007);			
Burndow chart	diagrama que indica a quantidade de trabalho que resta a ser executada pela equipe (MIRANDA; BOURQUE, 2010);			
Taskboard Digital	sistema de software onde um grupo pode manter a situação de itens do projeto e atualizá-los (PERRY, 2008);			
Taskboard analógica	local físico onde um grupo mantém a situação atual de itens do projeto (MORAN et al., 1999);			

Definição de pronto	análoga a definição de feito, porém os critérios indicam se uma tarefa pode ser considerada pronta para ser desenvolvida evitando muita discussão e retrabalho (JAKOBSEN; SUTHERLAND, 2009);			
Definição de feito	a equipe define uma lista de critérios que uma tarefa deve atingir para ser considerada executada e os utiliza consistentemente. Esse conceito apareceu pela primeira vez em um exercício de coaching para praticantes da Programação extrema (JAKOBSEN; SUTHERLAND, 2009);			
Estimativa por pontos	estimativas sem relação com unidades reais de tempo utilizadas para evitar confusão entre estimativa e tempo de trabalho (LECHNER, 2008);			
Estimativa relativa	estimativa de histórias de usuário (ou casos de uso) realizadas não em unidades absolutas de tempo mas em comparação com outras histórias de usuário (MIRANDA, 2001);			
Planning Poker	escala de pontuação baseada na escala de fibonacci para medir a complexidade de uma tarefa executada (HAUGEN, 2006);			
Backlog	Backlog do produto (ou da Sprint no SCRUM) é uma lista ordenada de tudo que deve ser desenvolvido, e é uma origem única de todas as mudanças que devem ser feitas no sistema(SCHWABER; SUTHERLAND, 2009);			
Tratamento do <i>backlog</i> ( <i>backlog grooming</i> )	é o ato de adicionar detalhes, estimativas e ordenar os itens no backlog (DULLEMOND; VAN GAMEREN; VAN SOLINGEN, 2012);			
Lead Time	tempo entre a descoberta de um requisito e sua implementação, essa medida é usada de forma equivalente ao <i>velocity</i> por equipes que utilizam Kanban (HÖST; REGNELL; WOHLIN, 2000);			
Kanban Board	semelhante aos taskboards tradicionais, porém com a visão de projeto baseada nos princípios do Kanban com o objetivo de visualizar o todo e identificar gargalos propondo melhorias continuas ao processo (SUGIMOR et al., 1977);			
Identificar o desperdício	Identificar quais pontos de sua produção podem ser fonte de desperdícios (POPPENDIECK, Mary; et al., 2003);			
Reduzir o desperdício	Reduzir burocracias, tarefas de gestão desnecessárias, reduzir ineficiências mapeando sua cadeia de valor (POPPENDIECK, Mary; et al., 2003);			
Realimentação de informação	Mantenha e aumente o ciclo de realimentação da informação, execute testes o mais cedo possível, reduza a documentação (POPPENDIECK, Mary; et al., 2003);			
Iterações	Tenha um ciclo curto de desenvolvimento (POPPENDIECK, Mary; et al., 2003);			
Sincronização	Solicite um sistema de gestão de configuração, build diário e testes automatizados (POPPENDIECK, Mary; et al., 2003);			
Desenvolvimento em conjunto	Trata-se de comunicar regras, não escolhas ou soluções. Desenvolva multiplas opções, comunique as regras e deixe a solução emergir (POPPENDIECK, Mary; et al., 2003);			
Opções de decisão	Tenha opções, não obrigações de fazer algo no futuro, não é possível prever o futuro, portanto, mantenha flexibilidade até que não haja mais incerteza (POPPENDIECK, Mary; et al., 2003);			
Decidir no último momento	Desenvolva um senso de quando as decisões precisam ser tomadas, decida apenas quando tiver informações suficientes (POPPENDIECK, Mary; et al., 2003);			

Tomada de decisões	Decidir entre solucionar problemas através de decisões em largura ou profundidade (POPPENDIECK, Mary; et al., 2003);
Estimular o sistema	Permita que as pessoas decidam o que precisa ser feito, estimulando a auto gestão (POPPENDIECK, Mary; et al., 2003);
Teoria de enfileiramento	Reduza o tempo do ciclo de desenvolvimento, ou o tempo que leva o processo ponta a ponta (POPPENDIECK, Mary; et al., 2003);
Custo de atraso	Saiba qual o impacto do atraso em sua cadeia de valor, o desenvolvimento rápido economiza tempo e dinheiro, determine quando (POPPENDIECK, Mary; et al., 2003);
Auto determinação	Deixe o time planejar sua própria forma de trabalho
Motivação	Crie um senso de propósito no trabalho, pessoas se importam com propósito mais do que consigo mesmas (POPPENDIECK, Mary; et al., 2003);
Liderança	Líderes lidam com a mudança, definem o rumo, alinham pessoas e possibilitam a motivação (POPPENDIECK, Mary; et al., 2003);
Perícia	Promova mentoria e atividades de programação pareada, encorage treinamento e aprendizado continuo (POPPENDIECK, Mary; et al., 2003);
Integridade percebida	Sistemas pequenos são desenvolvidos por um único time com acesso direto as pessoas que julgarão a integridade do sistema, pequenas iterações devem ser usadas para promover ciclos de realimentação (POPPENDIECK, Mary; et al., 2003);
Integridade conceitual	A efetividade da comunicação das decisões é crítico, remover a complexidade diante do projeto como discussões sobre tecnologia em momento não apropriado (POPPENDIECK, Mary; et al., 2003);
Refatoração	Sistemas complexos possuem efeitos que não são compreendidos durante o projeto, a arquitetura deve manter-se saudável durante a evolução do sistema e manter a integridade conceitual (POPPENDIECK, Mary; et al., 2003);
Teste	Comunicar como as coisas devem funcionar, promover resposta em tempo real se o sistema funciona de acordo com seu projeto, testes de desenvolvimento e testes de cliente (POPPENDIECK, Mary; et al., 2003);
Medições	Medições devem encorajar otimização do todo e o time deve colaborar para melhorar sua forma de trabalho (POPPENDIECK, Mary; et al., 2003);
Contratos	Procure trabalhar com contratos com escopo opcional e peça ajuda ao seu time jurídico para encontrar a melhor forma de trabalhar isso e proteger todas as partes. Uma solução é utilizar contratos por estágios e de benefício compartilhado podem ajudar a organizar acordos entre empresas que utilizam desenvolvimento ágil. (POPPENDIECK, Mary; et al., 2003);

Tabela 5 - Descrição de práticas contidas nos métodos eleitos

#### 4.2 MAPEAMENTO DE PRÁTICAS NO SEMAT KERNEL

Cada uma das 58 práticas apresentadas no item anterior serão mapeadas entre os três domínios da essência do SEMAT Kernel já apresentados na Figura 7. De acordo com a experiência do autor deste trabalho e o papel desempenhado pelas práticas em relação a cada domínio, na Figura 8 é possível a identificação dos domínios ao qual as respectivas práticas pertencem e o mapeamento feito poder ser visto no Quadro 2.

Cliente						
Solução	Integração contínua, Design simples, Refatoração					
	Desenvolvimento orientado à testes, Padrões de					
	Codificação, Lead Time, Desenvolvimento em conjunto.					
Empreitada	Ritmo sustentável, Programação pareada, Seleção					
	voluntária de tarefas, Velocity, Visão do trabalho - fluxo					
	de trabalho e riscos, Gestão de fluxo, Tornar o processo					
	explícito, Implementar ciclos de feedback, Aumentar					
	colaborativamente e evoluir experimentalmente, Reunião					
	diária, Três perguntas, Taskboard Digital, Taskboard					
	analógica, Estimativa por pontos, Estimativa relativa,					
	Planning Poker, Kanban Board, Identificar o desperdício,					
	Estimular o sistema, Auto determinação, Motivação,					
	Perícia, Medições.					
Cliente e Solução	Tratamento do backlog (backlog grooming),					
	Realimentação de informação, Opções de decisão,					
	Decidir no último momento, Contratos.					
Cliente e Empreitada	Implantação contínua, Tomada de decisões.					
Solução e Empreitada	Propriedade coletiva, Limitação do trabalho em					
	progresso, Reduzir o desperdício, Sincronização, Teoria					
	de enfileiramento, Refatoração.					
Cliente, Solução e	Histórias de usuários, Desenvolvimento incremental,					
Empreitada	Desenvolvimento iterativo, Timebox, Iterações,					

Burndown chart, Definição de pronto, Definição de feito,						
Backlog,	Iterações,	Custo	de	atraso,	Liderança,	
Integridade percebida, Integridade conceitual, Teste.						

Quadro 2 - Mapeamento de práticas no SEMAT Kernel

Entre a distribuição das 58 práticas levantadas, aproximadamente 40% estão unicamente relacionadas ao domínio de empreitada e 10% a solução e empreitada simultaneamente, indicando que em sua maioria tais práticas lidam com aspectos relacionados principalmente a forma de trabalho, seu processo e execução.

Aproximadamente 26% das práticas estão relacionadas aos três domínios em conjunto e apesar de não haverem práticas exclusivamente relacionadas ao domínio de cliente, todos os domínios possuem práticas que lidam com os estados descritos nas listas de checagem contidas em cada alpha.

#### 4.3 MAPEAMENTO DE PRÁTICAS ATRAVÉS DA ESSÊNCIA

Neste item o número de práticas ágeis será reduzido a um conjunto composto pelas práticas mais aplicáveis no ambiente de uma *startup* e que proporcionem a maior cobertura possível aos *alphas* presentes na essência para que possam ser analisadas em uma validação de sua aderência a processos de desenvolvimento em startups.

## 4.3.1 SEMAT COMO UMA FERRAMENTA DE CONTROLE PARA PROJETOS DE SOFTWARE

A priori o SEMAT *kernel* pode parecer algo abstrato, apesar de sua capacidade de contribuição para uma condução mais saudável em empreitadas de *software*. No entanto, o *kernel* foi desenvolvido principalmente para ser utilizado em conjunto com os métodos de desenvolvimento já existentes e também servir de ferramental para a criação de novos métodos através da extensão de seus *alphas* e da composição de práticas presentes em diferentes métodos (OMG, 2014).

Na Figura 12 é possível visualizar a forma como o SEMAT *kernel* pode interagir com os diferentes métodos e práticas existentes. Sua visão é de um meta-modelo de nível mais abstrato capaz de mapear e compor todas as práticas existentes.

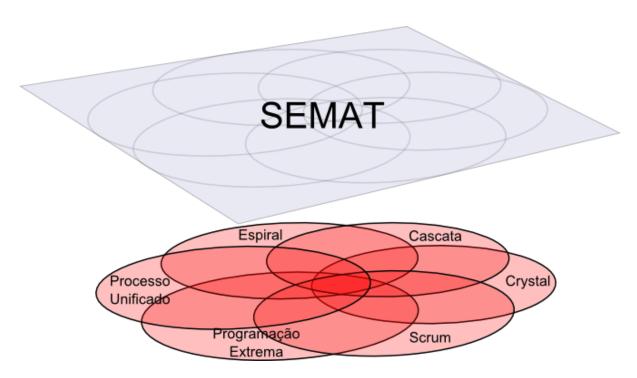


Figura 11 - SEMAT como ferramenta de controle Fonte: Patrício, 2013

## 4.3.2 COMPOSIÇÃO DE PRÁTICAS COM SEMAT KERNEL

De acordo com a proposta SEMAT, diferentes práticas de desenvolvimento enfrentam diferentes desafios e riscos o que implica na não existência de um método definitivo que seja possível atender a todas as facetas de uma empreitada de *software* (JACOBSON, I, 2013).

Um método nada mais é do que a composição de práticas e uma prática provê a diretiva para lidar com alguma dimensão do desenvolvimento de *software*, portanto um método é um conjunto de diretivas que lidam com diferentes dimensões do desenvolvimento de *software* de acordo com a sua finalidade.

