



Universidade Federal do Rio de Janeiro

Rafael Macacchero Lago de Sá Rodrigues

**ANÁLISE DAS PRINCIPAIS FORMAS DE
CONTRATAÇÃO DE DESENVOLVIMENTO
DE SOFTWARE COM MÉTODOS ÁGEIS**

DISSERTAÇÃO DE MESTRADO

Rio de Janeiro
2015



Instituto de Matemática



Instituto Tércio Pacitti de Aplicações
e Pesquisas Computacionais

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Rafael Macacchero Lago de Sá Rodrigues

ANÁLISE DAS PRINCIPAIS FORMAS DE CONTRATAÇÃO DE DESENVOLVIMENTO DE SOFTWARE COM MÉTODOS ÁGEIS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Orientador: Prof. Rodrigo Penteado Ribeiro de Toledo, Ph.D.

Rio de Janeiro
2015

Rafael Macacchero Lago de Sá Rodrigues

ANÁLISE DAS PRINCIPAIS FORMAS DE CONTRATAÇÃO DE DESENVOLVIMENTO DE SOFTWARE COM MÉTODOS ÁGEIS

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informática,
Instituto de Matemática e Instituto Tércio Pacciti,
Universidade Federal do Rio de Janeiro, como
requisito parcial à obtenção do título de Mestre
em Informática.

Aprovada em

Prof. Rodrigo Penteado Ribeiro de Toledo, Ph.D. PPGI/UFRJ

Prof. Geraldo Bonorino Xexéo, DSc, COPPE/UFRJ

Prof. Adriana Santarosa Vivacqua, DSc, PPGI/UFRJ

Mariano Angel Montoni, DSc

Aos meus pais, pelo apoio e confiança incondicionais.

Agradecimentos

A meus pais Sérgio e Rosangela, e a meu irmão Pedro, por todo o incentivo, pelo apoio moral, financeiro, e psicológico, pela confiança em meu trabalho, pela criação que me foi dada e pela liberdade de escolhas profissionais e acadêmicas, e pelo contínuo incentivo à excelência acadêmica e aos estudos.

A meus avós, Helena, Edison, Lúcia e Walcyr, pela confiança, apoio e incentivo a perseverar nos estudos e na minha área de interesse. Vocês me inspiram a continuar sempre estudando e buscar ser o melhor possível na minha carreira.

À minha namorada, Natália, pela compreensão e motivação constante. Foram dezenas de horas privada de companhia, mas em nenhum momento você reclamou. Obrigado por toda a força e dicas que você me deu durante todos esses meses de estudo. Você, com certeza, colaborou muito para o resultado desse trabalho.

A meu orientador Rodrigo de Toledo, que contribuiu incontáveis vezes com sugestões e ideias altamente relevantes, que agregaram muito valor a esse trabalho. Obrigado por confiar em mim e na nossa pesquisa.

Aos colegas Wesley e Avelino, que contribuíram significativamente para meu conhecimento na área de estudo, através de sugestões, artigos em parceria, aulas e trocas de email muito valiosas.

Aos colegas Pedro, Fernando, Mariano, Analia e David, que ofereceram muito apoio, sempre estiveram dispostos a contribuir e deram liberdade para que eu estudasse da forma que eu julgasse melhor. Obrigado pela confiança!

Por fim, mas não menos importante, a Deus, pela força, saúde, motivação e sabedoria que me fizeram perseverar no caminho acadêmico.

“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.” (Arthur Schopenhauer)

Resumo

RODRIGUES, Rafael Macacchero Lago de Sá. **ANÁLISE DAS PRINCIPAIS FORMAS DE CONTRATAÇÃO DE DESENVOLVIMENTO DE SOFTWARE COM MÉTODOS ÁGEIS**. 2015. 92 folhas. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

Em um ambiente corporativo, onde empresas de desenvolvimento de software atuais vêm modernizando suas metodologias de desenvolvimento com a adoção de métodos ágeis, é comum identificar que os contratos assinados dos projetos ainda são heranças das metodologias tradicionais. Empresas evitam contratar de forma diferente, por desconhecimento, medo do novo e comodidade, e não percebem que o contrato está muito atrelado à forma de trabalho executada. No entanto, esses contratos tradicionais são incompatíveis com as práticas ágeis e essa mistura gera prejuízos, tanto para o cliente quanto para o fornecedor. Este trabalho busca identificar novos modelos de contratos mais adequados à utilização com métodos ágeis, e também tem o objetivo de identificar quais são as características mais importantes presentes em contratos de desenvolvimento de software. Com modelos de contratos identificados e características comuns identificadas, é possível realizar um cruzamento para ranquear quais os modelos de contratos mais aderentes às características de maior importância em contratos de desenvolvimento de software, e, por fim, concluir se existem contratos adequados aos métodos ágeis e quais são os modelos de contrato mais recomendados.

Palavras-chave: métodos ágeis, agile, scrum, contratos, contratos ágeis

Abstract

RODRIGUES, Rafael Macacchero Lago de Sá. **ANÁLISE DAS PRINCIPAIS FORMAS DE CONTRATAÇÃO DE DESENVOLVIMENTO DE SOFTWARE COM MÉTODOS ÁGEIS**. 2015. 92 folhas. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

In a corporate environment where current software development companies have been modernizing their development methodologies with the adoption of agile methodologies, it is common to identify that contracts signed by projects are still legacies of traditional methodologies. Companies avoid hiring differently because of ignorance, fear of change and convenience, and they do not realize that the contract is strongly linked to how work is actually performed. However, these traditional contracts are incompatible with agile practices and this mixture generates losses for both client and supplier. This work seeks to identify new contract models that are more suitable to use with Agile Methodologies, and also aims to identify which are the most important features present in contracts. With models of contracts described and key characteristics identified, it is possible to perform a comparison to rank which models are more adherent to the most important characteristics identified in software development contracts, and, in the end, conclude if there are contracts that are adequate to agile projects, and which models of contracting are the most recommended.

Keywords: contracts, agile contracts, agile methodology, scrum, agile

Lista de Figuras

Figura 1: Percentual de sucesso em projetos pequenos e grandes	17
Figura 2: Diagramas de Planejamento do Protocolo de Revisão Sistemática e Execução da Revisão Sistemática	23
Figura 3: Tabela inicial de classificação do questionário.....	31
Figura 4: Tabela usada pelo Dot Voting.....	32
Figura 5: Exemplo de Modelo Cascata	35
Figura 6: Exemplo de iteração no <i>Scrum</i>	36
Figura 7: Quadro <i>Kanban</i> (AUDY, 2012)	39
Figura 8: Acúmulo de Valor Agregado por Entrega x Tempo & Custo por Entrega do <i>Money</i> <i>For Nothing, Change for Free</i>	48
Figura 9: Média de uso de funcionalidade de sistemas (STANDISH GROUP, 2002)	56
Figura 10: Gráfico sobre limitações/restrições contratuais	63
Figura 11: Gráfico total de relevância das características.....	64
Figura 12: Gráfico da relevância das características para o setor público	65
Figura 13: Gráfico da relevância das características para o setor privado	66
Figura 14: Gráfico da relevância das características para o lado Cliente.....	67
Figura 15: Gráfico da relevância das características para o lado Fornecedor	68
Figura 16: Consolidado de todos os gráficos.....	69
Figura 17: Matriz Comparativa entre Contratos	72
Figura 18: Matriz de Cálculo Ponderado entre Modelos de Contrato	85
Figura 19: Resultado do Cálculo Ponderado entre Contratos.....	86

Lista de Tabelas

Tabela 1: Protocolo de Revisão Sistemática	26
Tabela 2: Resumo dos resultados da Revisão Sistemática de Literatura	29

Lista de Siglas*

TI	Tecnologia da Informação
BDUF	<i>Big Design Up Front</i>
XP	<i>Extreme Programming</i>
RSL	Revisão Sistemática de Literatura
ACM	Association for Computing Machinery
RUP	<i>Rational Unified Process</i>
TPS	<i>Toyota Production System</i>
FDD	<i>Feature Driven Development</i>
ROI	<i>Return On Investment</i>
T&M	Tempo & Materiais
APF	Análise de Pontos de Função
PF	Pontos de Função
PCU	Pontos de Casos de Uso

**Em ordem de aparição no texto*

Sumário

1 Introdução.....	14
1.1 Cenário e Escopo	14
1.2 Descrição do Problema	15
1.3 Estrutura do Trabalho	18
2 Metodologia de Pesquisa	20
2.1 Revisão Sistemática de Literatura	20
2.1.1 Protocolo de Revisão Sistemática - Conceito.....	21
2.1.2 Protocolo de Revisão Sistemática - Execução.....	23
2.1.3 Execução da Busca	25
2.1.4 Extração de Dados das Publicações Seleccionadas	28
2.2 Questionário de Pesquisa.....	30
2.2.1 Preparação do Questionário	31
2.2.2 Validação e Correções no Questionário.....	31
2.2.3 Dot Voting	32
2.2.4 Identificação dos Respondentes e Divulgação.....	33
2.2.5 Coleta e Análise das Respostas	33
3 Referencial Teórico	34
3.1 Metodologias Ágeis.....	34
3.1.1 Histórico da Engenharia de Software e Metodologias Ágeis.....	34
3.1.2 Características das Metodologias Ágeis.....	37
3.2 Contratação de Desenvolvimento de Software Tradicional e Ágil	39
4 Identificação de Modelos de Contratos	41
4.1 Modelos de Contratos Tradicionais	41
4.1.1 Contrato Tradicional de Preço Fixo	41
4.1.2 Contrato de Tempo e Material (<i>Time & Materials</i>) com Escopo Fixo	42
4.1.3 Contrato de Lucro Fixo	43
4.1.4 Contrato com Cláusula de Bônus e Penalidade	43
4.2 Modelos de Contratos Ágeis	44
4.2.1 Contrato Ágil de Preço Fixo	44
4.2.2 Contrato de Tempo e Material (<i>Time & Materials</i>) com Escopo Flexível.....	45
4.2.3 Contrato com Preço Fixo por Unidades de Trabalho	46
4.2.4 Contrato “Money for Nothing, Change for Free”	47

4.2.5 Contrato Progressivo.....	48
4.2.6 Contrato “No Cure, No Pay”	49
4.2.7 Contrato com Objetivo de Custo.....	50
4.2.8 Outros Contratos.....	51
5 Características de Contratos de Desenvolvimento de Software	53
5.1 Precificação	53
5.2 Vigência do Contrato.....	54
5.3 Mudança de Escopo	55
5.4 Garantia.....	56
5.5 Envolvimento do Cliente	57
5.6 Método de Estimativa	57
5.7 Risco para o Cliente.....	58
5.8 Risco para o Fornecedor.....	59
5.9 Popularidade	59
5.10 Vantagem para Usuário Final	60
6 Pesquisa sobre as Características mais Importantes.....	62
6.1 Resultados da Pesquisa	62
6.1.1 Você já Teve seu Trabalho Limitado/Prejudicado por Restrições Contratuais?	63
6.1.2 Relevância Total das Características	64
6.1.3 Relevância das Características para o Setor Público	65
6.1.4 Relevância das Características para o Setor Privado	66
6.1.5 Relevância das características para o lado Cliente	67
6.1.6 Relevância das características para o lado Fornecedor	68
6.2 Análise do Resultado	68
7 Análise dos Modelos de Contratos Frente aos Resultados da Pesquisa.....	71
7.1 Análise da Matriz Comparativa	71
7.2 Cálculo Ponderado dos Modelos de Contrato em Relação à Pesquisa	83
8 Conclusão	87
8.1 Análise dos Objetivos e Condução do Trabalho	87
8.2 Principais Contribuições	88
8.3 Limitações do Trabalho	89
8.4 Trabalhos Futuros	90
9 Referências.....	91

1 Introdução

1.1 Cenário e Escopo

A área de Tecnologia de Informação (TI) teve grande crescimento nas últimas décadas, tornando indiscutível a sua necessidade e utilidade. Em face à necessidade de informatizar áreas inteiras de empresas, desenvolver aplicações e criar websites, empresas veem na terceirização da área de TI a melhor opção a ser tomada, visto que o custo para ter um setor de TI e funcionários especializados na própria empresa é muito alto (LAUDON, 2004; BEAL, 2005). Conforme Barthélemy (2001) e DiRomualdo (1998) apud. Franco (2014), “as principais razões pelas quais as empresas optam por terceirizar são: (a) custo reduzido; (b) melhor desempenho; e (c) acesso a mercados mais amplos de trabalho”.

Nesse cenário de terceirização de desenvolvimento de software, é de fundamental importância o conteúdo do contrato firmado entre clientes e fornecedores. Nas últimas décadas, os fornecedores vinham utilizando metodologias tradicionais de desenvolvimento de software, onde a estrutura BDUF (*Big Design Up Front*) era comum. Isso significa que o levantamento das necessidades e requisitos do cliente deveria ser completamente realizado no início do projeto, contemplando documentos de requisitos, design, modelos e diagramas, para que, somente então, fosse iniciado o desenvolvimento do sistema. Meses (ou anos) depois, o software desenvolvido seria, então, entregue ao cliente.

Esse modelo de desenvolvimento (chamado de tradicional) era comum no século passado, e ainda é utilizado por algumas organizações (PRESSMAN, 2009). No entanto, essas organizações, com o passar dos anos e dos projetos, foram adquirindo experiência e percebendo que esse modelo nem sempre trazia resultados positivos. Na verdade, de acordo com uma pesquisa (STANDISH, 1995), a grande maioria desses projetos falha, atrasa ou termina mais cara do que inicialmente planejado. E o motivo desses problemas é recorrente e perceptível: o cliente não sabe tudo que quer inicialmente, e vai sempre querer mudanças (NERUR et al., 2005). Esses problemas favoreceram o surgimento dos Métodos Ágeis.

A criação dos Métodos Ágeis teve como motivação os problemas dos modelos tradicionais, como dificuldade de mudanças nos requisitos, falta de comunicação com o cliente, desenvolvimentos em longos períodos, fases sequenciais e dependentes, e

desconfiança de ambas as partes. Entretanto, se forem analisados cada um desses itens, pode-se perceber que grande parte está atrelada a cláusulas dos contratos firmados entre as organizações como, por exemplo, a dificuldade de mudanças dos requisitos, porque o contrato não permite (ou é muito caro) que o escopo seja alterado, ou também a existência de longos períodos desenvolvimento porque o contrato não prevê entregas parciais. Claramente, podemos perceber que os contratos estão fortemente ligados à forma como o projeto vai se desenrolar. Alguns fornecedores, inclusive, tentam se beneficiar disso, como mostra o trecho abaixo:

“Os orçamentos previstos, na maioria dos contratos para construir uma aplicação de software, são severamente onerados pelas taxas de alteração. Cobrar taxas excessivas de alteração constitui, na verdade, o modelo de negócios de vários fornecedores que prestam esse tipo de serviço” (SUTHERLAND, 2014)

Resumindo, este trabalho está inserido em um cenário de contratações de desenvolvimento de software, onde houve uma mudança cultural muito grande devido ao surgimento dos métodos ágeis. Entretanto, o escopo da pesquisa foi voltado para organizações/empresas privadas. As públicas também são citadas em alguns momentos durante o texto, mas não são o enfoque da pesquisa, devido às legislações vigentes relacionadas à contratação de desenvolvimento de software impedirem muitas práticas e conceitos desenvolvidos ao longo do texto (FRANCO et al., 2013), (FRANCO, 2014), (PRIKLADNICKI et al, 2015). Apesar disso, diversas práticas aqui apresentadas podem ser de grande valor para profissionais envolvidos com contratações na esfera governamental, desde que avaliada a compatibilidade com as leis em vigor. Já há uma mobilização para a modificação dessas leis de contratação de desenvolvimento de software brasileiro, que atualmente é fortemente ligada a Pontos de Função. Porém, esse não é o escopo deste trabalho, apesar de ser uma área de estudo muito relevante atualmente.

1.2 Descrição do Problema

O problema abordado neste trabalho está relacionado com contratação de desenvolvimento de software e com o advento dos Métodos Ágeis. Atualmente, as Metodologias Ágeis já são uma realidade, adotadas por grande parte das organizações e colecionando inúmeros *cases* de sucesso (VERSION ONE, 2015). Entretanto, quando olhamos

para os contratos de desenvolvimento de software que as organizações têm usado, vemos que eles ainda não foram adequados à nova forma de trabalho introduzida pelas práticas ágeis. Além disso, essa falta de sinergia entre os contratos e os modelos de desenvolvimento é um dos fatores que pode impactar no sucesso do projeto. Tanto o cliente quanto o fornecedor podem ter desvantagens decorrentes dessa falta de alinhamento entre a metodologia de desenvolvimento utilizada e as cláusulas do contrato assinado.

Este trabalho tem o objetivo de expor o problema relacionado às divergências entre os contratos e as metodologias de desenvolvimento, mergulhando em suas causas e consequências. Relacionadas a esse problema, algumas questões são comuns:

- “Por que projetos falham ou atrasam, mesmo utilizando modernas metodologias ágeis?”
- “Por que algumas organizações ainda têm resistência para adotar métodos ágeis?”
- “Como melhorar a relação entre o cliente (solicitante do projeto) e o fornecedor (quem vai desenvolver o projeto)?”

As respostas para essas perguntas ficam mais claras no decorrer deste texto.

O cliente acredita que vai receber um produto em X meses, pois tem a ilusão de que o prazo, custo e escopo estão definidos. Historicamente, é possível perceber que o cliente não sabe tudo o que quer no início do projeto (KEITH, 2002). De acordo com um relatório do *Standish Group* de 2013, projetos de grande porte têm muito mais chance de falhar ou atrasar do que os de pequeno porte¹ (**Figura 1**) (STANDISH, 2013). Em projetos grandes, o cliente recebe seu projeto ou com atraso, ou mais caro, ou diferente daquilo que ele queria (ou até uma combinação dessas três consequências). Mesmo sabendo disso tudo, utilizam contratos tradicionais, por desconhecimento, medo do novo e comodidade. O fornecedor, que se vê forçado a usar um contrato que não reflete sua forma de trabalho, tem pouca participação do cliente, dificuldade de solicitar mudanças no projeto, pouco valor agregado no desenvolvimento e insatisfação da equipe.

¹ Projeto de grande porte é um projeto com mais de 10 milhões de dólares em custos, enquanto um projeto de pequeno porte tem menos de 1 milhão de dólares em custos. O status *Problemas* representa que houve problema em relação a prazo, custo ou escopo (STANDISH, 2013).

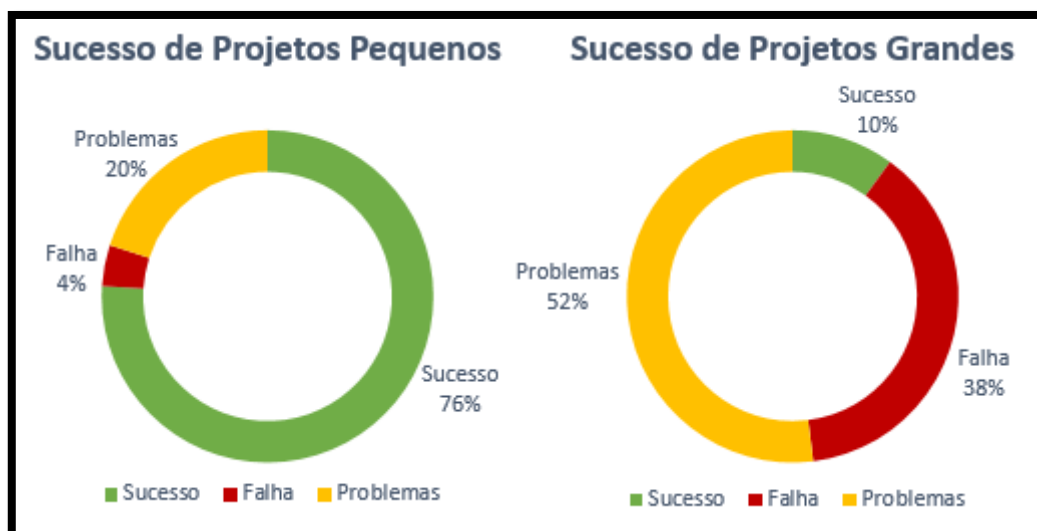


Figura 1: Percentual de sucesso em projetos pequenos e grandes

Há também o caso de organizações que ainda não trabalham com métodos ágeis porque gerentes e diretores têm receio de trocar seus contratos tradicionais de escopo fechado (que dificulta mudanças nos requisitos) por contratos de escopo flexível (que facilita mudança nos requisitos). Ficando no modelo tradicional, no caso de uma falha, a culpa é do modelo. Mas, se optar por um modelo ágil novo, a culpa de uma possível falha cai toda em quem mudou o que “funcionava”.

Há casos de organizações que sabidamente utilizam contratações tradicionais para projetos com desenvolvimento ágil. Essa prática afeta as características que tornam o projeto “ágil” e prejudica seus resultados.

Em resumo, este trabalho busca tratar o seguinte problema e tem as seguintes questões:

Problema: Várias organizações já estão utilizando métodos ágeis, obtendo resultados positivos com sua adesão. Porém, a maioria delas não se preocupou com o contrato utilizado quando se trata de desenvolvimento de terceiros. Elas continuam usando contratos de metodologias tradicionais, mesmo quando querem utilizar metodologias ágeis. Essa incompatibilidade gera problemas tanto para o lado contratante, quanto para o lado contratado.

Questões: Será que existe alguma forma de contratação que seja benéfica tanto para o contratante como para o contratado? Será que sempre alguém tem que sair perdendo ou existe alguma abordagem ganha-ganha?

1.3 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma:

Capítulo Dois: Explica a metodologia de pesquisa adotada ao longo desse trabalho. Está subdividido em duas seções. Na primeira, é explicada a Revisão Sistemática de Literatura utilizada para identificar trabalhos relacionados ao tema desta tese. É descrito o protocolo de revisão sistemática utilizado, apresentado o resultado obtido com a busca, explicada a seleção realizada e, por fim, são extraídos dados das publicações selecionadas. Na segunda seção, é explicado como é feita a preparação e a execução de questionários de pesquisa, que são utilizados ao longo da pesquisa para descobrir a opinião de respondentes de empresas públicas e privadas sobre a relevância das principais características de contratos de desenvolvimento de software.

Capítulo Três: Apresenta o referencial teórico necessário para que o trabalho seja compreendido por completo. A primeira seção fala sobre as Metodologias Ágeis e contém um breve histórico da engenharia de software até o surgimento dos Métodos Ágeis, e o detalhamento das principais características ágeis. A segunda seção apresenta conteúdo sobre contratação de desenvolvimento de software.

Capítulo Quatro: Apresenta os modelos de contrato identificados através de pesquisa na bibliografia selecionada. São descritos detalhadamente quatro modelos de contratos tradicionais, mais utilizados em projetos de desenvolvimento de software com metodologias tradicionais, e sete modelos de contratos ágeis, que são modelos novos utilizados em projetos de desenvolvimento de software com características ágeis.

Capítulo Cinco: Apresenta as principais características presentes em um contrato de desenvolvimento de software. São explicadas e analisadas minuciosamente dez características consideradas por este trabalho como as mais relevantes em um contrato. Essas características são as mesmas utilizadas no questionário de pesquisa citado no Capítulo Dois, e serão priorizadas de acordo com os resultados da pesquisa apresentados no Capítulo Seis.

Capítulo Seis: Mostra os resultados obtidos com a pesquisa através do questionário descrito no Capítulo Dois. São observados todos os resultados obtidos individualmente, considerando cada pergunta feita e grupamento de respondentes possíveis, como, por exemplo, uma análise para as respostas de respondentes apenas de setor privado. Na segunda

seção do capítulo é feita uma análise geral do resultado obtido com o questionário e são tomadas conclusões relacionadas às características dos contratos e seus respondentes.

Capítulo Sete: Apresenta, a partir dos resultados da pesquisa do Capítulo Seis, uma análise dos modelos de contrato identificados no Capítulo Quatro, com enfoque nas características priorizadas do capítulo anterior. Dessa forma, são identificados os modelos de contratos mais aderentes ao pensamento e opinião de cada setor (público/privado e cliente/fornecedor).

Capítulo Oito: Apresenta a conclusão do trabalho. Realiza uma análise do atingimento dos objetivos definidos na introdução e uma análise da condução do trabalho. Cita a principal contribuição desta tese, mas também deixa claras as limitações e as nuances que ela não cobriu. Por fim, são descritos os possíveis trabalhos futuros a partir desta dissertação.

2 Metodologia de Pesquisa

Neste capítulo, serão apresentadas as principais metodologias de pesquisa utilizadas ao longo do trabalho. Uma Revisão Sistemática de Literatura foi utilizada para buscar objetivamente referências bibliográficas e levantar o estado da arte do tema, e posteriormente foi desenvolvido e executado um questionário de pesquisa para coletar dados e opiniões de fontes confiáveis sobre as informações obtidas na Revisão Sistemática de Literatura.

2.1 Revisão Sistemática de Literatura

A Revisão Sistemática de Literatura (RSL) é uma metodologia de estudo secundária que surgiu em pesquisas na área de saúde, devido ao grande volume de informações disponíveis sobre medicina e necessidade de sintetizar e validar esses dados (KITCHENHAM, 2004), (KITCHENHAM, 2007). O fato de ser secundária significa que ela revisa os estudos primários – estudos empíricos responsáveis por investigar uma questão de pesquisa específica – com o objetivo de integrar e sintetizar as evidências relacionadas a essa questão de pesquisa.