A composição de práticas no SEMAT *kernel* lida com dois principais aspectos:

- "Coisas com o que trabalhamos": são os alphas que representam os elementos que sempre estarão presentes em uma empreitada de software e o produto de um trabalho que representa uma evidência gerada pelo progresso de um alpha específico através de um documento físico, relatório ou algo que mostre o indício de seu progresso;
- "Coisas a fazer": abrangem os espaços de atividades, Figura 12, que representam as atividades independentes de métodos que em algum momento serão adicionadas no topo do *kernel* para atuarem como diretivas para alcançarem os estados dos *alphas*;

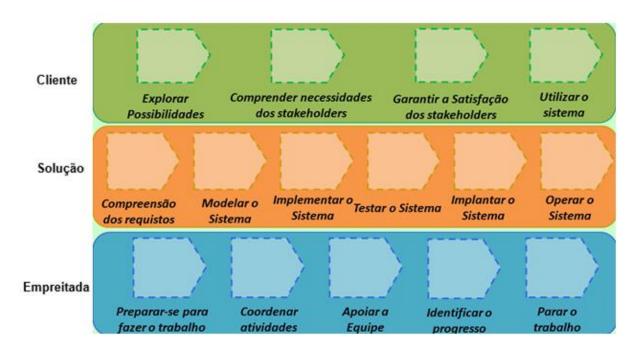


Figura 12 - Espaços de Atividades dos três domínios (Coisas que sempre fazemos)

Fonte: SEMAT, 2013

A relação entre produto de trabalho e os *alphas*, e o espaço de atividade e a atividade podem ser visualizados através do diagrama apresentado na Figura 13. O produto de trabalho atua como um elemento descritivo que indica a evolução ou

estado específico de um alpha. A atividade é vista como um refinamento de um espaço de atividade específico onde é necessária uma atuação descritiva de como conduzir aquele espaço de atividade de acordo com um contexto específico.

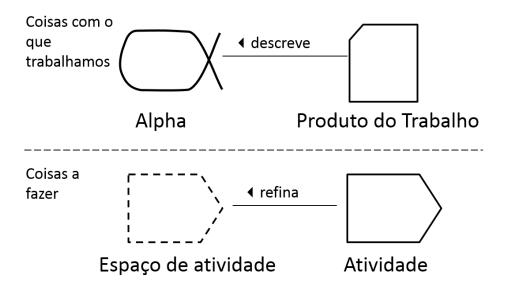


Figura 13 - Mapeamento de produto de trabalho e atividade Fonte: SEMAT, 2013

Para que seja possível compreender como práticas de métodos distintos atuam em conjunto é necessário descrevê-las em uma linguagem em comum que possibilite sua composição, sendo esta uma das principais utilidades proporcionadas pela linguagem contida na essência.

# 4.3.3 ELEIÇÃO DE PRÁTICAS PARA STARTUPS

Feitas as seleções dos métodos e identificadas suas respectivas práticas, é necessária a eleição de um subconjunto de práticas ágeis que respeitem o ambiente de uma *startup* e que possa cobrir os *alphas* de modo que seja garantido o atendimento dos aspectos contidos na essência de uma empreitada de *software*.

Os critérios na escolha destas práticas são:

Consumo de recursos, financeiros ou não;

- Excesso de burocracia, ainda que sejam práticas ágeis;
- Melhora na comunicação entre os envolvidos;

De acordo com os critérios citados foram selecionadas as práticas:

- Seleção Voluntária de Tarefas
- Integração Contínua
- Histórias de Usuário
- Propriedade Coletiva
- Implantação Contínua
- Design Simples
- Refatoração
- Testes
- Padrões de Codificação
- Visão do fluxo de trabalho e riscos
- Limitação do trabalho em progresso
- Tornar o processo explícito
- Implementar ciclos de feedback
- Aumentar colaborativamente e evoluir experimentalmente
- Desenvolvimento Iterativo
- Desenvolvimento Incremental
- Iterações
- Timebox
- Três perguntas
- Reunião Diária
- Kanban board
- Taskboard
- Estimativa Relativa
- Planning Poker
- Backlog
- Tratamento de Backlog (Grooming)
- Realimentação de Informação
- Desenvolvimento em Conjunto

- Opções de decisão
- Decidir no último momento
- Estimular o sistema
- Motivação
- Integridade Conceitual
- Integridade Percebida

Algumas destas práticas não são tarefas a serem executadas explicitamente, mas orientações de coisas consideradas boas a se praticar em um processo de desenvolvimento; isso, de certa forma, é benéfico, pois dá liberdade ao processo tornando-o flexível e, ao mesmo tempo, focalizando objetivos.

#### 4.3.4 PRÁTICAS VS. "COISAS A SE FAZER"

A identificação do relacionamento direto de cada prática com a evolução dos *alphas* é melhor visualizada por seu mapeamento entre as "coisas que sempre fazemos" e seus "espaços de atividades" que podem ser vistos na Figura 12. Estes "espaços de atividades" representam o que deve ser trabalhado dentro dos três domínios Cliente, Solução e Empreitada e não como deve ser executado.

Através destes "espaços de atividades" é possível analisar em quais momentos tais práticas podem ter influência durante o processo de desenvolvimento e, consequentemente, como se deve atuar na evolução dos *alphas*.

O mapeamento trata-se de uma avaliação onde foi analisada a possibilidade de uma prática contribuir ou influenciar em um determinado momento no processo de desenvolvimento, sendo possível uma única prática estar atrelada a diferentes "espaços de atividades" em diferentes domínios como pode ser analisado nos quadros a seguir, relacionando os espaços de atividades com as práticas selecionadas.

	Espaços de Atividades do Domínio: Cliente	o Domínio: Cliente	
Cyclosed Doctibilidador	Compreender as Necessidades	Garantir a Satisfação dos	com otoiD o acriliti
Explorar Possibilidades	dos Stakeholders	Stakeholders	Offilizal O Sisterila
	Histórias de Usuário	Histórias de Usuário	
		Implantação Contínua	
		Testes	
Desenvolvimento		Desenvolvimento	Desenvolvimento
Incremental e Iterativo		Incremental e Iterativo	Incremental e Iterativo
Timebox		Timebox	Timebox
	Backlog	Backlog	
Tratamento de Backlog	Tratamento de Backlog		
Realimentação de		Realimentação de	
Informação		Informação	
Opções de Decisão			
	Decidir no Último Momento		
		Integridade Conceitual	

Quadro 3 - Mapeamento de práticas ágeis e espaços de atividades do domínio Cliente

		Espaços de Atividades do Domínio: Solução	Domínio: Solução		
Compreensão dos Requisitos	Modelar o Sistema	Implementar o Sistema	Testar o Sistema	Implantar o Sistema	Operar o Sistema
		Integração Contínua	Integração Contínua	Integração Contínua	
Histórias de Usuário			Histórias de Usuário		
	Propriedade Coletiva	Propriedade Coletiva			
				Implantação Contínua	
	Design Simples	Design Simples			
	Refatoração	Refatoração			
			Testes		
		Padrões de Código			
Desenvolvimento		Desenvolvimento	Desenvolvimento	Desenvolvimento	
Incremental e Iterativo		Incremental e Iterativo	Incremental e Iterativo Incremental e Iterativo	Incremental e Iterativo	
Timebox			Timebox	Timebox	
Backlog					
Tratamento de Backlog					
Realimentação de Info			Realimentação de Info		
Desenvolvimento em	Desenvolvimento	Desenvolvimento em			
Conjunto	em Conjunto	Conjunto			
Opções de Decisão	Opções de Decisão	Opções de Decisão			
Decidir no Último Momento	Decidir no Último Momento				
Integridade Conceitual e Percebida			Integridade Conceitual e Percebida		
		Limitar o Trabalho em			
		Progresso			

Quadro 4 - Mapeamento de práticas ágeis e espaços de atividades do domínio Solução

	Espaços de At	Espaços de Atividades do Domínio: Empreitada	œ	
Preparar-se para fazer o trabalho	<b>Coordenar atividades</b>	Apoiar a Equipe	Identificar o Progresso	Parar o Trabalho
	Seleção Voluntária de Tarefas			
Histórias de Usuário	Histórias de Usuário			
		Propriedade Coletiva		
			Testes	
Visão de Fluxo de Trabalho	Visão de Fluxo de Trabalho		Visão de Fluxo de Trabalho	
Limitação do Trabalho em	Limitação do Trabalho em			
Progresso	Progresso			
Tornar o Processo Explícito				
		Implementar Ciclos de Feedback		
	Aumentar Colaborativamente, Evoluir Experimentalmente	Aumentar Colaborativamente, Aumentar Colaborativamente, Aumentar Colaborativamente, Evoluir Experimentalmente	Aumentar Colaborativamente, Evoluir Experimentalmente	
	באסומון באסכווויים וומוויים ווני		באסומוו באסכווווייייוני	
Desenvolvimento	Desenvolvimento Incremental		Desenvolvimento Incremental	
Incremental e Iterativo	e Iterativo		e Iterativo	
Timebox	Timebox		Timebox	
	Reunião Diária + 3 perguntas	Reunião Diária + 3 perguntas	Reunião Diária + 3 perguntas	
Taskboard + Kanban	Taskboard + Kanban		Taskboard + Kanban	
Estimativa Relativa	Estimativa Relativa		Estimativa Relativa	
Planning Poker	Planning Poker			
Backlog	Backlog		Backlog	
	Estimular o Sistema	Estimular o Sistema		
		Motivação		

Quadro 5 - Mapeamento de práticas ágeis e espaços de atividades do domínio Empreitada

# 5 AVALIAÇÃO DE ADERÊNCIA DAS PRÁTICAS ÁGEIS

Este capítulo tem por objetivo analisar e discutir os dados obtidos através da aplicação de um questionário que possibilita a avaliação da aderência das práticas ágeis selecionadas ao processo de desenvolvimento existente em startups de software.

#### 5.1 ESTUDO DE CASO

Para a construção deste estudo de caso será utilizado um questionário baseado em respostas na escala de Likert (LIKERT, 1932) que permitirá mensurar experimentalmente um conjunto de comportamentos. O questionário encontra-se no Apêndice A.

A amostra foi composta por 28 profissionais que trabalham ou trabalharam em startups de *software* e são, ou eram, ligados ao time de desenvolvimento de alguma forma. Na maior parte das respostas obtidas contou-se com desenvolvedores de *software*, mas também foram obtidas respostas de diretores de marketing, operação, inovação, sócios da startup, profissionais da área de marketing e gerentes em geral.

A disseminação do questionário foi feita através de redes sociais, emails, grupos diretamente ligados a empreendedorismo, incubadoras e parques tecnológicos. Dentre os 28 respondentes, 27 aceitaram participar do presente estudo, o aceite deste questionário encontra-se em propriedade do autor deste trabalho.

## 5.2 MÉTODO

As 34 práticas selecionadas foram reduzidas a um conjunto de 26 devido ao agrupamento por similaridades como, por exemplo, a implantação contínua e a integração contínua.

O questionário foi dividido em seis seções, a primeira com o objetivo de descrição da população estudada. E as cinco seguintes com as questões referentes à avaliação de aderência das práticas selecionadas ao time de desenvolvimento.

Para a avaliação da aderência das práticas ágeis foram utilizadas escalas Likert de "1" (Não concordo totalmente) a "5" (Concordo totalmente). Essa escala possui ordem de classificação mas os intervalos não são considerados. A utilização de média para escalas de Likert e testes paramétricos são inapropriadas, pois se tratam de dados ordinais e devem ser tratados a partir de mediana ou moda (JAMIENSON, 2004).

Foi desenvolvido um total de 32 questões referentes às práticas selecionadas, 26 afirmações diretas capazes de medir a atitude do respondente com alta ou baixa concordância referente à afirmação e seis outras questões para reforçar alguma característica das práticas às quais estejam relacionadas, para procurar algo mais específico ou para apresentar uma antítese de teste da compreensão do entrevistado (Apêndice B).

O questionário foi divulgado utilizando a ferramenta de formulários contida no Google Drive®; sua divulgação foi feita através de emails para grupos específicos ligados a startups e empreendedorismo, emails para fundadores de startups incubadas em parques tecnológicos brasileiros, sites e pessoas liagadas ao tema.

Os dados deste trabalho foram analisados com a ferramenta Microsoft Excel®.

As questões relacionadas às práticas foram divididas em cinco grupos para melhor organização e, além disso, as questões nestes grupos foram embaralhadas de forma distinta para cada respondente, de modo que nenhuma sequência predeterminada em respostas anteriores influencie nas respostas seguintes.