A RSL é utilizada para estabelecer um levantamento formal do estado da arte de um determinado tema de forma robusta e consistente a partir de um planejamento e execução criteriosos. Para sua execução, um processo rigoroso é conduzido segundo uma série metodologicamente bem definida de etapas, que são previamente estabelecidas em um protocolo de estudo chamado Protocolo de Revisão Sistemática.

Dentre as principais vantagens da utilização de uma Revisão Sistemática de Literatura (BIOLCHINI et al., 2005) estão:

- **Redução da subjetividade:** Todos os parâmetros e critérios relacionados à busca são definidos com antecedência no Protocolo de Revisão Sistemática, o que aumenta drasticamente a objetividade da pesquisa.
- **Redução do viés:** A definição e planejamento da revisão também minimizam os possíveis vieses na pesquisa.
- **É replicável:** Por ser amplamente planejada e criteriosa, é possível a qualquer momento reexecutar a revisão e obter resultados muito próximos dos obtidos anteriormente.

- **É sintético:** Possui grande poder de síntese, permitindo consolidar grandes volumes de informação.

“A revisão sistemática é o método mais científico de resumir a literatura, pois protocolos específicos são usados para determinar criteriosamente que estudos serão incluídos na revisão.” (CHUNG et al., 2006)

2.1.1 Protocolo de Revisão Sistemática - Conceito

O Protocolo de Revisão Sistemática é um conjunto de orientações e informações que norteiam a execução da RSL. Ele contém todas as informações, critérios e planejamentos necessários para a execução da Revisão. O protocolo deve ter detalhes sobre os objetivos da RSL, explicitar claramente a(s) questão(ões) de pesquisa e as estratégias de busca, definir os critérios de inclusão e exclusão de publicações, e detalhar a estratégia de extração dos dados obtidos (RANDOLPH, 2009). Na **Figura 2** estão ilustrados os procedimentos de definição do Protocolo de Revisão Sistemática e de execução do mesmo, conforme as seguintes atividades:

Planejamento do Protocolo de Revisão Sistemática:

1. Definir Objetivos da Revisão Sistemática: Inicialmente, definir qual é o objetivo almejado com a execução da RSL;
2. Definir Questão de Pesquisa: Escrever explicitamente qual é questão de pesquisa que vai servir de base para a definição das palavras-chave da pesquisa;
3. Identificar Bases de Dados: Listar quais bases de dados serão utilizadas como fontes de publicações durante a busca;
4. Escolher Tipos de Estudos Considerados: Definir quais tipos de trabalhos serão considerados na pesquisa. Ex: publicados, não publicados, artigos de revistas, artigos de *journals*, etc;
5. Definir Palavras-chave Utilizadas: Definir quais serão as palavras-chave e suas combinações ou traduções que serão entrada para a busca nas fontes selecionadas. A escolha das palavras-chave é justificada pela relação com a questão de pesquisa;
6. Identificar Restrições e Limitações da Pesquisa: Listar todas as restrições e limitações que devem ser aplicadas durante a pesquisa;

7. Identificar Critérios de Inclusão e Exclusão de Publicações: Identificar e descrever todos os critérios que serão utilizados para filtrar as publicações;
8. Definir Procedimento de Seleção das Publicações: Descrever procedimento de seleção, definindo ordem de aplicação de critérios e de execução de atividades;
9. Definir Estratégia para Extração dos Dados: Definir como será feita a extração das informações buscadas dentro das publicações.

Execução da Revisão Sistemática:

1. Conduzir Busca nas Bases de Dados Identificadas: Executar a busca simultânea das palavras-chave escolhidas nas bases de dados que foram selecionadas;
2. Registrar Todas as Informações e Dados Durante a Busca: Todas as informações da pesquisa e sobre a pesquisa devem ser registradas para uso posterior. Ex: quantidade de resultados retornados para determinada busca em determinada base de dados;
3. Aplicar Critérios Definidos no Protocolo de Revisão: Em cada publicação identificada, aplicar os critérios definidos para filtrar e decidir sobre sua possível inclusão nos resultados;
4. Selecionar Publicações: O processo deve ser repetido em todas as bases de dados e as restrições, limitações e critérios aplicados em todos os casos. Uma vez realizado todo esse procedimento, a saída será um conjunto de publicações de alta relevância para a questão de pesquisa;
5. Extrair Dados das Publicações: Com o conjunto filtrado de publicações, deve-se extrair delas tudo que for relevante para responder a questão de pesquisa, seguindo um procedimento que define o que deve ser aproveitado e onde deve ser registrada essa informação.

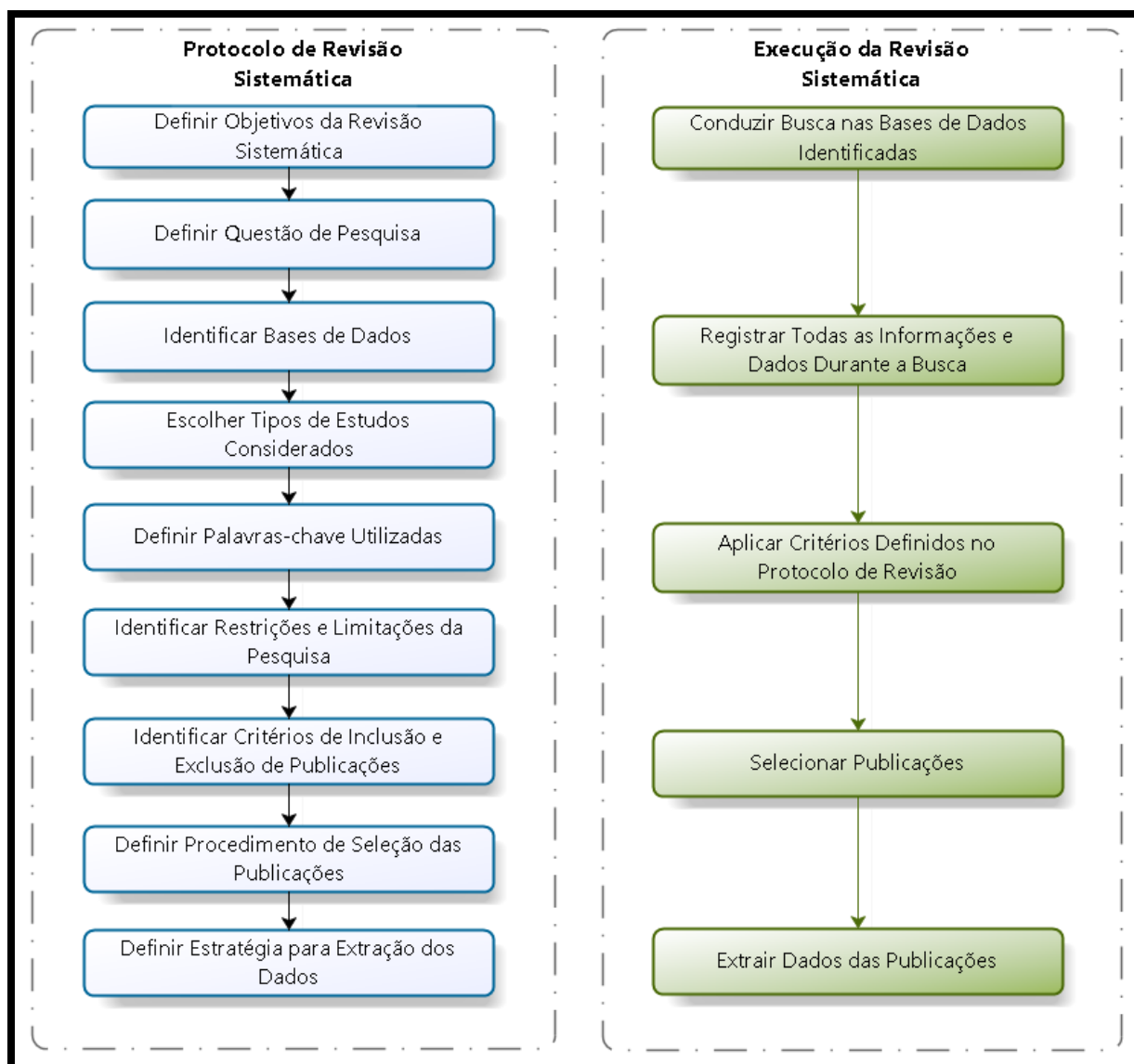


Figura 2: Diagramas de Planejamento do Protocolo de Revisão Sistemática e Execução da Revisão Sistemática

2.1.2 Protocolo de Revisão Sistemática - Execução

Seguindo o procedimento definido na seção anterior, este trabalho definiu um Protocolo de Revisão Sistemática da seguinte forma (**Tabela 1**):

Tabela 1: Protocolo de Revisão Sistemática

Objetivo da Revisão Sistemática	• Identificar a maior quantidade possível de modelos de contratos e características comuns em contratos.
Questão de Pesquisa	• “Quais são os principais modelos de contratos de desenvolvimento de software?”
Bases de Dados Consultadas	• ACM (Association for Computing Machinery); • IEEE; • Springer International Publisher Science; • Google Scholar.
Tipos de Estudo Considerados	• Artigos publicados em revistas ou jornais; • Teses de Mestrado; • Livros publicados.
Palavras-chave	• Contratos Ágeis; • Agile Contracts; • Agile+Contracts.
Restrições e Limites	• Buscas devem ser feitas na mesma semana; • Somente artigos em português ou inglês; • Buscas devem ser feitas manualmente; • Se a busca retornar mais de 1000 resultados, a análise manual e individual torna-se inviável. Por isso, nesse caso o limite para verificação é de 50 artigos.
CrITÉrios para Seleção de Artigos	• Artigos relacionados a empresas particulares (privadas); • Artigos publicados depois de 2011, porque as áreas de TI e Engenharia de Software são muito dinâmicas, além de que o uso de artigos antigos poderia implicar em diferentes contextos de época, o que poderia deturpar os resultados; • Artigos que falem exclusivamente de contratos de desenvolvimento de software;

	• Artigos claramente relacionados às metodologias ágeis de desenvolvimento de software; • Ordenação feita pelo algoritmo de relevância das bases de dados. Obs: Os critérios devem ser TODOS atendidos para que o artigo seja selecionado.
Procedimento de Seleção	1) Etapa 1: Verificação do ano de publicação 2) Etapa 2: Leitura do título e do resumo/<i>abstract</i> 3) Etapa 3: Leitura dinâmica do artigo Obs: Caso a base de dados não permita filtrar por ano, as etapas 1 e 2 são invertidas.
Estratégia para Extração dos Dados	1) Leitura completa do artigo; 2) Identificação de modelos de contratos; 3) Identificação de características comuns em modelos de contratos; 4) Coleta das informações

2.1.3 Execução da Busca

A busca foi realizada entre os dias 4/jan/2015 e 8/jan/2015, e obteve os resultados abaixo:

No **ACM**, a busca por *Agile Contracts* retornou inicialmente 201 resultados. Porém, a busca das expressões *Agile+Contracts* e Contratos Ágeis não retornaram resultados. Dos 201 resultados, apenas 29 eram posteriores a 2011 (etapa 1). Desses 29, apenas um mostrou-se relacionado com o tema buscado após a leitura do título e do abstract (etapa 2).

No **IEEE**, a busca por *Agile Contracts* retornou 2.080.000 resultados. Porém, a busca da expressão exata *Agile+Contracts* retornou 6 resultados. Além disso, a busca por Contratos Ágeis não retornou resultados. Dos 2.080.000 resultados, foram avaliados os 50 mais relevantes (relevância dada pela ferramenta de busca) porque o sistema não permite filtrar por ano de publicação. Desses, apenas 3 estavam na temática desejada (etapa 1), e desses 3, apenas 1 era mais recente que 2011 (era de 2012) (etapa 2). A busca pela expressão exata *Agile+Contracts* retornou 6 resultados, mas todos eram mais antigos que 2011 (etapa 2).

No **Springer**, a busca por *Agile Contracts* retornou 4.094 resultados. Porém, a busca da expressão exata *Agile+Contracts* retornou 11 resultados. Além disso, a busca por Contratos Ágeis não retornou resultados. Dos 4.094 resultados, foram avaliados os 50 mais relevantes (relevância dada pela ferramenta de busca) porque o sistema não permite filtrar por ano de publicação. Desses, 10 eram relevantes e relacionados ao tema. Entretanto, apenas 4 eram posteriores a 2011. A busca pela expressão exata não retornou relevantes diferentes da busca por *Agile Contracts*.

No **Google Scholar**, a busca por *Agile Contracts* retornou 33.200 resultados. Porém, a busca da expressão exata *Agile+Contracts* retornou 105 resultados. Além disso, a busca por Contratos Ágeis retornou 19.300 resultados. Dos 33.200 resultados para a busca *Agile Contracts*, 11.300 eram posteriores a 2011. Realizada a análise, considerando os critérios e limites definidos, foram selecionados 6 artigos relacionados ao tema de interesse. Com relação à expressão exata *Agile+Contracts*, não foi encontrado nenhum resultado relevante diferente dos resultados da busca por *Agile Contracts*. A busca por Contratos Ágeis não produziu nenhum resultado que estivesse aderente aos critérios descritos.

Após a etapa 2, temos como resultado parcial $1+6+1+4 = 12$ artigos. Esses 12, então, passaram em seguida pela etapa 3 do procedimento de seleção, que é a leitura dinâmica dos artigos. O objetivo desse último filtro é selecionar apenas os artigos que estão fortemente relacionados ao tema desejado, de acordo com os critérios de seleção definidos.

A leitura dinâmica foi realizada e, dos 12 artigos, apenas 6 eram de fato relacionados ao tema desejado. Os outros 6 desviaram do tema, e, dessa forma, foram filtrados seguindo os critérios definidos. Portanto, a partir desses 6, foi feita a extração dos dados relevantes para responder a questão de pesquisa e atingir o objetivo da RSL. A **tabela 2** resume os resultados após cada etapa da RSL.

Com um resultado tão reduzido, pode-se argumentar que a filtragem foi rigorosa até demais. No entanto, esse foi realmente o objetivo, já que, apesar do objetivo da RSL ser identificar o maior número de contratos de desenvolvimento de software, o objetivo da tese é mais específico, buscando analisar, principalmente, os modelos de contratos com características de métodos ágeis. Dessa forma, estreitamos a pesquisa para apenas as palavras-chave relacionadas a contratos ágeis para garantir esse alinhamento maior com o objetivo da dissertação.

Ao final do trabalho, depois de toda a pesquisa executada e resultados obtidos e registrados, foi executada uma nova RSL alterando um dos critérios definidos: seriam aceitos artigos e trabalhos a partir de 2008. O objetivo dessa nova rodada da RSL foi verificar se havia sido deixado de fora algum trabalho relevante que contivesse modelos de contratos não abordados ou características de contratos diferentes das utilizadas. A execução da busca nas quatro bases selecionadas não retornou nenhum novo resultado para a palavra-chave Contratos Ágeis. No entanto, para a palavra-chave Agile Contracts, dois novos resultados relevantes foram encontrados na IEEE, um resultado relevante novo foi encontrado no *Springer* e três resultados relevantes foram encontrados no Google Scholar (porém esses resultados eram os mesmo encontrados no IEEE e *Springer*). Dessa forma, três novos resultados foram analisados, e, por fim, após a leitura dos artigos, foi possível concluir que, apesar de apresentarem informações de contratos presentes nesta pesquisa, eles não citavam modelos de contratos ou características de contratos diferentes das encontradas ao executar a RSL utilizando como critério a publicação a partir de 2011.

Tabela 2: Resumo dos resultados da Revisão Sistemática de Literatura

Base	Palavras-chave	Resultados iniciais	Pós etapa 1	Pós etapa 2	Pós etapa 3
ACM	<i>Agile Contracts</i>	201	29	1	0
	<i>Agile+Contracts</i>	0	0		
	Contratos Ágeis	0	0		
IEEE	<i>Agile Contracts</i>	2,080,000	3	1	1
	<i>Agile+Contracts</i>	6	6		
	Contratos Ágeis	0	0		
Springer	<i>Agile Contracts</i>	4.094	10	4	1
	<i>Agile+Contracts</i>	11	10		
	Contratos Ágeis	0	0		
Google Scholar	<i>Agile Contracts</i>	33,200	11300	6	4
	<i>Agile+Contracts</i>	105	52		
	Contratos Ágeis	19,300			

2.1.4 Extração de Dados das Publicações Selecionadas

Foram, portanto, selecionadas 6 publicações. As principais características e semelhanças delas com relação a este trabalho são:

- 1) O artigo chama-se *Contracting for Agile Software Development Projects* (EDWARDS, 2012). Ele foi selecionado porque também busca uma solução para o mesmo problema estudado neste trabalho, citando pontos fundamentais que devem ser observados em cada um dos aspectos do contrato e como ele afeta cada uma das partes de uma equipe de desenvolvimento. As características identificadas no artigo motivaram a pesquisa executada no presente trabalho.
- 2) O artigo tem como título *Agile Contracts Primer* (ARBOGAST et al., 2012). Esse artigo contém uma abordagem mais abrangente, mas também busca discutir o mesmo problema já citado. Ele exhibe o mesmo problema e detalha modalidades de contrato ágeis já conhecidas, descrevendo vantagens e desvantagens. Ele chega a uma conclusão teórica, mas não demonstra uma opinião sobre qual traz mais vantagens e nem cita experimentação em empresas reais.
- 3) O artigo é intitulado *The Necessity of New Contract Models for Agile Project Management* (RICO, 2011). Esse artigo tem uma abordagem mais informal e concisa, e objetiva difundir na comunidade de desenvolvimento de software a

necessidade de novos modelos de contrato quando tratamos de projetos ágeis. Ele cita pontualmente diversas modalidades de contratações tradicionais e também modalidades de contratações ágeis, e realiza comparações entre elas. Apesar de não entrar a fundo nos detalhes de cada modelo de contrato, ele dá uma visão geral e clara dos motivos pelos quais são necessários novos modelos de contrato atualmente.

- 4) Uma tese de mestrado com o título *Contracts in Agile Software Development* (LAAKKONEN, 2014). Essa tese estuda o mesmo problema que o abordado neste trabalho, mas com um enfoque local do país de origem do autor, que é a Finlândia. O autor percebe que as metodologias ágeis já estão bem difundidas e não são mais uma novidade, mas, simultaneamente, nota que as formas de contratação das empresas não evoluíram junto, e as mesmas continuam contratando com escopo fechado e levantamento de requisitos completo no início do projeto, que são práticas que vão contra as orientações das metodologias ágeis. O autor, então, faz um estudo dos aspectos mais críticos a serem tratados durante a contratação de métodos ágeis, e acompanha empresas que tentaram realizar contratações diferenciadas. Além disso, o autor estuda as regras de contratação de software finlandesas e conclui que as próprias regras não contribuem para uma contratação mais ágil. Dessa forma, esse trabalho é muito relevante por estudar a fundo o problema que abordamos, e acompanhar empresas reais com o mesmo problema. No entanto, ele difere do presente trabalho porque este faz um levantamento de diversos modelos de contratos, tanto ágeis quanto tradicionais. Além disso, este trabalho tem um enfoque maior no mercado brasileiro e americano, diferentemente do enfoque europeu da tese de mestrado citada.
- 5) Um livro chamado *Agile Contracts – Creating and Managing Successful Projects with Scrum* (OPELT et al., 2013). Esse livro é inteiramente focado no problema do presente trabalho. Ele argumenta que as formas tradicionais de contratação com preço fixo são antiquadas e só trazem desvantagens. Para resolver tal problema, os autores propõem um formato de contrato chamado *Agile Fixed Price*, onde o preço é fixo, mas o escopo pode variar de forma que ainda fique sob controle de ambas as partes. Esse modelo de contrato é exemplificado por completo no livro,

e em cima dele é feita uma experimentação com projetos reais. Esse livro é relevante porque estuda a fundo, implementa e experimenta um modelo de contrato que é bastante interessante para este trabalho.

- 6) Uma publicação com o título *adVANTAGE: A Fair Pricing Model for Agile Software Development Contracting* (BOOK et al., 2012). Essa publicação também tem enfoque no problema de interesse, mas descreve um método original e inédito para a contratação de desenvolvimento de software ágil, com o principal objetivo de dividir igualmente os riscos do projeto para o contratante e o contratado. Apesar de ser uma metodologia interessante, ela difere parcialmente deste trabalho, por ela ser centrada em um domínio de clientes mais conservadores, que ainda apresentam alguma resistência para tratar mudanças.

Seguindo o procedimento de extração de dados definido no Protocolo de Revisão Sistemática, foram extraídos de todas as seis publicações os modelos de contratos neles explicados. Em seguida, os modelos de contratos foram analisados, para que fossem identificadas as características comuns desses modelos. Elas foram, então, registradas em uma planilha temporária para posterior tratamento e estudo, bem como os modelos de contrato descobertos.

2.2 Questionário de Pesquisa

Questionários de pesquisa são utilizados para coletar dados e informações de grandes quantidades de respondentes (CHAGAS, 2000). São preparadas sequências de perguntas relacionadas à questão de pesquisa que se pretende responder. Em seguida, o questionário é revisto e validado para garantir que seu resultado será representativo, não apresenta vieses e está alinhado ao objetivo da pesquisa. Identificam-se, então, os possíveis respondentes que proverão respostas significativas para o questionário, e, por fim, ele é divulgado. Deve-se, então, aguardar o recebimento das respostas, para que, finalmente, seja feita uma análise dos dados obtidos com as respostas dos questionários. Veremos com detalhes esse procedimento nas seções seguintes.

2.2.1 Preparação do Questionário

Através da ferramenta *Google Forms* e com base em referências em criação de questionários (GUNTHER, 2003), foi criado um questionário com questões objetivas introdutórias, como:

- “Você já passou pela experiência de ter o trabalho da sua equipe limitado/prejudicado por questões contratuais?”
- “Caracterize sua organização em Pública/Privada/Sens fins lucrativos/Instituto”
- “Você trabalha de que lado? (contratante, contratado, ambos ou nenhum dos dois)”

Em seguida, foi preparada uma tabela (**Figura 3**) com as características presentes em contratos de desenvolvimento de software, identificadas durante a Revisão Sistemática de Literatura, com cinco opções para classificá-las: “Fundamental”, “Muito importante”, “Importante”, “Relevante” e “Pouco Relevante”. O objetivo desse questionário era descobrir quais das características são consideradas mais importantes em um contrato de desenvolvimento de software.

	Fundamental	Muito Importante	Importante	Relevante	Pouco Relevante
Precificação	☐	☐	☐	☐	☐
Vigência do Contrato	☐	☐	☐	☐	☐
Mudança	☐	☐	☐	☐	☐
Garantia	☐	☐	☐	☐	☐
Envolvimento Cliente	☐	☐	☐	☐	☐
Método de Estimativa	☐	☐	☐	☐	☐
Risco para Contratante	☐	☐	☐	☐	☐
Risco para Contratado	☐	☐	☐	☐	☐
Popularidade	☐	☐	☐	☐	☐
Vantagem para Usuário Final	☐	☐	☐	☐	☐

Figura 3: Tabela inicial de classificação do questionário

2.2.2 Validação e Correções no Questionário

Com o questionário montado, foram executadas três rodadas de validação com respondentes diferentes. O objetivo dessa validação foi concluir se o questionário estava intuitivo, completo e gerando as respostas no padrão previsto. De fato, a validação se mostrou

fundamental para a melhoria do questionário. Foi detectado que, ao dar as cinco opções de classificação acima, os respondentes tendiam a classificar a maioria como Fundamental, e apenas ocasionalmente divergiam dessa opção. Percebeu-se, então, que esse formato em tabela estava induzindo a uma forma indesejada de classificação: estavam avaliando cada característica individualmente. Porém, para o objetivo dessa pesquisa, uma classificação que compara cada característica entre elas mesmas é mais adequada. Dessa forma, com o intuito de forçar um pensamento comparativo nos respondentes, a tabela foi substituída por uma pesquisa utilizando *Dot Voting*. Isso tornou a avaliação mais criteriosa e muito mais próxima do objetivo desejado, que é a identificação das características mais importantes (**Figura 4**).

The image shows a digital interface for a Dot Voting exercise. At the top, a blue header bar contains the text: "Vote on Dot Voting 'Quais são as características mais importantes em um contrato de desenvolvimento de software?'". Below this, a list of ten characteristics is presented, each preceded by a right-pointing triangle and followed by a radio button. The characteristics are: Precificação, Vigência do Contrato, Mudanças, Garantia, Envolvimento do Cliente, Método de Estimativa, Risco para Cliente, Risco para Fornecedor, Popularidade, and Vantagem para Usuário Final.

Vote on Dot Voting 'Quais são as características mais importantes em um contrato de desenvolvimento de software?'	
▶ Precificação	☐
▶ Vigência do Contrato	☐
▶ Mudanças	☐
▶ Garantia	☐
▶ Envolvimento do Cliente	☐
▶ Método de Estimativa	☐
▶ Risco para Cliente	☐
▶ Risco para Fornecedor	☐
▶ Popularidade	☐
▶ Vantagem para Usuário Final	☐

Figura 4: Tabela usada pelo Dot Voting

2.2.3 Dot Voting

O *Dot Voting* (DICEMAN, 2013) é um método de facilitação recente, que vem sendo usado por times ágeis para facilitar a tomada de decisão e votações durante projetos. Seu funcionamento é um pouco diferente de uma votação comum. Dada uma pergunta e um conjunto de tamanho médio/grande de opções, deseja-se descobrir quais das opções são as preferidas, por exemplo. Cada votante recebe certa quantidade de pontos (*dots*), que costuma ser uma quantidade próxima da metade das opções. Em seguida, vota-se distribuindo seus pontos nas opções identificadas, podendo colocar todos os pontos na mesma opção ou

distribuí-los da forma que desejar. O resultado é um ranking das opções que mostra claramente dos mais votados até os menos votados. Esse tipo de votação tem ganhado expressividade porque, diferentemente de métodos comuns onde se classifica individualmente cada opção em alguma escala (ex: Bom, Médio, Ruim), o *Dot Voting* força uma votação comparativa, na qual passa a ser necessária uma análise em conjunto mais criteriosa das opções. Além disso, é um método muito rápido e simples de ser utilizado.