#### 5.3 ANÁLISE DOS RESULTADOS

Analisando os perfis dos respondentes da amostra foi possível identificar uma idade média aproximada de 27 anos com um desvio padrão de aproximadamente cinco anos, o que indica um perfil esperado dos participantes de startups, representado por pessoas jovens abaixo dos 30 anos de idade.

A tabela 6 mostra o tempo de experiência dos respondentes na funçãoe exercida, onde é possível destacar que 44,4% possuem entre 1 e 3 anos de experiência e 37% entre 4 e 10 anos, o que é um tempo de experiência considerado muito grande para pessoas ligadas a startups.

Tempo	Número absoluto	Frequência relativa (%)
Menos de 1 ano	4	14,8
Entre 1 e 3 anos	12	44,4
4 e 10 anos	10	37
Mais de 10 anos	1	3,7

Tabela 6 - Experiência na função desempenhada

Com relação à quantidade de pessoas empregadas nas empresas *startups* utilizadas como parâmetro pelos respondentes, 59,3% para até 9 pessoas e 37% para de 10 a 49 pessoas, resultando que em 96,3% das *startups* se enquadram na definição de micro/pequena empresa.

#### **5.3.1 ANÁLISE DESCRITIVA**

A compreensão do grau de aderência de cada prática de forma independente depende de uma análise de cada conjunto de respostas dado às 32 perguntas. O conjunto de respostas pode ser analisado graficamente no Apêndice D deste trabalho, a relação de cada prática e sua pergunta correspondente pode ser encontrada no Apêndice B e as respostas de todos os candidatos estão no Apêndice C.

Por utilizar a escala de Likert variando de "1" a "5", onde "1" representa "não concordo totalmente" e "5" "concordo totalmente" para facilitar a análise dos conjuntos de respostas serão consideradas as respostas de grau "1" e "2" como representantes da baixa utilização de uma determinada prática, 3 como medida neutra para "não concordo nem discordo", e "4" e "5" como representantes da utilização de uma prática específica.

O grau de utilização de uma prática será analisado através de quatro grupos sendo:

- pouco utilizada se sua utilização for menor ou igual a 40%;
- medianamente utilizada se sua utilização for maior que 40% e menor ou igual a 50%;
- utilizada se sua utilização for maior que 50% e menor ou igual a 60% e;
- muito utilizada se sua utilização for maior que 60%.

Uma prática é considerada muito utilizada se sua aderência estiver acima de 60%, pois isso significa que seu uso é maior que a somatória dos percentuais que representam cada uma das faixas de respostas "1" e "2" (baixa utilização), e 3" (elemento neutro na escala Likert), onde cada um dos valores representa uma fatia de 20% na escala.

No Quadro 6 segue, os percentuais de cada prática obtidos através das respostas ao questionário.

As práticas "Implementar Ciclos de Feedback", "Integridade Conceitual e Percebida", "Planning Poker" e "Tratamento de Backlog (Grooming)" foram ainda avaliadas através de uma questão de antítese ou de reforço, portanto seu percentual de utilização é construído através da média das duas questões.

Práticas	Não utilização	Neutro	Utilização
Praticas	(%)	(%)	(%)
Seleção Voluntária de Tarefas	25,9	29,6	44,4
Integração Contínua	25,9	25,9	48,1
Implantação Contínua	33,3	14,8	51,9
Histórias de Usuário	37	18,5	44,4
Propriedade Coletiva	14,8	22,2	63
Design Simples	29,6	33,3	37
Refatoração	11,1	22,2	66,7
Testes - Código	33,3	18,5	48,1
Testes - Aplicação	34,6	19,2	46,2
Padrões de Codificação	18,5	22,2	59,3
Visão do fluxo de trabalho e			
riscos/Kanban Board	18,5	18,5	63
Limitação do trabalho em			
progresso	33,3	18,5	48,1
Tornar o processo explícito	18,5	11,1	70,4
Implementar ciclos de			
feedback	25	17,3	57,7
Aumentar colaborativamente			
e evoluir experimentalmente	11,5	30,8	57,7
Desenvolvimento Iterativo,			
Incremental e timeboxed	15,4	19,2	65,4
Três perguntas e Reunião			
Diária	34,6	26,9	38,5
Taskboard	15,4	3,8	80,8
Estimativa Relativa	20	28	52
Planning Poker	45,1	19,75	35,15
Backlog	19,2	19,2	61,5
Tratamento de Backlog			
(Grooming)	25,45	11,9	62,6
Realimentação de Informação	26,9	26,9	46,2
Desenvolvimento em			•
Conjunto	26,9	15,4	57,5
Opções de decisão e Decidir			•
no último momento	29,2	33,3	37,5
Estimular o sistema	15,4	23,1	61,5
Motivação	11,5	23,1	65,4
Integridade Conceitual e			•
Percebida	19,2	15,4	65,4

Quadro 6 - Percentuais de utilização de práticas

Algumas considerações podem ser feitas sobre a utilização ou não utilização de determinadas práticas no contexto das *startups* analisadas.

A alta utilização de "Propriedade Coletiva", com 63% de uso, representa a importância do nível de engajamento do time responsável pelo desenvolvimento do produto dentro de uma *startup*.

A pouca utilização de "Design Simples", 37% indica que os times tentam prever a evolução do produto ao projetá-lo produzindo código que não será necessariamente utilizado no exato momento.

A alta utilização de "Refatoração", 66,7% representa a busca por um código que se mantenha constantemente coeso, limpo e manutenível. Apesar de certas refatorações serem custosas, se forem praticadas desde o início do projeto, a longo prazo resultarão sempre em um código fácil de se trabalhar.

A mediana utilização de "Testes", tanto em nível de código quanto aplicação, é compreensível neste contexto, pois, testar um projeto completamente exige recursos que normalmente não estão disponíveis no início de operação das *startups*, nesse caso, a mescla entre "testes parciais de código" e "aplicação" são uma forma de garantir certo nível de qualidade do produto.

Com 59,3% de utilização, quase pertencente à categoria de alta utilização, a utilização de "Padrões de Codificação" indica a importância de todos codificarem seguindo um mesmo padrão em sua escrita de código.

Com alta utilização de 63% é possível afirmar que a "Visão do Fluxo de Trabalho e Riscos" e o uso de "Kanban Board" são altamente utilizados pois permitem a identificação de problemas, riscos e gargalos, evitando possíveis problemas de percurso na execução do projeto.

A alta utilização de 70,4% indica que "Tornar o Processo Explícito" é uma prática importante para que haja transparência na forma de trabalho de todo o time envolvido no desenvolvimento.

As práticas "Desenvolvimento Iterativo, Incremental" e "*Timeboxed*" com alta utilização de 65,4% reforçam um dos princípios ágeis de que pequenas entregas proporcionam mais valor a seus *stakeholders* reduzindo o risco durante o projeto e proporcionando maior adaptabilidade a mudanças.

"Planning Poker" com apenas 28% de utilização indica que o uso de métricas como Fibonacci ou ordens específicas de grandezas não é algo muito utilizado. Sua questão antítese reforça esta afirmação com 46,2% dos respondentes concordando que o time não utiliza métricas que representem a grandeza de forma quantitativa para as tarefas. A priori, era esperado que alguma métrica específica poderia ser utilizada, mas os dados mostram que, talvez devido à dinâmica e velocidade deste ambiente, alguma outra forma de mensuração do custo de tarefas deve ser utilizada.

A prática "Taskboard", com alta utilização de 80,8%, reforça os 63% de utilização de práticas ligadas ao "Kanban" e "Visualização de Fluxo de Trabalho".

Com alta utilização de 61,5% a prática de "Backlog" indica que o uso de listas de coisas a serem futuramente desenvolvidas e organizadas de acordo com sua importância contribui como ferramenta de priorização de entregas, sendo complementada pela também alta utilização da prática de "Tratamento de Backlog (Grooming)" de 62,2%, que indica que há a discussão em maior detalhamento de cada item a ser futuramente desenvolvido como forma de esclarecimento e melhor compreensão da evolução do produto.

A prática "Estimular o Sistema", com alta utilização de 61,5%, mostra que os respondentes reforçam a importância de um time ágil que possa se auto gerenciar e se sinta independente, com liberdade para tomar suas próprias decisões.

Com 65%, representando alta utilização, a prática "Motivação" indica a importância do time compartilhar senso de propósito e engajamento com o que está sendo feito, de se reconhecer como contribuintes diretos da construção de algo com maior retorno.

As práticas "Opções de decisão" e "Decidir no Último Momento", com pouca utilização de 37,5%, indicam equilíbrio esperado em um ambiente de uma *startup,* onde certos riscos são assumidos como apostas para ganho de mercado, muitas vezes sem haver garantia do retorno esperado.

### 5.3.2 DIVISÃO POR UTILIZAÇÃO

Dividindo as práticas em grupos com figuras de utilização parecidas, pode se chegar ao Quadro 7.

Utilização <= 40% (pouco utilizada)	Design Simples, Planning Poker, Três perguntas, Reunião Diária, Opções de Decisão, Decidir no Último Momento
Utilização > 40% e <= 50% (medianamente utilizada)	Seleção Voluntária de Tarefas, Integração Contínua, História de Usuário, Testes (Código), Testes (Aplicação), Limitação do Trabalho em Progresso, Realimentação de Informação
Utilização > 50% e <= 60% (utilizada)	Implantação Contínua, Padrões de Codificação, Implementar Ciclos de Feedback, Aumentar Colaborativamente e Evoluir Experimentalmente, Estimativa Relativa, Desenvolvimento em Conjunto
Utilização > 60% (muito utilizada)	Propriedade Coletiva, Refatoração, Visão do Fluxo de Trabalho e Riscos, Kanban Board, Tornar o Processo Explícito, Desenvolvimento Iterativo, Desenvolvimento Incremental, Desenvolvimento Timeboxed, Integridade Conceitual, Integridade Percebida, Taskboard, Backlog, Tratamento de Backlog (Grooming), Estimular o Sistema, Motivação

Quadro 7 - Divisão de práticas por seu percentual de utilização

Com esta divisão é possível analisar a cobertura do conjunto de práticas de cada faixa de utilização de acordo com as "coisas que sempre fazemos" presentes nos 3 domínios do SEMAT *Kernel*, que foram apresentadas na seção 4.3.4.

Nos quadros seguintes é possível visualizar a cobertura das quatro faixas de utilização, demonstradas pelo código de cores da seguinte forma:

• Amarelo: prática muito utilizada;

• Verde: prática utilizada;

• Azul: prática medianamente utilizada;

• Cinza: prática não utilizada/pouco utilizada;

	Espaços de Atividades do Domínio: Cliente	do Domínio: Cliente	
Explorar Possibilidades	Compreender as Necessidades dos Stakeholders	Garantir a Satisfação dos Stakeholders	Utilizar o Sistema
	Histórias de Usuário	Histórias de Usuário	
		Implantação Contínua	
		Testes (Código e Aplicação)	
Desenvolvimento		Desenvolvimento	Desenvolvimento
Incremental e Iterativo		Incremental e Iterativo	Incremental e Iterativo
Timebox		Timebox	Timebox
	Backlog	Backlog	
Tratamento de Backlog	Tratamento de Backlog		
Realimentação de		Realimentação de	
Informação		Informação	
Opções de Decisão			
	Decidir no Último Momento		
		Integridade Conceitual	

Quadro 8 - Mapeamento da cobertura de práticas por utilização nas atividades do domínio Cliente

		Espaços de Atividades do Domínio: Solução	omínio: Solução		
Compreensão dos Requisitos	Modelar o Sistema	Modelar o Sistema Implementar o Sistema	Testar o Sistema	Implantar o Sistema	Operar o Sistema
		Integração Contínua	Integração Contínua	Integração Contínua	
Histórias de Usuário			Histórias de Usuário		
	Propriedade Coletiva	Propriedade Coletiva			
				Implantação	
	Design Simples	Design Simples			
	Refatoração	Refatoração			
			Testes (Código e Aplicação)		
		Padrões de Código			
Desenvolvimento		Desenvolvimento	Desenvolvimento	Desenvolvimento	
Incremental e Iterativo		Incremental e Iterativo	Incremental e Iterativo	Iterativo	
Timebox			Timebox	Timebox	
Backlog					
Tratamento de Backlog					
Realimentação de Info			Realimentação de Info		
Desenvolvimento em	Desenvolvimento	Desenvolvimento em			
Conjunto	em Conjunto	Conjunto			
Opções de Decisão	Opções de Decisão	Opções de Decisão			
Decidir no Último Momento	Decidir no Último Momento				
Integridade Conceitual e			Integridade Conceitual		
Percebida			e Percebida		
		Limitar o Trabalho em			
		Progresso			