2.2.4 Identificação dos Respondentes e Divulgação

O passo seguinte é a identificação de potenciais respondentes para a pesquisa. O autor deste trabalho buscou em seus contatos pessoas que trabalham com contratação de desenvolvimento de software e/ou que têm vasto conhecimento nessa área, e enviou o questionário a elas. Foram identificadas aproximadamente 60 pessoas com conhecimento suficiente para oferecer uma opinião significativa para a pesquisa.

2.2.5 Coleta e Análise das Respostas

Alguns dias depois da divulgação, foi feita uma consolidação das respostas dos questionários, com a preparação de gráficos e classificações para ilustrar o resultado obtido com a pesquisa. Ao longo deste trabalho serão apresentados mais detalhes sobre a execução desse questionário. O Capítulo Cinco explica individualmente cada uma das características de contratos de desenvolvimento de software incluídas no questionário, e o Capítulo Seis contém os resultados, análises e gráficos obtidos a partir do questionário.

3 Referencial Teórico

Neste capítulo é apresentado todo o conteúdo teórico necessário para o completo entendimento do presente trabalho. Haja vista que o tema desta dissertação permeia a contratação de projetos com metodologias ágeis, julga-se pertinente que o leitor seja contextualizado nas principais características dos métodos ágeis e tenha conhecimento sobre o funcionamento de contratos de desenvolvimento de software.

3.1 Metodologias Ágeis

Para que seja possível entender o problema tratado neste trabalho, é necessário apresentar algumas informações sobre as Metodologias Ágeis, como histórico, conceitos e premissas. Isso é importante para que o leitor possa entender a diferença entre as necessidades de um projeto tradicional e de um projeto ágil. Fala-se um pouco da história da engenharia de software e a motivação que incentivou o surgimento dos métodos ágeis. Por fim, são explicadas algumas das principais características ágeis existentes.

3.1.1 Histórico da Engenharia de Software e Metodologias Ágeis

Ao olhar uma linha do tempo da história da computação, é possível perceber a evolução no uso de metodologias de desenvolvimento de software. No início da Informática, não existia metodologia de desenvolvimento definida. Levantavam requisitos, desenvolviam e entregavam tudo ao mesmo tempo, sem nenhuma organização clara. Visando organizar esse caos, criou-se o modelo Cascata (*Waterfall*) (1970) (PRESSMAN, 2009), que representou um grande avanço na área de engenharia de software. Nesse modelo, o processo de desenvolvimento é dividido em fases sequenciais, e cada uma deve ser concluída para que a seguinte seja iniciada. Dessa forma, são criadas fases como Requisitos (levantamento dos requisitos do projeto), Elaboração/Desenho (design e modelagem do projeto), Implementação (codificação do projeto), Verificação (Teste), e Manutenção do Sistema. (Figura 5)



Figura 5: Exemplo de Modelo Cascata

Apesar de ganhos no processo de desenvolvimento, novos problemas surgiram. Com o entendimento de que uma fase tem que terminar para que a outra comece, tornou-se fundamental um levantamento de requisitos perfeito para que a construção ocorra sem muitos problemas. Surgiu, posteriormente, a metodologia *RUP* em 1996, desenvolvida pela empresa *Rational Software Corporation*, que é uma divisão da *IBM* (IBM, 2001). Essa metodologia passou a sugerir a implementação de iterações dentro das fases do modelo Cascata (ABRAHAMSSON et al., 2002). A partir daí, surgiu o modelo Iterativo-Incremental, que prevê iterações nas fases e incrementos no produto, criando entregas parciais durante o ciclo de vida (PRESSMAN, 2009).

Muita coisa melhorou, mas alguns problemas persistiam. Ainda era comum os projetos falharem, atrasarem, estourarem o orçamento, ou ainda, implementarem funcionalidades que nem eram usadas (STANDISH, 2002). Ainda, era difícil modificar um requisito e se comunicar com o cliente. Os projetos acabavam não atendendo às reais necessidades do cliente, que normalmente tinham que pagar um valor elevado pelo produto, e provavelmente usariam menos de 20% das funcionalidades implementadas (STANDISH, 2002). Pensando nesses problemas, os métodos ágeis começaram a aparecer.

As metodologias ágeis trouxeram diversas quebras de paradigma que modificaram a forma de pensar e de desenvolver software no mundo inteiro. Elas começaram a surgir no

final dos anos 90, tendo como inspiração as ideias de produção da *Toyota* (*TPS - Toyota Production System*). Várias metodologias, como o *Extreme Programming* (*XP*) (2000), de Kent Beck (BECK, 2000), o *Scrum* (2000), de Ken Schwaber e Jeff Sutherland (BEEDLE et al., 2000), e o *Feature Driven Development* (*FDD*) (1999), de Peter Coad (PRIKLADNICKI et al., 2014), foram criadas nesse período. Os autores dessas metodologias perceberam que tinham muito em comum e decidiram se reunir. Esse encontro dos autores ocorreu em 2001 e ficou conhecido como Manifesto Ágil (BECK et al., 2001). Esse evento foi um marco para a engenharia de software mundial, e é possível resumi-lo em 4 frases:

Indivíduos e Interação entre eles **MAIS QUE** Processos e Ferramentas
 Software em Funcionamento **MAIS QUE** Documentação Abrangente
 Colaboração com Cliente **MAIS QUE** Negociação de Contratos
 Responder a Mudanças **MAIS QUE** Seguir um Plano

Os métodos ágeis então começaram a se popularizar e ganhar adeptos. O XP foi muito utilizado no início da década passada, e deixou claras as vantagens da utilização de metodologias iterativo-incrementais e do *pair programming*. O *Scrum* veio em seguida, e se tornou o método ágil mais adotado (STANDISH, 2011), pregando ciclos curtos, entregas constantes e de valor, times pequenos e escopo flexível (**Figura 6**). Mais recentemente, o *Kanban* tem ganhado popularidade, com seu foco em entrega e integração contínua, sistema puxado e iterações ainda menores.

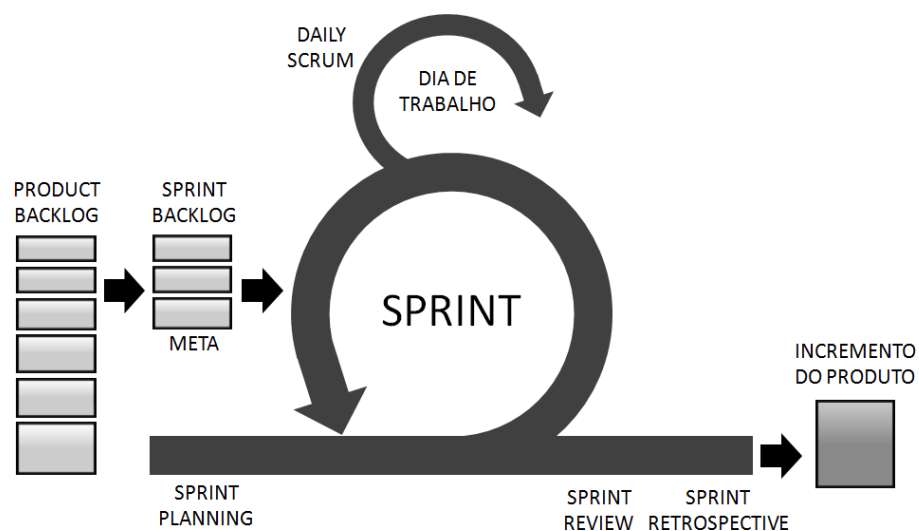


Figura 6: Exemplo de iteração no *Scrum*

3.1.2 Características das Metodologias Ágeis

Os métodos ágeis – principalmente o Scrum, que é o mais adotado (VERSION ONE, 2015) – têm princípios diferentes das práticas clássicas das metodologias tradicionais. As principais são:

- ❖ **Iterações curtas, de 1 a 4 semanas, com entregáveis em todas as iterações:** ao contrário de longos períodos de desenvolvimento, com um grande entregável no final, os métodos ágeis trabalham com pequenas iterações, normalmente de 2 semanas, onde são disponibilizadas no fim de cada uma delas um entregável com funcionalidades de grande valor ao cliente. Essas iterações curtas evitam que seja desenvolvido algo que o cliente não quer e motiva sua interação com a equipe de desenvolvimento. (SCHWABER; SUTHERLAND, 2011)
- ❖ **Times pequenos, multidisciplinares e autogerenciáveis:** Diferente de projetos tradicionais, que utilizam equipes grandes com funções bem divididas, hierarquias e gerências rigorosas, os projetos ágeis limitam seus times em 7~9 pessoas. Nesses times, há desde desenvolvedores, testadores, analistas de requisito até designers. Porém, não há um gerente de projetos. Os times são autogerenciáveis, estimam seu próprio trabalho e são ajudados pelo *scrummaster*. A cobrança passa a ser intrínseca (a própria pessoa se cobrando), e não extrínseca (vinda de um superior). (SCHWABER; SUTHERLAND, 2011)
- ❖ **Foco na entrega de VALOR para o cliente:** O foco do projeto muda em projetos ágeis (GOMES FILHO, 2014). Tradicionalmente, a entrega deveria consistir no que está especificado no documento, independentemente se aquilo ainda é o que interessa para o cliente. Nos métodos ágeis, o foco passa a ser as funcionalidades que agreguem mais valor ao cliente. Devem ser priorizadas as funcionalidades mais importantes para o cliente, e essas são as primeiras a ser entregues. Dessa forma, o cliente já recebe produto de grande valor ainda no desenvolvimento do sistema, sendo possível ter um ROI (*return on investment*) mesmo antes de o projeto estar concluído. (HIGHSMITH, 2002)
- ❖ **Mudanças são bem-vindas e esperadas:** Em projetos tradicionais, o escopo é fechado e completamente definido e acordado no início do projeto. Caso seja necessário mudar algum item do escopo, como um novo requisito ou regra de negócio, há a necessidade

de fazer uma reavaliação do contrato, considerando aditivos a esse contrato para que possam ser incluídas as alterações. Esse procedimento é lento, difícil, caro e burocrático. Então, os autores das metodologias ágeis perceberam que os clientes quase sempre querem fazer mudanças (SCHWABER; SUTHERLAND, 2011), porque é praticamente impossível saber tudo o que eles querem no início do desenvolvimento. Além disso, as condições do mercado costumam mudar (novos concorrentes, novas leis...), o que motiva ainda mais as mudanças. Logo, percebeu-se que é importante criar alguma forma de facilitar as mudanças, porque absorver as mudanças prontamente será mais barato e mais rápido do que ter que tratá-las no final do projeto, o que pode fazer o lucro simplesmente desaparecer.

- ❖ **O escopo é flexível, e o tempo passa a ser fixo (*timebox*):** Metodologias clássicas trabalham com o escopo fechado, e o tempo (que deveria ser fixo) varia para comportar todo o escopo, que deve obrigatoriamente ser todo desenvolvido. Isso gera atrasos e possíveis gastos extras para o projeto. Métodos ágeis mudam isso fixando o TEMPO, ao invés do escopo. (HIGHSMITH, 2002) O escopo passa a ser flexível. O projeto tem um período fixo para ser feito, com N iterações ocorrendo dentro desse período. O escopo de cada iteração vai sendo definido e refinado com o passar do tempo, e o melhor *software*, com as funcionalidades de maior valor, serão entregues no prazo acordado. Se algo ficar de fora, serão as funcionalidades de menor valor. Cabe então ao cliente avaliar se vale a pena estender o projeto para desenvolver o que faltou ou se já pode encerrar o projeto.
- ❖ **Necessário constante envolvimento do cliente:** Nos projetos tradicionais, a interação com o cliente é muito reduzida. Normalmente, ele aparece no início, quando são levantados os requisitos, durante o projeto para pontuais aprovações de documentos, e no final, quando vai receber a entrega do produto construído. Como é possível perceber a partir dos princípios anteriores, nos métodos ágeis a participação constante do cliente é fundamental. Ele precisa participar da priorização das histórias (requisitos), precisa dar *feedback* sobre a entrega realizada, precisa informar mudanças e participar de reuniões de revisão periódicas, dentre outras atividades. (HIGHSMITH, 2002), (SCHWABER; SUTHERLAND, 2011).