Quadro 9 - Mapeamento da cobertura de práticas por utilização nas atividades do domínio Solução

	Espaços de At	Espaços de Atividades do Domínio: Empreitada	e	
Preparar-se para fazer o trabalho	<b>Coordenar atividades</b>	Apoiar a Equipe	Identificar o Progresso	Parar o Trabalho
	Seleção Voluntária de Tarefas			
Histórias de Usuário	Histórias de Usuário			
		Propriedade Coletiva		
			Testes (Código e Aplicação)	
Visão de Fluxo de Trabalho	Visão de Fluxo de Trabalho		Visão de Fluxo de Trabalho	
Limitação do Trabalho em	Limitação do Trabalho em			
Progresso	Progresso			
Tornar o Processo Explícito				
		Implementar Ciclos de		
		Feedback		
	Aumentar Colaborativamente, Evoluir Experimentalmente	Aumentar Colaborativamente, Aumentar Colaborativamente, Evoluir Experimentalmente	Aumentar Colaborativamente, Evoluir Experimentalmente	
Desenvolvimento	Desenvolvimento Incremental		Desenvolvimento Incremental	
Incremental e Iterativo	e Iterativo		e Iterativo	
Timebox	Timebox		Timebox	
	Reunião Diária + 3 perguntas	Reunião Diária + 3 perguntas	Reunião Diária + 3 perguntas	
Taskboard + Kanban	Taskboard + Kanban		Taskboard + Kanban	
Estimativa Relativa	Estimativa Relativa		Estimativa Relativa	
Planning Poker	Planning Poker			
Backlog	Backlog		Backlog	
	Estimular o Sistema	Estimular o Sistema		
		Motivação		

Quadro 10 - Mapeamento da cobertura de práticas por utilização nas atividades do domínio Empreitada

Na análise dos quadros nota-se que o conjunto das práticas em amarelo, com sua utilização acima de 60% de acordo com os respondentes, há pelo menos uma prática do conjunto de alta utilização presente em cada um dos "espaços de atividades" dos três domínios.

## **5.4 DISCUSSÃO**

Analisando os mapeamentos de práticas nos "espaços de atividades" pertencentes aos três domínios e classificando devidamente cada prática mapeada de acordo com seu grau de utilização, constatou-se que o conjunto de práticas muito utilizadas está presente em todas as atividades dos três domínios.

Isto significa que o conjunto de práticas mais utilizadas atende todos os aspectos que são considerados a essência da engenharia de *software* pelo SEMAT *kernel*, que propõe que se um projeto deve progredir em todas estas nuances ou terá problemas durante a sua empreitada (SEMAT, 2012).

Outra constatação importante é que a distribuição da utilização das práticas não é homogênea, isto é, não há um subconjunto de práticas sobressalentes que indique que estas *startups* tenham preferência por algum dos métodos, mas que praticam a composição de práticas que melhor se aplicam às suas necessidades.

Por se tratar da utilização de práticas de forma heterogênea confirma-se a hipótese de que o melhor método para um ambiente ágil e dinâmico como o de uma *startup* é composto de práticas de diversos métodos distintos (YAU; MURPHY, 2013).

O conjunto de práticas medianamente utilizadas e práticas pouco utilizadas reforçam a afirmação de que práticas que envolvam maior número de recursos utilizados como as práticas de "planning poker", "testes" de um modo geral e "reunião diária" não são tão praticadas devido ao alto custo envolvido (GIARDINO, 2014).

Com a análise das práticas contidas nos métodos ágeis mais utilizados (VERSIONONE, 2014) e que são pertinentes ao contexto de uma *startup*, constatou-se através deste trabalho um conjunto de práticas altamente utilizadas, sendo elas:

- Propriedade Coletiva;
- Refatoração;
- Visão do Fluxo de Trabalho e Riscos;
- Kanban Board;
- Tornar o Processo Explícito;
- Desenvolvimento Iterativo;
- Desenvolvimento Incremental;
- Desenvolvimento Timeboxed;
- Integridade Conceitual;
- Integridade Percebida;
- Taskboard;
- Backlog;
- Tratamento de Backlog (Grooming);
- Estimular o Sistema;
- Motivação.

Com este conjunto de alta utilização entre as práticas estudadas é possível afirmar que para uma *startup* cobrir todos os aspectos mínimos e necessários presentes na essência da engenharia de *software*, ela deve começar por este conjunto de práticas em sua forma de trabalho.

O termo "conjunto mínimo" será utilizado para se referir ao conjunto de alta utilização de práticas que atendem a todos os aspectos da essência da engenharia de software.

Apesar de ter sido encontrado um conjunto mínimo de práticas ágeis para a execução de empreitadas em *startups*, a adição de práticas de outras faixas de utilização pode ser feita de acordo com a sua necessidade para que haja um

processo que melhor se ajuste ao ambiente. Isso significa que práticas de métodos menos utilizados e não considerados nesta pesquisa também podem agregados a este conjunto mínimo, caso haja alguma especificidade que necessite dele.

Constatou-se também que o conjunto mínimo é majoritariamente composto por práticas relacionadas às relações humanas em um projeto e não naquelas puramente técnicas, o que mostra certa humanização do desenvolvimento de *software,* mantendo-se alinhado aos valores contidos no manifesto ágil (BECK ET AL., 2001).

Apesar da amostra utilizada neste trabalho ser considerável e permitir a descoberta de um conjunto mínimo de práticas, ela possui limitações com relação a distribuição geográfica e perfil de respondentes. A aplicação do questionário em outros grupos, por exemplo, pode melhorar o escopo desse trabalho.

# **6 CONSIDERAÇÕES FINAIS**

A utilização do SEMAT *Kernel* como ferramenta para mapeamento e composição de práticas, independente de seus métodos, permitiu a visualização do alcance de cada prática com relação às atividades a serem desempenhadas nos três domínios ali definidos, que, se executadas proporcionam o bom andamento de um projeto.

Os resultados obtidos nesta pesquisa mostram que a divisão de práticas ágeis pela incidência de utilização permite a identificação de um conjunto mínimo que permeia todas as atividades a serem desempenhadas nos três domínios do SEMAT *Kernel*.

Este conjunto mínimo encontrado serve como um ponto inicial para o desenvolvimento de projetos ágeis nestes ambientes dinâmicos e que envolvem alto risco.

Devido a este conjunto mínimo ser composto de práticas de métodos distintos e não haver um subconjunto de práticas de um determinado método, neste conjunto mínimo, que se sobressaia, fica confirmada a afirmação inicial identificada através do levantamento bibliográfico de que é melhor para uma *startup* o uso de um método composto de diversas práticas que atendam as peculiaridades deste tipo de ambiente.

Identifica-se que as práticas ágeis que envolvem os maiores gastos de recursos são as menos utilizadas neste tipo de ambiente, enquanto práticas com foco no relacionamento humano e engajamento de equipe mostraram-se as de maior valor para este ambiente.

#### 6.1 CONTRIBUIÇÕES

Dentre as contribuições deste trabalho para o desenvolvimento de *software* ágil e sua aplicação em ambientes de *startups* de *software*, é possível destacar:

- Classificação das práticas ágeis mais utilizadas sob a ótica do SEMAT Kernel, encontrando os domínios em que se encontram cada uma destas práticas;
- (2) Mapeamento de cada uma das práticas ágeis identificadas nas atividades ligadas a cada um dos três domínios do SEMAT Kernel, possibilitando a visualização clara de seu alcance quanto às atividades essenciais para o progresso dos alphas;
- (3) Verificação através de questionário da faixa de aderência ao ambiente de startups de cada uma das práticas ágeis eleitas independentes de seus métodos;
- (4) Identificação de um conjunto mínimo de práticas ágeis com alta aderência ao ambiente de startups que serve como conjunto de práticas iniciais mínimas para a aplicação de um método ágil composto para as especificidades deste contexto;

#### 6.2 TRABALHOS FUTUROS

Os resultados desta pesquisa, construída com base no levantamento de literatura e questionário aplicado a um contexto específico deste trabalho, devem ser validados de forma direta com sua aplicação ao contexto de *startups* para que haja a confirmação das constatações feitas.

O ponto inicial para a continuidade desta pesquisa é a proposta de um método composto pelas práticas de alta aderência identificadas neste trabalho através da linguagem para a descrição de métodos fornecida pelo SEMAT. Isso permitirá não somente a utilização clara deste método composto, mas a sua comparação direta com outros métodos também descritos nesta mesma linguagem.

Os dados obtidos pelo questionário utilizado por esta pesquisa podem fornecer outra visão sobre a análise das práticas independentes de seus métodos. Estes mesmos dados podem ser analisados sob a forma de um grafo direcionado onde seus vértices representam as práticas e o peso de suas arestas indicam a força da

utilização de uma prática específica com relação à outra prática que podem ser calculados, por exemplo, através da tabela de adjacência abaixo.

Escala de Resposta	5	4	3	2	1
5	3	2	1	-2	-3
4	2	3	1	-2	-2
3	1	1	3	1	1
2	-2	-2	1	3	2
1	-3	-2	1	2	3

Figura 14 - Tabela de adjacência para análise por grafo

O grafo gerado pelo quadrante superior esquerdo, destacado pela cor azul, possui os pesos que indicam forte relação entre uma prática e outra e resultará em um grafo que mostre apenas as fortes relações entre as práticas.

O grafo gerado pelo quadrante inferior direito, destacado pela cor amarela, possui os pesos que indicam fraca relação entre uma prática e outra e resultará em um grafo que mostre apenas as fracas relações entre as práticas.

O grafo gerado pelo quadrante inferior esquerdo e superior direito, destacado pela cor verde, possui os pesos que indicam a concordância com a utilização de uma determinada prática e a discordância com a outra.

A análise destes grafos permitirá um maior aprofundamento nas práticas ágeis aplicadas ao ambiente de *startups* de *software*, permitindo verificar relações como quais outras práticas menos aderentes são utilizadas em conjunto com as mais aderentes e qual o grau de força da relação entre as práticas mais aderentes.

Esta análise por grafo exigirá a utilização do ferramental matemático sobre as redes formadas por estas práticas, e que está fora do escopo deste trabalho, podendo proporcionar novas visões sobre as práticas estudadas.

## **REFERÊNCIAS**

ABBAS, N.; GRAVELL, A.; WILLS, G. Historical roots of Agile methods: where did "Agile thinking" come from? **Agile Processes in Software Engineering and Extreme Programming**, p. 94–103, 2008.

AGILE ALLIANCE, Agile Practices Subway Map, 2015. Disponível em: <a href="http://guide.agilealliance.org/subway.html/">http://guide.agilealliance.org/subway.html/</a>

ALLEMAN, G. Making agile development work in a government contracting environment- measuring velocity with earned value. **Agile Development Conference**, **2003**. **ADC 2003**. **Proceedings of the**, p. 114–119, 2003.

ANDERSON, DAVID J. **Kanban: successful evolutionary change in your technology business**. Sequim, Washington: Blue Hole Press, 2010. Print.

PERKINS, A. B., PERKINS, M. C. The Internet Bubble: Inside the Overvalued World of High-Tech Stocks -- And What You Need to Know to Avoid the Coming Catastrophe, HarperInformation, 1999.

BARBOSA, A.; LACERDA, G. Estudo Investigativo da Adoção do CMMI e das Metodologias Ágeis em Empresas Brasileiras. [s.d.].

BECK, K. **Test Driven Development: By Example**. Addison Wesley Longman, p. 240, 2002.

BECK, K. Embracing change with extreme programming. Computer, 32(10), p. 70–77, 1999.

BECK, K. Extreme Programming Explained. Embrace change, 1996.

BECK, K.; BEEDLE, M.; BENNEKUM, A. VAN. **Manifesto for agile software development**. 2001.

Disponível em: <a href="http://agilemanifesto.org/">http://agilemanifesto.org/</a>

BROOKS, F. P. **The Mythical Man-Month** (Anniversary Ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

C. NOBEL, Why Companies Fail - and How Their Founders Can Bounce Back. Harvard Business School, 2011.

COHEN, D.; LINDVALL, M.; COSTA, P. **An introduction to agile methods**. Advances in computers, v. 62, n. 03, p. 1–66, 2004.

DULLEMOND, K.; VAN GAMEREN, B.; VAN SOLINGEN, R. Supporting distributed software engineering in a fully distributed organization. **Cooperative and Human Aspects of Software Engineering (CHASE), 2012 5th International Workshop on**, p. 30–36, jun. 2012.

Essence – Kernel and Language for Software Engineering Methods - OMG Submission, 2014.

Disponível em: <a href="http://www.omg.org/spec/Essence/">http://www.omg.org/spec/Essence/</a>

FAYAD, M. E., LAITINEN, M., & WARD, R. P. (2000). **Thinking objectively: software engineering in the small**. Communications of the ACM, 43(3), p. 115–118, 2000.

FERNANDES, J. M.; ALMEIDA, M. Classification and Comparison of Agile Methods. Seventh International Conference on the Quality of Information and Communications Technology, p. 391–396, set. 2010.

FOWLER, M. Is design dead? **Software Development-San Francisc**o, p. 1–14, 2001.

FUNDAÇÃO DOM CABRAL. Causas da Mortalidade de Startups Brasileiras. O que fazer para aumentar as chances de sobrevivência no mercado?, 2014.

Disponível em:

<a href="http://www.fdc.org.br/blogespacodialogo/Documents/2014/causas\_mortalidade\_startups\_brasileiras.pdf">http://www.fdc.org.br/blogespacodialogo/Documents/2014/causas\_mortalidade\_startups\_brasileiras.pdf</a>

GIARDINO, C., UNTERKALMSTEINER, M., PATERNOSTER, N., GORSCHEK, T., & ABRAHAMSSON, P. What Do We Know about Software Development in Startups? IEEE Software, 31(5), p. 28–32, 2014.

HAUGEN, N. An empirical study of using planning poker for user story estimation. **Agile Conference**, 2006.

HIGHSMITH, J. **Agile Software Development Ecosystems. Solutions**, p. 448, 2002.

HIGHSMITH, J., & Cockburn, A. Agile software development: the business of innovation. **Computer**, 34, 2001.

HÖST, M.; REGNELL, B.; WOHLIN, C. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. **Empirical Software Engineering**, v. 5, n. 3, p. 201–214, 2000.

HUMBLE, J.; READ, C.; NORTH, D. The deployment production line. **Agile Conference**, 2006, p. 113–118, 2006.

JACOBSON, I.; MEYER, B.; SOLEY, R. The SEMAT initiative: A call for action. **Dr. Dobb's Journal**, v. 10, 2009.

JACOBSON, I., ET AL. The essence of software engineering: applying the **SEMAT kernel**. Upper Saddle River, NJ: Addison-Wesley, 2013.

JAKOBSEN, C. R.; SUTHERLAND, J. Scrum and CMMI - Going from Good to Great. **Agile Conference**, p. 333–337, ago. 2009.

JAMIESON, S. Likert scales: how to (ab)use them. **Medical education**, v. 38, n. 12, p. 1217–8, dez. 2004.

JEFFRIES, R. E. What is Extreme Programming? 1999. Disponível em: <a href="http://xprogramming.com/what-is-extreme-programming">http://xprogramming.com/what-is-extreme-programming></a>

JOHNSON, P.; EKSTEDT, MATHIAS; JACOBSON, I., Where's the Theory for Software Engineering?, IEEE Software, vol.29, no.5, p. 96, 2012.

KRUCHTEN, P. Contextualizing agile software development. **Journal of Software: Evolution and Process**, v. 25, n. 4, p. 351–361, 2013.

LAGO, L. S. M. Fatores humanos na dependabilidade de sistemas de software desenvolvidos com práticas ágeis, p. 35-40, Escola Politécnica da Universdade de São Paulo, 2014.

LECHNER, M. XP Team Psychology-An Inside View. **PPIG 2008: Psychology of Programming Interest Group**, 2008.

LIKERT, R. A Technique for the Measurement of Attitudes. **Archives of Psychology**, 140: pp. 1-55. 1932.

MACKENZIE, A.; MONK, S. From Cards to Code: How Extreme Programming Re-Embodies Programming as a Collective Practice. **Computer Supported Cooperative Work (CSCW)**, v. 13, n. 1, p. 91–117, mar. 2004.

MARTIN, R. C. Iterative and incremental development (IID). **C++ Report**, 1999.

MAURER, F.; MARTEL, S. Extreme programming: Rapid development for Webbased applications. **IEEE Internet computing**, v. 6, n. 1, p. 86–90, 2002.

MCCONNEL, S. Daily Build and Smoke Test. **Best Practices IEEE**, v. 13, n. 4, p. 144–144, 1996.

MIRANDA, E.; BOURQUE, P. Agile monitoring using the line of balance. **Journal of Systems and Software**, v. 83, n. 7, p. 1205–1215, jul. 2010.

MIRANDA, E. Improving Subjective Estimates Using Paired Comparisons. n. February, p. 87–91, 2001.

MORAN, T. P. et al. Design and Technology for Collaborage: Collaborative Collages of Information on Physical Walls. v. 1, p. 197–206, 1999.

NAUR, P., RANDELL, B. (1969) Software Engineering: Report of a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, Brussels, Scientific Affairs Division, NATO, p. 7-11, 1968.

PAASIVAARA, M.; DURASIEWICZ, S.; LASSENIUS, C. Using Scrum in Distributed Agile Development: A Multiple Case Study. **2009 Fourth IEEE International Conference on Global Software Engineering**, p. 195–204, jul. 2009.

PATRÍCIO, N. S. Engenharia de requisitos em software para e-learning, **Escola Politécnica da Universidade de São Paulo**, 2013.

PAULK, MARK C., Weber, Charles V., Curtis, Bill, Chrissis, Mary B. **The Capability Maturity Model: Guidelines for Improving the Software Process**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

PERRY, T. Drifting Toward Invisibility: The Transition to the Electronic Task Board. **Agile 2008 Conference**, p. 496–500, 2008.

POPPENDIECK, M., POPPENDIECK, T. Lean software development: an agile toolkit. Boston, Mass: Addison-Wesley, 2003. Print.

FAIRLIE, R. W., **Kauffman Index of Entrepreneurial Activity**. Kauffman Foundation, 2013.

FAIRLIE, R. W., State of Entrepreneurship Address. Kauffman Foundation, 2014.

RAFIQUE, Y.; MISIC, V. The Effects of Test-Driven Development on External Quality and Productivity: A Meta-Analysis. IEEE Trans. Softw. Eng. p. 835-856, 2013.

ROYCE, W. W. Managing the Development of Large Software Systems. IEEE WESCON, p. 1–9, 1970.

RUBIN, K. S. Essential Scrum a practical guide to the most popular agile process. Upper Saddle River, NJ: Addison-Wesley, 2012.

SCHWABER, K., SUTHERLAND, J. The Scrum guide, 2001.

SCHWABER, K.; SUTHERLAND, J. Guia do SCRUM, Rafael Sabbagh. Disponível em: <a href="http://scholar.google.com/scholar?">http://scholar.google.com/scholar?</a> hl=en&btnG=Search&g=intitle:Guia+do+Scrum#0>.

SEBRAE NACIONAL, O que é uma startup?, 2014. Disponível em: <a href="http://www.sebrae.com.br/sites/PortalSebrae/artigos/O-que-%C3%A9-uma-startup%3F">http://www.sebrae.com.br/sites/PortalSebrae/artigos/O-que-%C3%A9-uma-startup%3F</a>

SEBRAE NACIONAL. Dieese. Anuário do trabalho na micro e pequena empresa, p. 17, 2013.

Disponível em: <a href="https://www.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/Anuario%20do%20Trabalho">www.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/Anuario%20do%20Trabalho</a>

%20Na%20Micro%20e%20Pequena%20Empresa\_2013.pdf>

SUGIMORI, Y. et al. Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system. **International Journal of Production Research**, v. 15, n. 6, p. 553–564, jan. 1977.

SUTTON, S. M. The role of process in software start-up. IEEE Software, 17, p. 33–39, 2000.

THE STANDISH GROUP, The CHAOS Report (1994). Report of the Standish Group,

1995.

Disponível em:

<a href="http://www.standishgroup.com/sample\_research\_files/chaos\_report\_1994.pdf">http://www.standishgroup.com/sample\_research\_files/chaos\_report\_1994.pdf</a>

VERSIONONE. State of Agile Development Survey, 2007.

VERSIONONE. State of Agile Development Survey, 2008.

VERSIONONE. State of Agile Development Survey, 2009.

VERSIONONE. State of Agile Development Survey, 2010.

VERSIONONE. State of Agile Development Survey, 2011.

VERSIONONE. State of Agile Development Survey, 2012.

VERSIONONE. State of Agile Development Survey, 2013.

VERSIONONE. State of Agile Development Survey, 2014.

WRAY, S. How pair programming really works. **Software**, **IEEE**, 2010.

YAU, A., & MURPHY, C. Is a Rigorous Agile Methodology the Best Development Strategy for Small Scale Tech Startups?, p. 9, 2013. Disponível em: <a href="http://repository.upenn.edu/cis\_reports/980/">http://repository.upenn.edu/cis\_reports/980/</a>

# **APÊNDICE A – QUESTIONÁRIO**

# Pesquisa de aderência de práticas ágeis no ambiente de desenvolvimento em startups de software

Este questionário faz parte de uma das pesquisas de mestrado em desenvolvimento no Laboratório de Engenharia do Conhecimento da Escola Politécnica da Universidade de São Paulo.

Esta pesquisa visa identificar qual a aderência das principais práticas ágeis isoladamente de seus métodos na forma de trabalho no ambiente de uma startup de software.

\*Para responder a este questionário você deve trabalhar ou ter trabalhado em uma startup de software.

#### Sobre o respondente

Perguntas sobre você para auxiliar na identificação da população do estudo.

Qual a quantidade de pessoas que trabalham ou trabalhavam na startup que está utilizando como parâmetro para responder a esta pesquisa? \* \*obrigatório Até 9 pessoas De 10 a 49 pessoas De 50 a 99 pessoas 100 ou mais empregados Email \*opcional Função desempenhada na startup \* \*obrigatório Idade \* \*obrigatório Nome da startup em que trabalha ou trabalhou \*opcional Nome \*opcional Qual é o seu tempo de experiência em sua função? \* \*obrigatório Menos de 1 ano Entre 1 e 3 anos 4 e 10 anos Mais de 10 anos Você aceita participar deste questionário? \*

Todas as respostas desse questionário serão usadas apenas para fim de pesquisa, sendo o resultado utilizado em um trabalho de mestrado para a Universidade de São Paulo.