- ❖ **Visibilidade e previsibilidade do trabalho do time:** A individualidade é uma característica comum em metodologias tradicionais. É difícil saber no que o colega de equipe está trabalhando, em quanto tempo ele vai terminar, se ele precisa de ajuda, ou até mesmo se ele está conseguindo trabalhar adequadamente. Isso vem como consequência das longas entregas e das hierarquias rigorosas. Contudo, a forma de trabalho ágil modifica toda essa dinâmica. As reuniões diárias favorecem a comunicação do time, deixando claro em que cada pessoa está trabalhando, criando interação entre a equipe e gerando uma cobrança pessoal muito benéfica (SCHWABER, 2004). Com os quadros ágeis, onde cada tarefa é um *post-it* em algum *status* – *to do*, *doing* e *done*, fica fácil ver quem está trabalhando em que, onde estão os gargalos e a previsão de conclusão das tarefas da iteração (**Figura 7**). (AUDY, 2012)

Desenvolvimento		Testes		Homologação		Produção	
TO DO	DOING	TO DO	DOING	TO DO	DOING	DONE	READY
Produto:				Impedimentos:			

Figura 7: Quadro Kanban (AUDY, 2012)

3.2 Contratação de Desenvolvimento de Software Tradicional e Ágil

Contratos estão e sempre estiveram presentes nas relações de negócio entre empresas do ramo de desenvolvimento de software e seus clientes (CRUZ; ANDRADE; FIGUEREIDO, 2011). Um contrato é um acordo entre ambas as partes, onde são definidas todas as regras, obrigações, valores e datas que devem ser respeitadas durante o período que aquele contrato estiver em vigor. Sua necessidade é reconhecida pela maioria das empresas que o utilizam como forma de proteção e garantia legal de que terão o que foi acordado. Existem diversos modelos de contratos que são frequentemente utilizados no mercado, mas é comum que não sejam totalmente adequados ao cenário das empresas envolvidas. Cláusulas e adaptações são, então, criadas para adequar o contrato à situação pretendida.

Nesse trabalho, o principal critério para categorizar um contrato como Tradicional ou Ágil é a forma como o contrato trata o escopo. Caso o contrato defina que o escopo do desenvolvimento é fechado (dificultando mudanças de requisitos), é categorizado como

“Modelo de Contrato Tradicional”. Se o contrato trabalhar com escopo flexível, permitindo e facilitando mudanças de escopo durante o projeto, fica sob a categoria “Modelo de Contrato Ágil”.

Historicamente, organizações vêm utilizando contratos tradicionais para contratação de desenvolvimento de software (CRUZ; ANDRADE; FIGUEREIDO, 2011). Tais contratos tratam escopo de forma fechada e o pagamento costuma ser um valor pré-definido ou por horas de trabalho (*time & materials*). O problema é que a forma de trabalho das empresas mudou e os contratos não acompanharam essa mudança. Recentemente, tentando buscar resolver esses problemas, pesquisadores e empresários estão estudando novas formas de contratação para projetos que utilizam metodologias ágeis. Com isso, novos modelos de contratos vêm surgindo e se tornando mais populares. Porém, até o momento, pouca experimentação foi feita com esses novos modelos de contrato, e ainda há espaço para muita pesquisa.

É importante recordar que, apesar de todo o esforço em volta da criação de novos modelos de contrato para projetos com métodos ágeis, o Manifesto Ágil (BECK et al., 2001) nos lembra:

“Colaboração com o cliente mais que negociação de contratos.”

Ou seja, em métodos ágeis, a criação de centenas de regras em contratos deve ser menos importante que a relação direta com o cliente. Isso não quer dizer que projetos com metodologias ágeis não devem ter contratos. Na verdade, essa citação mostra que os contratos devem, sim, existir. Porém devem conter apenas o que for indispensável, as informações mais importantes. Essas informações mais importantes são o objetivo do questionário de pesquisa citado na seção 2.2 e executado no Capítulo Seis.

4 Identificação de Modelos de Contratos

4.1 Modelos de Contratos Tradicionais

Modelos de contratos que tratam o escopo de forma **fixa**, dificultando mudanças e estimulando a prática sabidamente falha de levantar todo o escopo detalhado do projeto em seu início são considerados neste trabalho como Modelos de Contratos Tradicionais, porque são muito utilizados em empresas que praticam desenvolvimento de software seguindo processos de desenvolvimento tradicionais, como o Cascata. São apresentados nas seções seguintes quatro modelos de contrato identificados através da revisão sistemática de literatura, que exemplificam a forma de trabalho dos modelos tradicionais de contratação.

4.1.1 Contrato Tradicional de Preço Fixo

Como o próprio nome diz, o fornecedor é obrigado a executar o serviço a um preço fixo, acordado inicialmente com o cliente. Esse modelo é muito usado para contratações que seguem a legislação brasileira, e utilizam Análise de Pontos de Função (APF) como forma de estimativa. Os guias e manuais do APF também recomendam esse modelo de contratação (VAZQUES et al., 2004).

Nesse modelo de contrato, além de o preço ser fixo, o escopo e o prazo também o são (UCHOA, 2013). O problema é que esses tipos de contrato geralmente levam a situações de perda-perda para os contratantes e fornecedores. Os clientes costumam receber um produto diferente do que realmente precisam, e fornecedores podem perder dinheiro com muita facilidade, já que para atender o escopo fixo, pode ser necessário aumentar a quantidade de recursos, pagar hora extra, alocar recursos mais caros, ou simplesmente diminuir a qualidade do produto entregue, eliminando testes e qualidade de código, o que gera insatisfação do cliente e um provável atrito no relacionamento entre as partes.

Sabendo de todos esses riscos, os fornecedores não podem subestimar o custo de suas propostas, já que se mal estimado, o lucro pode simplesmente desaparecer durante o projeto. Ao mesmo tempo, se estimarem alto demais, podem perder competitividade e, caso estejam participando de licitações em que ganha a proposta de menor preço, podem perder o projeto. Tudo isso tende a acarretar custos extras para o cliente no futuro, que eventualmente terá que pagar pelos erros do passado através de solicitações de mudança e elevados gastos com

manutenção. De acordo com (KERZNER, 2009), esse modelo de contratação só é recomendado para situações em que os requisitos são bem conhecidos e estáveis. Caso contrário, esse modelo deve ser evitado.

4.1.2 Contrato de Tempo e Material (*Time & Materials*) com Escopo Fixo

Modelos de contrato que usam *Time and Materials* estão entre os mais comuns atualmente (ARBOGAST et al., 2012). Paga-se um valor fixo por unidade de trabalho, que normalmente é homem-hora. Esse tipo de contrato é ótimo para o fornecedor, que recebe pela quantidade de horas trabalhadas. Há pouquíssimo risco para ele, que, mesmo se atrasar a entrega, sairá ganhando porque serão necessárias mais horas de desenvolvimento. Em contrapartida, o lado do cliente assume um risco gigantesco. O primeiro é não saber precisamente quantas horas de trabalho serão feitas. Outro é ficar dependente do fornecedor, que pode atrasar o desenvolvimento e acabar necessitando de mais horas. Dessa forma, o trabalho tende a se arrastar indefinidamente, sem um prazo para terminar, já que o fornecedor não tem vantagem alguma em terminar cedo o projeto. O escopo fixo define que tudo deve ser desenvolvido, mas o prazo é aberto, e por isso há o perigo de o projeto se estender mais que o esperado.

Apesar desses problemas, o modelo de *Time and Materials* é recomendado principalmente quando há uma forte confiança no trabalho do fornecedor. Há vantagens bastante consideráveis na adoção desse modelo, que envolve principalmente o escopo. Como o pagamento é feito por horas de trabalho, mudanças no escopo não têm um impacto muito grande no projeto. Além disso, é um bom modelo para quando não se sabe medir com precisão a duração do projeto. Por fim, há também o fato de o cliente poder terminar o contrato a qualquer momento.

Como é muito utilizado, este modelo possui diversas variações. Ele também é utilizado com escopo flexível, quando se sabe pouco do escopo no início do projeto, fazendo com que ele seja aprimorado com o passar das iterações (em casos ágeis). Outra forma de utilização é com um custo máximo definido, que reduz drasticamente o risco do cliente, mas, em contrapartida, pode fazer com que o mesmo não receba tudo que deseja do software.

4.1.3 Contrato de Lucro Fixo

Em modelos de contrato que trabalham com lucro fixo (ARBOGAST et al., 2012), o risco é compartilhado. As partes definem no início do projeto um valor fixo para o lucro do fornecedor. Fixa-se, também, o escopo do projeto e parcialmente o prazo do projeto. Caso o projeto termine antes do prazo fixado, o fornecedor vai receber o seu lucro inteiro e o cliente vai deixar de pagar os custos do projeto, sendo vantajoso para os dois lados. No caso de o projeto passar do prazo fixado, o fornecedor recebe também o lucro fixo, mas a partir desse momento, o fornecedor passa a pagar apenas os custos do projeto.

Por exemplo, considere que o cliente paga mensalmente uma parcela de R\$10.000,00, sendo R\$3.000,00 o lucro acordado e 12 meses o prazo do projeto. Se o projeto terminar no mês 10, o cliente só precisará pagar mais R\$6.000,00 (lucro de 2 meses) para o fornecedor. Porém, se o projeto não estiver concluído no mês 12, as parcelas subsequentes que o cliente deve pagar serão de apenas R\$7.000,00 (retirando os R\$3.000,00 de lucro). A partir daí, o fornecedor é forçado a trabalhar sem margem de lucro apenas para concluir o projeto. Essa situação não é interessante para nenhuma parte, e, por isso, considera-se que os riscos são compartilhados nesse modelo de contrato.

Mudanças de escopo nesse tipo de contrato são difíceis, pois podem afetar o prazo fixo acordado e resultar em problemas para os dois lados. Isso acaba ocasionando em um produto que não reflete exatamente o que o cliente desejava.

4.1.4 Contrato com Cláusula de Bônus e Penalidade

Modelos de contrato com cláusulas de bonus e de penalidades (ARBOGAST et al., 2012) são modelos que trabalham ao redor de escopo e prazo fixos, com o custo variável. As partes definem um prazo fixo para o projeto, bem como o escopo a ser desenvolvido. Em seguida, define-se uma cláusula de bônus para o fornecedor, que é em função de quantos dias antecipados o fornecedor entrega o produto. Ou seja, recebe um bônus maior o quanto mais cedo for entregue o produto. Paralelamente, é estipulada uma cláusula de penalidade, que é função dos dias passados após a data de entrega acordada. O fornecedor, nesse caso, recebe menos a cada dia que passa sem entregar o produto.

Esse modelo de contrato transfere a maior parte do risco para o fornecedor, pelo simples fato de o cliente poder dificultar o aceite do produto, caso ele não tenha uma necessidade real de receber o produto cedo ou até mesmo no prazo. Ou seja, se o cliente não

precisar de fato do produto na data acordada, ele pode criar empecilhos, dificultar a comunicação e também o aceite do produto, com o objetivo de atrasar propositalmente e pagar menos ao fornecedor. Para que este contrato funcione bem, é necessário estar bem explícitos os critérios de aceitação e de entrega do produto (LARMAN et al., 2010), para que seja minimizada a subjetividade ao entregar o desenvolvimento.

Novamente, para que esse tipo de contrato funcione, é necessária uma grande confiança. Com um cliente confiável, esse tipo de contrato pode apresentar bons resultados, que premiam as duas partes no caso de entrega antecipada. É um modelo muito interessante para casos em que o cliente tem necessidade real de entrega rápida do produto. Não é uma decisão óbvia para o contratante, que deve estimar quanto monetariamente a sua organização economizaria, ampliaria sua capacidade produtiva ou exploraria uma oportunidade de mercado no tempo obtido com a antecipação do projeto. A partir desse valor, deve-se analisar se realmente há necessidade e ganho de acelerar o projeto e também quanto seria a perda em caso de atraso.

4.2 Modelos de Contratos Ágeis

Modelos de contratos que tratam o escopo de forma **flexível**, facilitando mudanças e estimulando a prática de priorização das histórias, fatiamento do escopo, entrega de valor ao cliente, descarte de funcionalidades pouco importantes e possibilidade de encerramento antecipado do projeto, são considerados Modelos de Contratos Ágeis neste trabalho. Normalmente, esses contratos fixam o prazo ou o valor, mas nunca o escopo. São apresentados nas seções seguintes dez modelos de contrato identificados através da revisão sistemática de literatura, que exemplificam a forma de trabalho dos modelos ágeis de contratação.