- Aceito
- Não aceito

# Pesquisa de aderência de práticas ágeis em startups de software (1 de 5)

Independente do grau de senioridade dos integrantes do time os desenvolvedores podem escolher, dentre um conjunto de tarefas, quaisquer uma independente do seu grau de dificuldade para sua execução.

para sua execução.						
	1	2	3	4	5	
Não concordo totalmente			0			Concordo totalmente
Durante o desenvolvime pequenas refatorações	par	a m	ante	ê-lo	ínte	
	1	2	3	4	5	
Não concordo totalmente	0					Concordo totalmente
	tuit	o de		rem	ând	ios ou textos simples, sobre o que deve ser coras para maiores discussões com os clientes.
Não concordo totalmente						Concordo totalmente
do trabalho desenvolvid	<b>o p</b> 1	<b>or t</b>	odo 3	<b>s o</b> s	5 <b>de</b> :	
São desenvolvidos apen tentam prever o futuro.			ligo:			tendam ao requisito solicitado, sem adição de coisas qu
Não concordo totalmente	0	0	0	0	0	Concordo totalmente
	res	ser	nten			oprietário do que é desenvolvido, sendo responsáveis po
sua qualidade tanto em	mo	dela				•

# Pesquisa de aderência de práticas ágeis em startups de software (2 de 5)

Há um processo diário (ou com regularidade menor que semanalmente) de implantação contínua do trabalho desenvolvido por todos os desenvolvedores para homologação do trabalho desenvolvido.

desenvolvido.							
	1	2	3	4	5		
Não concordo totalmente						Concordo totalmente	
O fluxo/processo de tral possível propor melhori	as c	cont		as n	a fo		scos e gargalos de modo que seja ime.
Não concordo totalmente						Concordo totalmente	
É definido um certo pad individuais de programa	ção	e l		um	cóc		liferem diferentes estilos
Não concordo totalmente					0	Concordo totalmente	
As regras e a forma do f desenvolvimento.			e tra			ão conhecidas por t	odos os envolvidos no
Não concordo totalmente	_		_	_		Concordo totalmente	
Um desenvolvedor exec mesmo tempo perdendo	o o f	foco		nsta	ante		e não faça diversas coisas ao
Não concordo totalmente						Concordo totalmente	
	ado						ível de cobertura aceitável de sua pontuais como validações
	1	2	3	4	5		
Não concordo totalmente						Concordo totalmente	

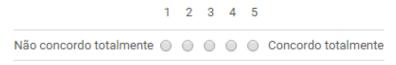
# Pesquisa de aderência de práticas ágeis em startups de software (3 de 5)

O desenvolvimento é planejado para ser executado em curtos espaços de tempo onde	
funcionalidades são entregues em partes menores que agreguem mais valor ao "cliente" ao long	JO
do tempo como em iterações/timebox.	

	3						
	1	2	3	4	5		
Não concordo totalmente						Concordo totalmente	
Experimentamos novas tornem mais produtivos	s e o	rga	niza	ados	5.	ar nosso trabalho ati	avés de novas práticas que nos
	1	2	3	4	5		
Não concordo totalmente						Concordo totalmente	
Temos testes a nível de na aplicação para cobri	r po		eis/	fall	nas		ndo as possíveis ações do usuário
Não concordo totalmente	0	0			0	Concordo totalmente	
como escala de fibonac	1	2	3	4	5		
Não concordo totalmente						Concordo totalmente	
	pro		o co	mo	dis		jue é praticada a remoção da logia em momento não apropriado
Não concordo totalmente						Concordo totalmente	
Temos um momento en trabalho feito.	i					edbacks sobre o pro	cesso e a evolução/qualidade do
	1	2	3	4	5		
Não concordo totalmente						Concordo totalmente	

### Pesquisa de aderência de práticas ágeis em startups de software (4 de 5)

A lista d	e coisas	a serem f	futuramente	desenvol	vidas é	discutida e	em mais	detalhes,	como
discuss	ão de sua	importâi	ncia para o r	negócio, p	ara que	sua prioriz	zação sej	a feita.	



São feitos testes com os interessados o quanto antes para que haja realimentação da informação de modo a propor melhorias constantes no produto desenvolvido.

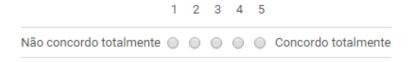
	1	2	3	4	5	
Não concordo totalmente	0	0				Concordo totalmente

Estimativas do trabalho a ser realizado não são feitas em medida de tempo, mas de acordo com sua complexidade e baseadas em outras tarefas já desenvolvidas.

1	1 :	2	3	4	5	
Não concordo totalmente		)				Concordo totalmente

Há um momento diário em que os integrantes do time rapidamente dizem o que realizaram no dia anterior, o que realizarão no dia atual e se há algum impedimento ao seu trabalho.

Há um quadro físico/digital onde o acompanhamento das tarefas pode ser feito por todos os envolvidos.

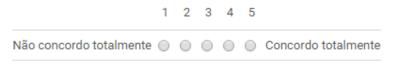


Existe uma lista de coisas a serem futuramente desenvolvidas pré-organizadas e de forma ordenada de acordo com sua importância para o negócio.

	1	2	3	4	5	
Não concordo totalmente						Concordo totalmente

## Pesquisa de aderência de práticas ágeis em startups de software (5 de 5)

C	) time	não	utiliza	nenhuma	métrica	para	estimativa	que	represente	grandezas	de cor	mplexid	lade
p	ara as	tare	efas el	encadas.									



É estimulada a auto gestão para que todos sintam-se independentes para efetuarem seu trabalho da forma que acham melhor.

	1	2	3	4	5	
Não concordo totalmente						Concordo totalmente

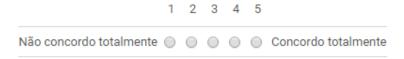
Não tentamos adivinhar o futuro, esperamos ter informações suficientes para que seja tomada a melhor decisão.

	1	2	3	4	5	
Não concordo totalmente	0					Concordo totalmente

Decisões são tomadas em conjunto, são pensadas multiplas opções e a melhor solução emerge através da escolha entre os prós e contras de cada opção.

	1	2	3	4	5	
Não concordo totalmente						Concordo totalmente

O time não tem a oportunidade de dar feedback sobre a forma de trabalho ou os ciclos de desenvolvimento.



Existe um senso de propósito nos envolvidos, as pessoas são engajadas com o trabalho que é feito e sabem de sua importância para o sucesso do projeto.

	1	2	3	4	5	
Não concordo totalmente						Concordo totalmente

qualidade do sistema (usuários testes, detentores da ideia).													
	1	2	3	4	5								
Não concordo totalmente		0		0	0	Concordo totalmente	е						
As tarefas futuras não s para o negócio.						priorização é realizad	ıda	sem	cons	iderar	a imp	ortân	cia
	1	2	3	4	5								
Não concordo totalmente						Concordo totalmente	е						

O time mantém-se pequeno e tem acesso direto as pessoas que julgarão a integridade e

# APÊNDICE B – QUESTÕES E SUAS RESPECTIVAS PRÁTICAS

Práticas	Questões
Seleção Voluntária de Tarefas	Independente do grau de senioridade dos integrantes do time os desenvolvedores podem escolher, dentre um conjunto de tarefas, quaisquer uma independente do seu grau de dificuldade para sua execução.
Integração Contínua	Há um processo diário (ou com regularidade menor que semanalmente) de integração contínua do trabalho desenvolvido por todos os desenvolvedores.
Implantação Contínua	Há um processo diário (ou com regularidade menor que semanalmente) de implantação contínua do trabalho desenvolvido por todos os desenvolvedores para homologação do trabalho desenvolvido.
Histórias de Usuário	Escrevemos pequenas histórias de usuários ou textos simples, sobre o que deve ser implementado com o intuito de serem âncoras para maiores discussões com os clientes.
Propriedade Coletiva	Todos os desenvolvedores sentem-se proprietário do que é desenvolvido, sendo responsáveis por sua qualidade tanto em modelagem quanto em qualidade.
Design Simples	São desenvolvidos apenas códigos que atendam ao requisito solicitado, sem adição de coisas que tentam prever o futuro.
Refatoração	Durante o desenvolvimento ao encontrarmos códigos duplicados ou incoerentes promovemos pequenas refatorações para mantê-lo íntegro e coeso.
Testes	Temos testes a nível de aplicação do tipo caixa preta ou seguindo as possíveis ações do usuário na aplicação para cobrir possíveis falhas do sistema.
Testes	Temos testes em unidades/componentes para garantir certo nível de cobertura aceitável de suas interfaces. (não é esperado cobertura completa, apenas testes pontuais como validações algorítmicas e negócio)

Padrões de Codificação	É definido um certo padrão de codificação para que não se proliferem diferentes estilos individuais de programação e haja um código íntegro e coeso.
Visão do fluxo de trabalho e riscos/Kanban Board	O fluxo/processo de trabalho permite identificar problemas, riscos e gargalos de modo que seja possível propor melhorias contínuas na forma de trabalho do time.
Limitação do trabalho em progresso	Um desenvolvedor executa apenas uma tarefa por vez para que não faça diversas coisas ao mesmo tempo perdendo o foco constantemente.
Tornar o processo explícito	As regras e a forma do fluxo de trabalho são conhecidas por todos os envolvidos no desenvolvimento.
Implementar ciclos de feedback	Temos um momento em que todos dão feedbacks sobre o processo e a evolução/qualidade do trabalho feito.
Implementar ciclos de feedback (antítese)	O time não tem a oportunidade de dar feedback sobre a forma de trabalho ou os ciclos de desenvolvimento.
Aumentar colaborativamente e evoluir experimentalmente	Experimentamos novas formas de executar nosso trabalho através de novas práticas que nos tornem mais produtivos e organizados.
Desenvolvimento Iterativo, Incremental e timeboxed	O desenvolvimento é planejado para ser executado em curtos espaços de tempo onde funcionalidades são entregues em partes menores que agreguem mais valor ao "cliente" ao longo do tempo como em iterações/timebox.
Três perguntas e Reunião Diária	Há um momento diário em que os integrantes do time rapidamente dizem o que realizaram no dia anterior, o que realizarão no dia atual e se há algum impedimento ao seu trabalho. (ex. reunião diária - SCRUM)
Taskboard	Há um quadro físico/digital onde o acompanhamento das tarefas pode ser feito por todos os envolvidos.
Estimativa Relativa	Estimativas do trabalho a ser realizado não são feitas em medida de tempo, mas de acordo com sua complexidade e baseadas em outras tarefas já desenvolvidas.
Planning Poker	Para que a estimativa não seja influenciada entre os integrantes do time é usada alguma métrica como escala de fibonacci ou uma ordem específica para representar grandezas.

Planning Poker (antítese)	O time não utiliza nenhuma métrica para estimativa que represente grandezas de complexidade para as tarefas elencadas.
Backlog	Existe uma lista de coisas a serem futuramente desenvolvidas pré-organizadas e de forma ordenada de acordo com sua importância para o negócio.
Tratamento de Backlog - Grooming	A lista de coisas a serem futuramente desenvolvidas é discutida em mais detalhes, como discussão de sua importância para o negócio, para que sua priorização seja feita.
Tratamento de Backlog - Grooming (antítese)	As tarefas futuras não são discutidas, e a priorização é realizada sem considerar a importância para o negócio.
Realimentação de Informação	São feitos testes com os interessados o quanto antes para que haja realimentação da informação de modo a propor melhorias constantes no produto desenvolvido.
Desenvolvimento em Conjunto	Decisões são tomadas em conjunto, são pensadas multiplas opções e a melhor solução emerge através da escolha entre os prós e contras de cada opção.
Opções de decisão e Decidir no último momento	Não tentamos adivinhar o futuro, esperamos ter informações suficientes para que seja tomada a melhor decisão.
Estimular o sistema	É estimulada a auto gestão para que todos sintam- se independentes para efetuarem seu trabalho da forma que acham melhor.
Motivação	Existe um senso de propósito nos envolvidos, as pessoas são engajadas com o trabalho que é feito e sabem de sua importância para o sucesso do projeto.
Integridade Conceitual e Percebida	O time mantém-se pequeno e tem acesso direto as pessoas que julgarão a integridade e qualidade do sistema (usuários testes, detentores da ideia).
Integridade Conceitual e Percebida (reforço)	A efetividade na comunicação é prezada por todos, de modo que é praticada a remoção da complexidade diante do projeto como discussões sobre tecnologia em momento não apropriado.