4.2.1 Contrato Ágil de Preço Fixo

Esse modelo de contrato (OPELT et al., 2013) é uma adaptação do modelo tradicional de preço fixo, porém para projetos ágeis que têm escopo flexível. O prazo e o custo são fixados, mas o escopo é mantido flexível durante o desenvolvimento. Nesse modelo, o escopo vai sendo refinado com o tempo, ao longo das iterações. Divide-se o macro escopo inicial em iterações para um longo período (ex: 1 ano) com um determinado preço fixo e um tamanho

total, por exemplo em *story points* (COHN, 2004) (perceba que tamanho é diferente de escopo). Ao longo do projeto, as funcionalidades que representam maior valor vão sendo priorizadas e detalhadas para o contratante durante as *sprints*. Usualmente, os requisitos são escritos em forma de histórias do usuário, detalhados de maneira a proporcionar um fácil entendimento para o time de desenvolvimento e para o cliente.

Nesse modelo, uma opção é definir que o fornecedor só recebe quando as histórias são entregues, testadas e aprovadas. Dessa forma, ao longo das *sprints*, os critérios para aprovação da história devem estar bem explícitos. Caso não estejam, pode acontecer muita discussão e impasse com o cliente, que podem dificultar manter o preço e o prazo fixos (HODA et al., 2011).

Apesar do prazo e custo fixos, esse modelo já representa um grande avanço nos modelos de contratação, principalmente pelo fato do escopo tornar-se flexível. O preço é fixado com base num tamanho estimado no início do projeto. O escopo pode mudar à vontade, as mudanças são 'gratuitas', contanto que se mantenham dentro do tamanho estipulado para o projeto. Dessa forma, tornou-se possível realizar entregas periódicas, mudanças e repriorização de histórias, que são boas práticas dos métodos ágeis.

Mesmo com todas essas vantagens, esse modelo não aborda um possível término antecipado do projeto, e não há vantagens explícitas para o fornecedor entregar seu trabalho adiantado.

4.2.2 Contrato de Tempo e Material (*Time & Materials*) com Escopo Flexível

Como foi citado anteriormente, é possível utilizar o modelo de *Time and Materials* para projetos com escopo flexível (ARBOGAST et al., 2012). A ideia é a mesma: realizar o pagamento por unidade de trabalho (homem-hora). Porém, dessa vez, o escopo é flexível, o que significa que mudanças podem ser feitas em relação às histórias acordadas. Na prática, para esse modelo, pouco muda, já que a cobrança ainda é feita com base nas horas trabalhadas. A diferença é que, agora, pode ser solicitada qualquer alteração no desenvolvimento, já que não há acordo de prazo, escopo ou custo.

Apesar dessa liberdade, recomenda-se utilizar, junto com essa modalidade de contrato, uma cláusula que defina um teto de custo para o projeto, de forma que seja possível evitar perder o controle dos seus custos. Esse teto deve ficar claro para ambas as partes e deve ser acordado no início do projeto.

Como no anterior, o risco ainda é todo do cliente, que deve ter confiança no fornecedor para que o contrato crie uma relação amistosa entre as partes. Algumas cláusulas podem ser criadas para mitigar esse risco, como a do teto de custos para o projeto ou para a iteração, mas isso pode acabar levando a consequências, como o recebimento de software diferente do esperado.

Além disso, para prover visibilidade para o cliente e justificar o valor do dinheiro investido, é possível reportar diversas métricas, como gráficos *Burn-down* ou *Cumulative Flowcharts*, que mostram em detalhes o andamento das histórias trabalhadas na iteração. Essa transparência é fundamental entre as partes, visto que é ela que vai construir a confiança para que o projeto se mantenha durante os meses seguintes.

Esse tipo de contrato é recomendado para projetos de pequeno porte ou duração, como pequenos desenvolvimentos, projeto de consultoria, criação de webpages ou de aplicativos mobile de pouca complexidade.

4.2.3 Contrato com Preço Fixo por Unidades de Trabalho

Nesse modelo de contrato, define-se um valor fixo para a unidade de trabalho (LAAKKONEN, 2014). Essa unidade pode ser medida de diversas formas, como *Story Points* (pontos de história), PCU (pontos de caso de uso) e PF (pontos de função). A unidade de trabalho recomendada é pontos de função. A justificativa para essa escolha está no fato de que pontos de função têm mais regras e padrões, e isso facilita a precificação de mudanças. Infelizmente, PF não calcula esforço, diferentemente dos *Story Points*, que são medidas de esforço.

É feita uma estimativa inicial do trabalho, e são priorizadas as mais importantes para as primeiras iterações. Essas estimativas vão sendo refinadas com o passar do tempo, e as mudanças vão sendo incorporadas através de alterações nos pontos de função, o que reflete claramente o impacto da mudança em termos financeiros. Nessa modalidade de contratação, o pagamento é feito somente quando as unidades de trabalho são entregues, testadas e aprovadas, ao final de cada iteração. Dessa forma, o valor recebido é uma função direta da quantidade de pontos entregues na iteração. Novamente, ressalta-se a importância de ter critérios bem claros para facilitar o acordo entre as partes de quando a funcionalidade está realmente pronta. Aqui também é importante citar que a confiança no cliente se faz necessária.

Para que seja calculado o valor de cada ponto, algumas formas podem ser utilizadas. Uma delas é recuperar o histórico de estimativas de projetos anteriores e usá-lo como base para estimar os projetos seguintes, o que dá um valor bem realista e condizente com a forma de trabalho e restrições da organização. Outra possibilidade, para o caso de não haver um histórico de projetos prévios, é parametrizar um valor estimado, que vai sendo refinado a cada iteração concluída.

4.2.4 Contrato “Money for Nothing, Change for Free”

Money for Nothing, Change for Free (SUTHERLAND, 2008) é um modelo de contrato ágil proposto por Jeff Sutherland em 2008, e apresenta características que favorecem fortemente a adoção de práticas ágeis. Nesse modelo, o escopo é flexível, mas o tamanho é fixo. As partes devem acordar inicialmente o prazo final do projeto, o tamanho total estimado e o custo do projeto. Esse tamanho normalmente é definido em *Story Points* representando uma visão bem alto nível dos requisitos, que vão sendo aprimoradas no decorrer das sprints. O próprio nome do modelo define duas características bem marcantes:

Change for Free: Mudanças são bem-vindas a qualquer momento. Só há uma restrição quanto às mudanças solicitadas. Para entrar uma funcionalidade nova, algo de mesmo tamanho deve sair. Com isso, mantém-se o prazo e o custo previamente estipulados. Daí vem o nome: *change for free*, ou seja, mude à vontade. Resumindo, as mudanças são liberadas, contanto que não afetem o tamanho acordado.

Money for Nothing: Em projetos de escopo fechado, tudo que foi acordado deve ser desenvolvido. No caso do *Money for Nothing*, o escopo é flexível. Isso significa que podem ser removidas (ou adicionadas) funcionalidades no escopo. O detalhe é que Sutherland afirma que se o escopo for bem modularizado e priorizado corretamente, as funcionalidades que agregam mais valor vão ser implementadas nos meses iniciais de desenvolvimento, deixando as funcionalidades menos importantes para o final. No contrato idealizado por Sutherland, há uma cláusula de rescisão que define que o contrato pode ser encerrado no início de qualquer mês (ou *Sprint*, ou iteração, depende de como o projeto for modularizado) do projeto (representado na **Figura 8**), e a organização contratante paga, por exemplo, 20% (esse valor foi tomado como exemplo para representar a margem de lucro do contrato) do que pagaria nos meses restantes. Dessa forma, o escopo foi reduzido, e foi vantajoso para o contratante, que só pagou 20% do que pagaria nos meses restantes, foi vantajoso para o contratado, que

recebeu sua margem de lucro e teve seus recursos liberados com antecedência, e ainda foi vantajoso para o usuário final, que vai receber um software mais simples e fácil de usar e sem funcionalidades de pouca utilidade. A única área que possivelmente vai sair perdendo nesse modelo de contrato é a área comercial do fornecedor, que vai ter que trabalhar mais para trazer *leads* (possíveis clientes) para substituir os projetos finalizados com antecedência e manter a equipe com trabalho.

Para que esse contrato tenha sucesso, é importante que as características ágeis sejam respeitadas, ou até mesmo incluídas como cláusulas no contrato. Por exemplo, nesse modelo de contrato é fundamental que as entregas sejam feitas de forma iterativa. Outro exemplo é a necessidade da participação ativa do cliente no decorrer do projeto, para garantir a priorização e estimativa otimizada das histórias definidas.

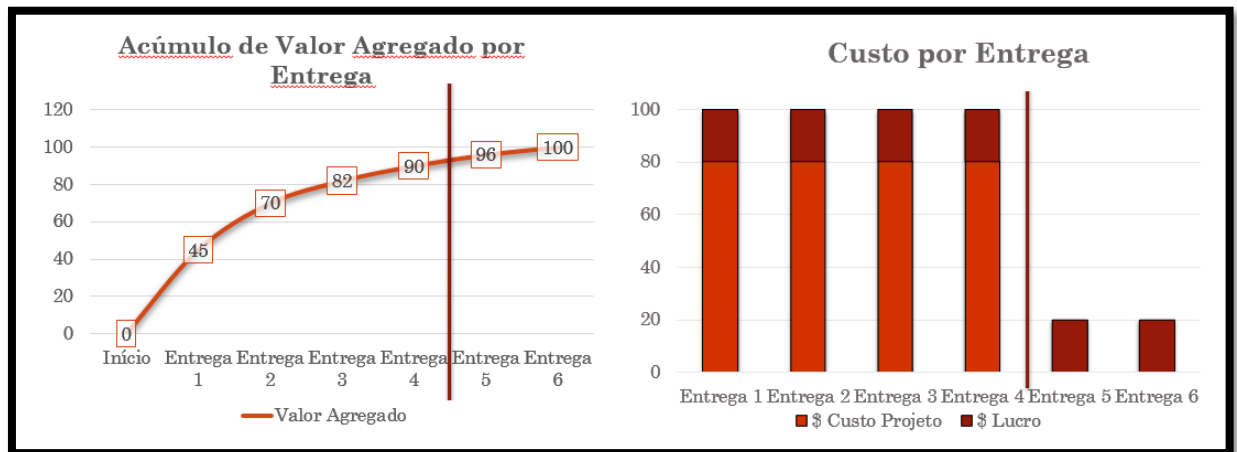


Figura 8: Acúmulo de Valor Agregado por Entrega x Tempo & Custo por Entrega do *Money For Nothing, Change for Free*

4.2.5 Contrato Progressivo

O modelo de contrato progressivo (ARBOGAST et al., 2012) (GILB, 2007) é um modelo usado comumente em acordos de serviços e contratos guarda-chuva (que englobam vários pequenos serviços). Nele, o escopo é totalmente flexível, assim como o preço e a duração do projeto. Inicialmente, as partes entram em acordo sobre a forma de pagamento e precificação, podendo ser, por exemplo, preço fixo por iteração, *Time & Materials* ou preço fixo por unidade de trabalho entregue (*story point*). O que caracteriza esse modelo é que ele força a cooperação do cliente, que é fundamental para o bom desempenho de projetos ágeis. Através da obrigação de recomprometimento com as histórias e com o contrato a cada fim de iteração, o cliente e o fornecedor têm uma forma de constantemente validar o que foi

entregue, definir novos objetivos e testes de aceitação para a próxima iteração, além de priorizar as histórias que serão desenvolvidas em sequência.

As duas partes têm interesse em fazer um bom trabalho com esse modelo. Ao final de cada iteração, o cliente paga o fornecedor e, nesse momento, ele tem a escolha de renovar o contrato ou cancelá-lo imediatamente. O que vai suportar essa decisão vai ser o trabalho do fornecedor, que tem que mostrar constantemente que está entregando algo de valor e funcionando, além de demonstrar que ainda há histórias que agregam valor e valem a pena ser desenvolvidas. Caso o cliente julgue que o que será entregue nas iterações seguintes não agrega valor suficiente para cobrir o custo do projeto, ou mesmo se ele estiver insatisfeito com o trabalho do fornecedor, ele pode cancelar o contrato. O mesmo vale para o fornecedor, que também tem a autoridade para encerrar o contrato no final das iterações caso não esteja gostando de trabalhar com esse cliente.

4.2.6 Contrato “No Cure, No Pay”

O modelo de contrato *No Cure, No Pay* foi proposto por Tom Gilb (GILB, 2007), e teve como sua maior motivação o fato de organizações gastarem muito dinheiro com projetos que fracassam:

“A maioria dos fracassos dos projetos foi totalmente paga. Nós não apenas pagamos bem pelo fracasso, e sim quanto maior o fracasso (em termos de tamanho e duração do projeto), mais as pessoas são pagas!” (GILB, 2007)

Com essa motivação, a proposta de Gilb em seu contrato parte de uma premissa simples, mas que tem desdobramentos relevantes: pagar apenas quando os resultados são realmente entregues. Boehm, em 2004, já sugeria condicionar o valor do pagamento de cada entrega ao valor de negócio que esta proporciona ao cliente (BOEHM, 2004). Isso implica que o cliente tem a opção de simplesmente não pagar caso não esteja satisfeito com o resultado demonstrado. Entretanto, se esse for o caso, o cliente não tem o direito de ficar com o produto, tendo que devolver ao fornecedor. Por isso, é fundamental estar bem definidos os critérios de aceitação (LARMAN et al, 2010), testes de aceitação ou o que for utilizado para aprovar a entrega. Minimizar a subjetividade nesse momento é um dos pontos importantes para o sucesso desse modelo (ARBOGAST et al., 2012).