Quadro 11 - Relação entre práticas e suas questões para escala Likert

### **APÊNDICE C - RESPOSTAS DOS CANDIDATOS**

2 Aceito       De 10 a 49 pessoas         3 Aceito       De 50 a 99 pessoas         4 Aceito       Até 9 pessoas         5 Aceito       De 10 a 49 pessoas         6 Aceito       Até 9 pessoas         7 Aceito       Até 9 pessoas         8 Aceito       De 10 a 49 pessoas         9 Aceito       Até 9 pessoas         10 Aceito       Até 9 pessoas         11 Não aceito         12 Aceito       Até 9 pessoas         13 Aceito       De 10 a 49 pessoas         14 Aceito       Até 9 pessoas         15 Aceito       Até 9 pessoas         16 Aceito       Até 9 pessoas         17 Aceito       De 10 a 49 pessoas         18 Aceito       Até 9 pessoas         20 Aceito       Até 9 pessoas         21 Aceito       De 10 a 49 pessoas         22 Aceito       Até 9 pessoas         23 Aceito       Até 9 pessoas         24 Aceito       De 10 a 49 pessoas         25 Aceito       Até 9 pessoas         26 Aceito       Até 9 pessoas				
2 Aceito       De 10 a 49 pessoas         3 Aceito       De 50 a 99 pessoas         4 Aceito       Até 9 pessoas         5 Aceito       De 10 a 49 pessoas         6 Aceito       Até 9 pessoas         7 Aceito       Até 9 pessoas         8 Aceito       De 10 a 49 pessoas         9 Aceito       Até 9 pessoas         10 Aceito       Até 9 pessoas         11 Não aceito       Até 9 pessoas         12 Aceito       Até 9 pessoas         13 Aceito       De 10 a 49 pessoas         14 Aceito       Até 9 pessoas         15 Aceito       Até 9 pessoas         16 Aceito       Até 9 pessoas         17 Aceito       De 10 a 49 pessoas         18 Aceito       Até 9 pessoas         20 Aceito       Até 9 pessoas         21 Aceito       De 10 a 49 pessoas         22 Aceito       Até 9 pessoas         23 Aceito       Até 9 pessoas         24 Aceito       De 10 a 49 pessoas         25 Aceito       Até 9 pessoas         26 Aceito       Até 9 pessoas	ID		ou trabalhavam na startup que está utilizando como parâmetro para responder a esta	Idade
3 Aceito         De 50 a 99 pessoas           4 Aceito         Até 9 pessoas           5 Aceito         De 10 a 49 pessoas           6 Aceito         Até 9 pessoas           7 Aceito         Até 9 pessoas           8 Aceito         De 10 a 49 pessoas           9 Aceito         Até 9 pessoas           10 Aceito         Até 9 pessoas           11 Não aceito         Até 9 pessoas           12 Aceito         Até 9 pessoas           13 Aceito         De 10 a 49 pessoas           14 Aceito         Até 9 pessoas           15 Aceito         Até 9 pessoas           16 Aceito         Até 9 pessoas           17 Aceito         De 10 a 49 pessoas           18 Aceito         Até 9 pessoas           20 Aceito         Até 9 pessoas           21 Aceito         De 10 a 49 pessoas           22 Aceito         Até 9 pessoas           23 Aceito         Até 9 pessoas           24 Aceito         De 10 a 49 pessoas           25 Aceito         Até 9 pessoas           26 Aceito         Até 9 pessoas	1	Aceito	De 10 a 49 pessoas	29
4 Aceito Até 9 pessoas 5 Aceito De 10 a 49 pessoas 6 Aceito Até 9 pessoas 7 Aceito Até 9 pessoas 8 Aceito De 10 a 49 pessoas 9 Aceito Até 9 pessoas 10 Aceito Até 9 pessoas 11 Não aceito 12 Aceito Até 9 pessoas 13 Aceito De 10 a 49 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito Até 9 pessoas 18 Aceito Até 9 pessoas 19 Aceito Até 9 pessoas 10 Aceito Até 9 pessoas 11 Aceito Até 9 pessoas 12 Aceito Até 9 pessoas 13 Aceito Até 9 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito De 10 a 49 pessoas 18 Aceito Até 9 pessoas 19 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito De 10 a 49 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	2	Aceito	De 10 a 49 pessoas	28
5 Aceito De 10 a 49 pessoas 6 Aceito Até 9 pessoas 7 Aceito Até 9 pessoas 8 Aceito De 10 a 49 pessoas 9 Aceito Até 9 pessoas 10 Aceito Até 9 pessoas 11 Não aceito 12 Aceito Até 9 pessoas 13 Aceito De 10 a 49 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito Até 9 pessoas 18 Aceito Até 9 pessoas 19 Aceito Até 9 pessoas 10 Aceito Até 9 pessoas 11 Aceito Até 9 pessoas 12 Aceito Até 9 pessoas 13 Aceito Até 9 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito De 10 a 49 pessoas 18 Aceito Até 9 pessoas 19 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito De 10 a 49 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	3	Aceito	De 50 a 99 pessoas	33
Aceito Até 9 pessoas  Aceito De 10 a 49 pessoas  Aceito Até 9 pessoas  Aceito De 10 a 49 pessoas  Aceito Até 9 pessoas	4	Aceito	Até 9 pessoas	27
7 Aceito Até 9 pessoas 8 Aceito De 10 a 49 pessoas 9 Aceito Até 9 pessoas 10 Aceito Até 9 pessoas 11 Não aceito 12 Aceito Até 9 pessoas 13 Aceito De 10 a 49 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito Até 9 pessoas 18 Aceito De 10 a 49 pessoas 19 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito Até 9 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito Até 9 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas 27 Aceito Até 9 pessoas 28 Aceito Até 9 pessoas 29 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito Até 9 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	5	Aceito	De 10 a 49 pessoas	26
8 Aceito De 10 a 49 pessoas 9 Aceito Até 9 pessoas 10 Aceito Até 9 pessoas 11 Não aceito 12 Aceito Até 9 pessoas 13 Aceito De 10 a 49 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito De 10 a 49 pessoas 18 Aceito De 10 a 49 pessoas 19 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito Até 9 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito Até 9 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas 27 Aceito Até 9 pessoas 28 Aceito Até 9 pessoas 29 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito Até 9 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito Até 9 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	6	Aceito	Até 9 pessoas	32
9 Aceito Até 9 pessoas  10 Aceito Até 9 pessoas  11 Não aceito  12 Aceito Até 9 pessoas  13 Aceito De 10 a 49 pessoas  14 Aceito Até 9 pessoas  15 Aceito Até 9 pessoas  16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas	7	Aceito	Até 9 pessoas	27
10 Aceito Até 9 pessoas  11 Não aceito  12 Aceito Até 9 pessoas  13 Aceito De 10 a 49 pessoas  14 Aceito Até 9 pessoas  15 Aceito Até 9 pessoas  16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas	8	Aceito	De 10 a 49 pessoas	27
11 Não aceito 12 Aceito Até 9 pessoas 13 Aceito De 10 a 49 pessoas 14 Aceito Até 9 pessoas 15 Aceito Até 9 pessoas 16 Aceito Até 9 pessoas 17 Aceito De 10 a 49 pessoas 18 Aceito Até 9 pessoas 19 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito De 10 a 49 pessoas 22 Aceito De 10 a 49 pessoas 23 Aceito Até 9 pessoas 24 Aceito Até 9 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas 27 Aceito Até 9 pessoas 28 Aceito Até 9 pessoas 29 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito Até 9 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas	9	Aceito	Até 9 pessoas	20
12 Aceito Até 9 pessoas  13 Aceito De 10 a 49 pessoas  14 Aceito Até 9 pessoas  15 Aceito Até 9 pessoas  16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito De 10 a 49 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas	10	Aceito	Até 9 pessoas	37
13 Aceito De 10 a 49 pessoas  14 Aceito Até 9 pessoas  15 Aceito Até 9 pessoas  16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito De 10 a 49 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito Até 9 pessoas  25 Aceito Até 9 pessoas	11	Não aceito		
Até 9 pessoas  15 Aceito Até 9 pessoas  16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito De 10 a 49 pessoas  23 Aceito Até 9 pessoas  24 Aceito De 10 a 49 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas  22 Aceito Até 9 pessoas	12	Aceito	Até 9 pessoas	21
15 Aceito Até 9 pessoas  16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito De 10 a 49 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas	13	Aceito	De 10 a 49 pessoas	36
16 Aceito Até 9 pessoas  17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito De 10 a 49 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas	14	Aceito	Até 9 pessoas	34
17 Aceito De 10 a 49 pessoas  18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito De 10 a 49 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas  27 Aceito Até 9 pessoas  28 Aceito Até 9 pessoas  29 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito Até 9 pessoas	15	Aceito	Até 9 pessoas	26
18 Aceito Até 9 pessoas  19 Aceito Até 9 pessoas  20 Aceito Até 9 pessoas  21 Aceito De 10 a 49 pessoas  22 Aceito Até 9 pessoas  23 Aceito Até 9 pessoas  24 Aceito De 10 a 49 pessoas  25 Aceito Até 9 pessoas  26 Aceito Até 9 pessoas	16	Aceito	Até 9 pessoas	20
19 Aceito Até 9 pessoas 20 Aceito Até 9 pessoas 21 Aceito De 10 a 49 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas 27 Aceito Até 9 pessoas 28 Aceito Até 9 pessoas 29 Aceito Até 9 pessoas	17	Aceito	De 10 a 49 pessoas	29
20 Aceito Até 9 pessoas 21 Aceito De 10 a 49 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas 27 Aceito Até 9 pessoas 28 Aceito Até 9 pessoas	18	Aceito	Até 9 pessoas	31
21 Aceito De 10 a 49 pessoas 22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	19	Aceito	Até 9 pessoas	34
22 Aceito Até 9 pessoas 23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	20	Aceito	Até 9 pessoas	29
23 Aceito Até 9 pessoas 24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	21	Aceito	De 10 a 49 pessoas	21
24 Aceito De 10 a 49 pessoas 25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	22	Aceito	Até 9 pessoas	21
25 Aceito Até 9 pessoas 26 Aceito Até 9 pessoas	23	Aceito	Até 9 pessoas	21
26 Aceito Até 9 pessoas	24	Aceito	De 10 a 49 pessoas	24
	25	Aceito	Até 9 pessoas	22
27 Aceito De 10 a 49 pessoas			•	25
·			De 10 a 49 pessoas	23 27

ID	Qual é o seu tempo de experiência em sua função?	Seleção Voluntária de Tarefas	Integração Contínua	Histórias de Usuário
1	4 e 10 anos	4	5	4
2	4 e 10 anos	5	5	5
3	4 e 10 anos	1	5	5
4	Entre 1 e 3 anos	5	3	2
5	4 e 10 anos	5	5	5
6	4 e 10 anos	3	1	1
7	4 e 10 anos	5	4	3
8	4 e 10 anos	5	5	3
9	4 e 10 anos	5	1	4
10	Entre 1 e 3 anos	2	5	5
11				
12	4 e 10 anos	2	5	4
13	Mais de 10 anos	4	3	4
14	Menos de 1 ano	4	3	2
15	Menos de 1 ano	2	1	1
16	Entre 1 e 3 anos	3	1	1
	Entre 1 e 3 anos	2	3	2
18	Entre 1 e 3 anos	1	2	3
19	4 e 10 anos	3	3	3
20	Entre 1 e 3 anos	3	3	5
21	Entre 1 e 3 anos	4	4	1
	Entre 1 e 3 anos	3	4	5
23	Entre 1 e 3 anos	2	5	5
24	Entre 1 e 3 anos	4	2	1
	Menos de 1 ano	3	1	1
	Entre 1 e 3 anos	3	3	4
27		3	4	2
28	Entre 1 e 3 anos	4	5	3

ID	Propriedade Coletiva	Design Simples	Refatoração	Testes - Código	Implantação Contínua
1	3	1	2	3	4
2	5	1	4	4	4
3	5	4	4	5	5
4	5	4	4	2	2
5	5	5	5	5	5
6	2	4	3	2	2
7	2	5	5	3	2
8	5	3	4	2	5
9	4	3	5	1	2
10	3	5	4	4	5
11					
12	5	1	5	5	5
13	3	2	4	4	4
14	5	3	5	3	3
15	2	1	3	3	2
16	3	3	4	1	2
17	3	2	3	2	3
18	4	2	2	2	4
19	3	3	3	4	5
20	5	4	5	5	5
21	4	5	3	3	1
22	5	4	5	4	3
23	5	3	5	5	5
24	4	3	3	5	3
25	1	2	1	1	1
26	4	3	5	2	4
27	4	5	4	4	2 5
28	5	3	4	5	5

ID	Padrões de Codificação	Visão do Fluxo de Trabalho e Riscos/Kanban Board	Limitação do Trabalho em Progresso	Tornar o Processo Explícito
1	4	3	4	4
2	4	4	3	5
3	5	5	5	4
4	2	1	2	2
5	5	5	5	5
6	3	2	1	1
7	4	4	3	4
8	3	5	5	5
9	3	5	4	4
10	4	3	4	5
11				
12	5	5	5	5
13	5	4	4	4
14	5	4	3	4
15	4	3	1	3
16	5	2	2	5
17	2	5	3	5
18	3	2	2	3
19	4	5	4	4
20	5	5	4	5
21	4	4	1	1
22	3	4	1	4
23	5	5	4	5
24	1	3	2	3
25	1	1	2	2
26	5	4	4	5 2
27	2	4	3	2
28	3	3	4	5

1       4       2       3         2       1       2       3         3       5       4       3         4       1       1       5         5       4       5       3         6       2       2       2         7       1       2       3         8       5       4       5         9       3       3       4         10       3       5       3         11       1       1       1         12       5       5       4         13       3       5       4         13       3       5       4         13       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       4         20       3       5       4         21       2       3       3         22       1       5       5	ID	Testes - Aplicação	Implementar Ciclos de Feedback	Aumentar Colaborativamente e Evoluir Experimentalmente	Desenvolvimento Iterativo, Incremental e Timeboxed
3       5       4       3         4       1       1       5         5       4       5       3         6       2       2       2         7       1       2       3         8       5       4       5         9       3       3       4         10       3       5       3         11       12       5       5         13       3       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         4       4       4       4         19       4       4       4         20       3       5       4         21       2       3       3         22       1       5       5         23       2       1       5         25       1       1       2 <t< td=""><td>1</td><td>4</td><td>2</td><td>3</td><td>4</td></t<>	1	4	2	3	4
4       1       1       5         5       4       5       3         6       2       2       2         7       1       2       3         8       5       4       5         9       3       3       4         10       3       5       3         11       12       5       5         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         4       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       2       1       5         24       4       2       5         25       1       1       2         26       4       4       4 <td>2</td> <td>1</td> <td>2</td> <td>3</td> <td>3</td>	2	1	2	3	3
5       4       5       3         6       2       2       2         7       1       2       3         8       5       4       5         9       3       3       4         10       3       5       3         11       1       1       1         12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       4         20       3       5       4         21       2       3       3         22       1       5       5         23       2       1       5         25       1       1       2         26       4       4       4	3	5	4	3	5
6       2       2       2         7       1       2       3         8       5       4       5         9       3       3       4         10       3       5       3         11       11       12       5       4         13       3       5       4       4         14       3       5       4       4         15       1       3       3       3         16       5       2       2       2         17       2       5       5       5         18       4       4       4       4         19       4       4       4       5         20       3       5       4       4         21       2       3       3       3         22       1       5       5       5         23       2       1       1       2         25       1       1       1       2         26       4       4       4       4	4	1	1	5	5
7       1       2       3         8       5       4       5         9       3       3       4         10       3       5       3         11            12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       4         20       3       5       4         21       2       3       3         22       1       5       5         23       2       1       5         25       1       1       2         26       4       4       4	5	4	5	3	4
8       5       4       5         9       3       3       4         10       3       5       3         11       11       12       5       4         12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       4         20       3       5       4         21       2       3       3         21       2       3       3         22       1       5       5         23       2       5       5         24       4       2       5         25       1       1       2         26       4       4       4	6	2	2	2	2
9       3       3       4         10       3       5       3         11            12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       4         20       3       5       4         21       2       3       3         22       1       5       5         23       2       5       5         25       1       1       2         26       4       4       4	7	1	2	3	4
10       3       5       3         11       12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       2       5       5         25       1       1       2         26       4       4       4	8	5	4	5	3
11         12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       2       5       5         24       4       2       5         25       1       1       2         26       4       4       4	9	3	3	4	5
12       5       5       4         13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       2       5       5         24       4       2       5         25       1       1       2         26       4       4       4	10	3	5	3	4
13       3       5       4         14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       2       5       5         24       4       2       5         25       1       1       2         26       4       4       4	11				
14       3       5       4         15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       2       5       5         25       1       1       2         26       4       4       4	12	5	5	4	5
15       1       3       3         16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       5       5         24       4       2       5         25       1       1       2         26       4       4       4	13	3	5	4	5
16       5       2       2         17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       5       5         24       4       2       5         25       1       1       2         26       4       4       4	14	3	5	4	2
17       2       5       5         18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       5       5         24       4       2       5         25       1       1       2         26       4       4       4	15	1	3	3	3
18       4       4       4         19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       5       5         24       4       2       5         25       1       1       2         26       4       4       4	16	5	2	2	1
19       4       4       5         20       3       5       4         21       2       3       3         22       1       5       5         23       5       5         24       4       2       5         25       1       1       2         26       4       4       4	17	2	5	5	4
20       3       5       4         21       2       3       3         22       1       5       5         23       5       5         24       4       2       5         25       1       1       2         26       4       4       4	18	4	4	4	4
21       2       3       3         22       1       5       5         23       24       4       2       5         25       1       1       2         26       4       4       4	19	4	4	5	4
22     1     5     5       23      5       24     4     2     5       25     1     1     2       26     4     4     4	20	3	5	4	4
23       24     4     2     5       25     1     1     2       26     4     4     4	21	2	3	3	1
24     4     2     5       25     1     1     2       26     4     4     4	22	1	5	5	5
25 1 1 2 26 4 4 4	23				
26 4 4 4	24	4	2	5	4
	25	1	1	2	3
27 / / 5					3
28 4 5 4	27		4	5	5

1 2 3	3 1 4	1 3	1	5	-
3		3		U	2
	4		4	5	4
4		5	5	5	4
4	2	5	2	3	3
5	4	3	3	4	
6	3	1	2	2	2
7	2	1	1	4	2
8	5	1	1	5	3
9	5	3	2	4	4
10	3	5	5	5	5
11					
12	2		5	5	5
13	5	3	5	5	4
14	4	1	3	4	3
15	3	1	2	2	4
16	4	1	3	1	5
17	4	4	4	5	3
18	4	2	3	5	4
19	5	4	4	4	5
20	5	3	4	5	3
21	5	4	3	5	4
22	4	1	5	5	5
23					
24	2	1	1	1	3
25	1	1	1	4	1
26	2	4	3	5	4
27 28	4	3	3 4	5 5	2

1       1       2       4       3         2       4       5       3       1         3       5       5       3       2         4       1       1       1       1         5       3       4       5       5       5         6       1       2       2       2       3         7       2       2       2       3       3         8       3       3       3       1       1         9       5       5       1       1       1       1         10       4       4       3       1       1       1         11       12       5       5       3       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 </th <th>ID</th> <th>Backlog</th> <th>Tratamento de Backlog - Grooming</th> <th>Realimentação de Informação</th> <th>Implementar Ciclos de Feedback (antítese)</th> <th>Planning Poker (antítese)</th>	ID	Backlog	Tratamento de Backlog - Grooming	Realimentação de Informação	Implementar Ciclos de Feedback (antítese)	Planning Poker (antítese)
3       5       5       3       2         4       1       1       1       1         5       3       4       5       5         6       1       2       2       3         7       2       2       2       3         8       3       3       3       1         9       5       5       1       1         10       4       4       3       1         11       12       5       5       3       1         11       12       5       5       3       1         11       12       5       3       1       1         12       5       5       3       1       1       1         14       5       5       3       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1	1	1	2	4	3	5
4       1       1       1       1         5       3       4       5       5         6       1       2       2       3         7       2       2       2       3         8       3       3       3       1         9       5       5       1       1         10       4       4       3       1         11       12       5       5       3       1         11       12       5       5       3       1         11       12       5       5       3       1         11       12       5       3       1       1         14       5       5       3       1       1         15       4       3       3       3       3       3         16       4       5       2       2       2         17       4       5       4       1       1         18       4       5       2       2       2         20       5       5       5       5       5         21       5       1	2	4	5	3	1	4
5       3       4       5       5         6       1       2       2       3         7       2       2       2       3         8       3       3       3       1         9       5       5       1       1         10       4       4       3       1         11       12       5       5       3       1         11       12       5       5       3       1         14       5       5       5       1       1         15       4       3       3       3       3         16       4       5       2       2       2         17       4       5       4       1       1         18       4       5       2       2       2         20       5       5       5       1       1         21       5       1       2       3       3       2         22       3       3       5       5       5       3       1         22       3       3       5       5       5       5       <	3	5	5	3	2	2
6       1       2       2       3         7       2       2       2       3         8       3       3       1         9       5       5       1       1         10       4       4       3       1         11       11       11       11       11         12       5       5       5       3         13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       3       3       4       4         25       3       1       1       5	4	1	1	1	1	1
7       2       2       2       3         8       3       3       1         9       5       5       1       1         10       4       4       3       1         11       11       11       11       11       11         12       5       5       5       3       3       1         14       5       5       5       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1 <td>5</td> <td>3</td> <td>4</td> <td>5</td> <td>5</td> <td>5</td>	5	3	4	5	5	5
8       3       3       1         9       5       5       1       1         10       4       4       3       1         11       11       11       1       1         12       5       5       5       3         13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       3       4       4         25       3       1       1       5	6	1	2	2	3	4
9       5       5       1       1         10       4       4       3       1         11       11       11       11         12       5       5       5       3         13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       3       5       5       5         23       24       3       3       4       4         25       3       1       1       5	7	2	2	2	3	3
10       4       4       3       1         11       12       5       5       3         13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       22       3       4       4         25       3       1       1       5	8	3	3	3	1	1
11         12       5       5       3         13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       3       4       4         25       3       1       1       5	9	5	5	1	1	4
12       5       5       3         13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       3       4       4         25       3       1       1       5	10	4	4	3	1	1
13       5       5       5       1         14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       3       4       4         25       3       1       1       5	11					
14       5       5       3       1         15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       3       4       4         25       3       1       1       5	12	5		5	3	5
15       4       3       3       3         16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       5       3       4       4         25       3       1       1       5	13	5	5	5	1	1
16       4       5       2       2         17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       2       4       4       4         25       3       1       1       5	14	5	5	3	1	1
17       4       5       4       1         18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       24       3       3       4       4         25       3       1       1       5	15	4	3	3	3	3
18       4       5       4       2         19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       24       3       3       4       4         25       3       1       1       5	16	4	5	2	2	5
19       4       4       5       2         20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       24       3       3       4       4         25       3       1       1       5	17	4	5	4	1	1
20       5       5       5       1         21       5       1       2       3         22       3       3       5       5         23       24       3       3       4       4         25       3       1       1       5	18	4	5	4	2	3
21     5     1     2     3       22     3     3     5     5       23       24     3     3     4     4       25     3     1     1     5	19	4	4	5	2	1
22       3       3       5       5         23         4       4         24       3       3       4       4         25       3       1       1       5	20	5	5	5	1	2
23 24 3 3 4 4 25 3 1 1 5	21	5	1	2	3	1
24     3     3     4     4       25     3     1     1     5	22	3	3	5	5	4
25 3 1 1 5	23					
	24	3	3	4	4	4
26 5 4 4 4		3	1	1	5	5
						5
27     5     5     4     1       28     1     3     3     1						4

ID	Tratamento de Backlog - Grooming (antítese)	Desenvolvimento em Conjunto	Opções de Decisão e Decidir no Último Momento	Estimular o Sistema
1	4	3	3	4
2	1	2	3	4
3	1	5	2	3
4	1	5	4	5
5	4	5	5	5
6	1	2	2	2
7	2	2	4	4
8	1	5	4	5
9	1	3	1	5
10	1	4	5	3
11				
12	4	5		3
13	1	4	2	2
14	1	5	4	4
15	4	2	3	4
16	2	2	3	2
17	1	5	3	3
18	4	4	3	4
19	2	4		3
20	1	5	4	5
21	3	3	1	5
22	1	2	1	4
23				
24	1	4	3	5
25	5	1	2	2
26	4		5	3
27	1	3	4	5
28	1	4	3	5

ID	Motivação	Integridade Conceitual e Percebida
1	3	5
2	4	4
3	5	5
4	2	1
5	5	4
6	3	4
7	2	4
8	4	5
9	4	4
10	3	4
11		
12	3	5
13	4	5
14	4	3
15	3	3
16	4	3
17	5	4
18	4	4
19	4	3
20	5	5
21	4	2
22	3	4
23		
24	5	5
25	1	2
26 27	4 5	4 5
28	5	5
	•	•

### APÊNDICE D – ANÁLISE GRÁFICA DAS RESPOSTAS