Nesse relacionamento, o risco do cliente é muito reduzido, já que ele pode optar por não pagar. Porém, o contratante tem um risco bem alto. Para compensar esse risco, é

adicionada uma cláusula que, a cada vez o cliente não aceitar o produto, o preço aumenta. Dessa forma, fica menos interessante para o cliente rejeitar as entregas, e o motiva a especificar bem as histórias e manter um bom relacionamento com a equipe de desenvolvimento.

Esse contrato também requer forte confiança nas duas partes. Principalmente no cliente, que, por estar exposto a menos riscos, pode não colaborar para o sucesso do projeto através de, por exemplo, atrasos em respostas de emails, deixar de participar de decisões da iteração relacionadas às priorizações e faltando a reuniões importantes. O cliente teoricamente não tem nada a perder, então pode dificultar a relação com o fornecedor.

A forma de pagamento desse contrato é, normalmente, estipulada em *Time & Materials (T&M)* e paga a cada entrega. A conta da iteração é gerada após a aprovação da entrega pelo cliente, mas o pagamento só é liberado após 30 dias de validação no ambiente do cliente. Se for encontrado algum problema durante esse período, o pagamento é bloqueado e fica retido até que a correção seja realizada.

Esse contrato tem duração indefinida, e pode ser estendido como bem entenderem as duas partes. O escopo também é flexível. Como o pagamento é feito em T&M, mudanças não são um problema nesse tipo de contrato, e são bem aceitas tanto pelo contratante quanto pelo fornecedor.

4.2.7 Contrato com Objetivo de Custo

A principal característica dessa modalidade de contrato (ECKFELDT, 2005) é a distribuição igualitária dos riscos do projeto. Inicialmente, identifica-se e estima-se, em colaboração entre cliente e fornecedor, o escopo do desenvolvimento de forma genérica. Em seguida, definem-se diretrizes e custos para alterações de escopo no decorrer das iterações, pois podem ocorrer mudanças que não alteram o Objetivo de Custo do projeto, mas também podem surgir mudanças que o afetem. O próximo passo é estabelecer um Objetivo de Custo, juntamente com um Objetivo de Lucro (% do objetivo de custo) e uma estimativa de duração do projeto. Esse Objetivo de Custo então é dividido em custo estimado por iteração, derivando um lucro estimado por iteração (ARBOGAST et al., 2012)

Durante as iterações, é importante garantir que todos os detalhes e resultados são compartilhados com o cliente, para dar transparência sobre todo o cálculo do custo real e sobre o andamento do projeto. Como foi citado anteriormente, o aspecto chave desse

contrato é o risco compartilhado (POPPENDIECK et al., 2003). Se o projeto foi bem, e gastou menos do que o custo esperado para a iteração, o cliente e o fornecedor dividem o lucro adicional. Entretanto, se o projeto não caminhou conforme esperado e acabou a iteração ultrapassando o Objetivo de Custo do período, as duas partes repartem o prejuízo, reduzindo parcialmente o lucro do fornecedor e aumentando os gastos do cliente. Dessa forma, os dois têm interesse na entrega antecipada.

Uma das desvantagens desse modelo é que mudanças no escopo, apesar de aceitáveis, não são tão facilmente inseridas no projeto. Como há um Objetivo de Custo a ser atingido em cada iteração, é necessária uma análise de cada mudança para verificar se ela impacta no Objetivo de Custo acordado inicialmente. Para isso, recomenda-se definir três tipos de mudança possíveis: Correção, que é um ajuste em uma funcionalidade já implementada por motivo de desentendimento do que deveria ter sido feito; Clarificação, que é uma modificação pontual em uma funcionalidade implementada corretamente, mas que pode receber algum ajuste; e Melhoria, que contempla uma nova funcionalidade não implementada. Das três citadas, a única que requer um ajuste no Objetivo de Custo (e consequentemente no Objetivo de Lucro) é a melhoria, que adiciona novas funcionalidades no sistema. É interessante observar que mudanças relacionadas a trocas de funcionalidades também são permitidas, já que não afetam o Objetivo de Custo do projeto.

4.2.8 Outros Contratos

Além dos contratos citados, diversas variações vêm surgindo para melhor assentar as metodologias ágeis nos contratos firmados e melhorar a relação entre cliente e fornecedor que trabalha com metodologias ágeis. O problema ainda existe, e pesquisadores e organizações buscam novas soluções de contratos ágeis.

Uma proposta sugerida pela empresa alemã *Adesso AG* é um modelo de contrato chamado *adVANTAGE*, (BOOK et al., 2012) que combina elementos de contratos de preço fixo com elementos de *T&M*. Ele tenta mesclar a ideia de conhecimento prévio do escopo genérico do projeto para os desenvolvedores, da forma como teriam em um projeto de preço fixo, porém sem o risco de estar comprometido a entregar tudo, com a ideia de que o cliente vai pagar exatamente pelo que será entregue, similar a uma situação de *T&M*, mas sem expô-lo ao risco do projeto descontrolar e ficar interminável. A precificação e pagamento são feitos

com base nos elementos do processo ágil, e é realizada por história desenvolvida, no final de cada iteração.

Outro modelo de contrato, também desenvolvido por uma empresa, é o *Flexible Contract* (FLEXIBLE, 2013). Ele consiste de entregas modulares, focadas em entregar valor rápido ao cliente. O mesmo mescla o pagamento em *T&M* com pagamento no momento da entrega dos módulos, e incentiva o contratado a entregar rápido com cláusulas que beneficiam essa prática. O pagamento nas entregas é relativo ao valor agregado de cada módulo. Para definição do valor, são usadas métricas que são rigorosamente definidas e coletadas com frequência.

Por fim, há um modelo criado por outra empresa (que preferiu não se identificar) que eleva as entregas e planejamentos periódicos a outro nível. Semanalmente, o cliente deve participar de uma reunião de priorização e *review* da iteração. Nesse mesmo momento, é feita uma entrega de software funcionando e a priorização do escopo da próxima semana. O contrato é renovado semana a semana, e pode ser cancelado a qualquer momento por ambas as partes. O tempo máximo de duração do projeto é de doze semanas.

5 Características de Contratos de Desenvolvimento de Software

Neste capítulo, o enfoque está nas características dos modelos de contratos. Atributos como, por exemplo, forma de precificação, período de vigência do contrato e método de estimativa estão presentes na maioria dos modelos. A partir da revisão sistemática de literatura (explicada na seção 2.1) relacionada a contratações de desenvolvimento de software e métodos ágeis, além da análise de dezenas de modelos de contratos, foi possível identificar algumas características marcantes e comuns em alguns tipos de contratos. Algumas delas serão aqui tratadas como características diretas, e outras como características indiretas. As **diretas** são inseridas através de cláusulas propositalmente, enquanto as **indiretas** são derivadas de um conjunto de cláusulas, podendo ser vistas como consequências das outras características diretas. Foram identificadas, então, 7 características diretas e 3 indiretas. Mais detalhes sobre cada uma delas estão descritos abaixo.

5.1 Precificação

Quando se fala em precificação, que para este trabalho é considerada uma característica direta, automaticamente outros aspectos como mudança e estimativa de tempo do projeto são acionados para serem levados em conta. Em uma análise superficial, o preço de um projeto pode ser definido como o custo do mesmo para o fornecedor, mais eventual margem de lucro. Assim, como as demais características detalhadas abaixo, pode-se variar entre abordagens mais tradicionais, como Preço Fixo e Tempo & Material, até abordagens mais modernas, como pagamento a cada iteração, história ou até funcionalidade entregue.

O aprendizado na precificação é: controlar os custos, aprender com os projetos passados e suas estimativas e combinar ferramentas tradicionais com adaptações ágeis para melhor entendimento da saúde financeira do projeto. Esse é o primeiro passo para começar a precificar de forma eficiente.

Um importante fator a ser considerando nas contratações por organizações públicas, sobretudo as regidas pela lei 8.666/93, é que o modelo de menor preço tende a gerar contratações inadequadas para ambas as partes quando não são observadas algumas características do projeto. A própria dinâmica do processo de contratação, majoritariamente

por meio de pregão eletrônico do tipo leilão reverso, incentiva a área comercial dos fornecedores a propor preço que deixa a margem para operação comprometida, podendo gerar um desequilíbrio financeiro e, conseqüentemente, uma futura ruptura contratual. A exequibilidade do preço é fator crítico para ambos, não sendo trivial sua aferição por conta dos inúmeros aspectos nos requisitos do contrato e/ou software a ser desenvolvido, que influenciam diretamente no custo e no preço da prestação de serviço.

As organizações privadas, que não precisam observar a rigidez da legislação acima citada, embora muitas vezes sendo regulada por normas e/ou legislação setorial que impõem certa conduta, têm maior flexibilidade para explorar soluções de precificação por meio de modelos de contratos mais ágeis e, conseqüentemente, conseguirem uma melhor relação nesse importante fator de contratação.

5.2 Vigência do Contrato

Cláusulas e regras relacionadas à vigência do contrato são itens que devem estar presentes – e normalmente estão – em todos os contratos. São características diretas que consistem na definição explícita do período de duração do contrato e das cláusulas de rescisão do mesmo. Pode parecer um aspecto sem muito valor e que não impacta decisivamente no andamento do trabalho, mas é exatamente o contrário (LARMAN et al., 2010). Como veremos, características sobre vigência e duração do contrato podem definir diretamente toda a dinâmica da relação.

É comum pensar que rescindir o contrato é algo negativo. Na maioria dos contratos a rescisão se dá apenas quando alguém não cumpre algum acordo no contrato. E, de fato, esse tipo de regra precisa existir. No entanto, será que é possível definir cláusulas de rescisão de contrato para o caso inverso, quando tudo caminha muito bem e sem problemas? Parece improvável, mas a resposta é sim.

Alguns contratos que envolvem métodos ágeis já abordam esse novo pensamento de rescisão de contrato. Nesse caso, podem definir cláusulas permitindo, por exemplo, que qualquer parte (ou um consenso entre as duas) rescinda o contrato ao final de qualquer iteração, desde que, caso parta do cliente, seja paga uma taxa relativa ao lucro que o fornecedor teria nos meses de projeto restantes. A princípio, não parece uma cláusula interessante. No entanto, ela se torna viável devido às outras práticas ágeis inseridas no contrato, como entregas contínuas e que contenham sempre o maior valor agregado possível

para o cliente. Dessa forma, com o cliente recebendo sempre as funcionalidades de maior valor, é possível que chegue um momento que as funcionalidades restantes não compensem o custo de desenvolvimento das próximas iterações, e o cliente assim pode optar por rescindir o contrato. Essa rescisão é vantajosa para o cliente, que vai deixar de pagar o custo de desenvolvimento de iterações que não trazem um valor agregado considerável, e é vantajosa para o fornecedor, que vai receber sua parcela de lucro normalmente e ainda vai ter seus recursos disponibilizados mais cedo para alocar em novos projetos (BOEHM, 2006).

5.3 Mudança de Escopo

Uma verdade que todo envolvido em projetos deve considerar: Os requisitos vão mudar. O grande esforço de planejamento, com tentativa de restringir as mudanças e prever aquilo que é imponderável tem pouco ou nenhum valor para garantir o sucesso do projeto.

As abordagens mais tradicionais exigem um grande esforço para controlar a mudança, determinando comitês de controle de mudanças com responsabilidade de aprovar ou rejeitar uma solicitação, como se fosse possível entregar algo de valor sem considerar que a mudança já ocorreu e o projeto deve se adequar a ela.

Por outro lado, o gerenciamento com métodos ágeis preconiza que a mudança é inerente aos projetos, acolhendo as incertezas e criatividade para gerar valor e entregando o que realmente vai fazer diferença para o negócio.

Ao falar de contratos, sabe-se que mudanças – que é considerada para esse trabalho uma característica direta – quase que invariavelmente vão consumir recursos diversos e que o impacto no custo e/ou prazo poderá ser considerável. Uma abordagem que pode trazer um equilíbrio é não estabelecer um valor monetário pela mudança, mas sim em tamanho funcional. Com as funcionalidades do projeto estimadas em alguma unidade de medida (*story points*, pontos de função, etc) é possível, em caso de mudança, consumir do tamanho total estimado e não afetar o custo do projeto ao remover funcionalidades com baixa prioridade, agregando em mudanças que sabidamente trarão valor ao produto. O esforço de se eliminar funcionalidade com baixo ou nenhum valor para o projeto é amparado por um estudo que mostra que 45% das funcionalidades nunca são utilizadas e que 19% são usadas raramente (Figura 9) (STANDISH GROUP, 2002).

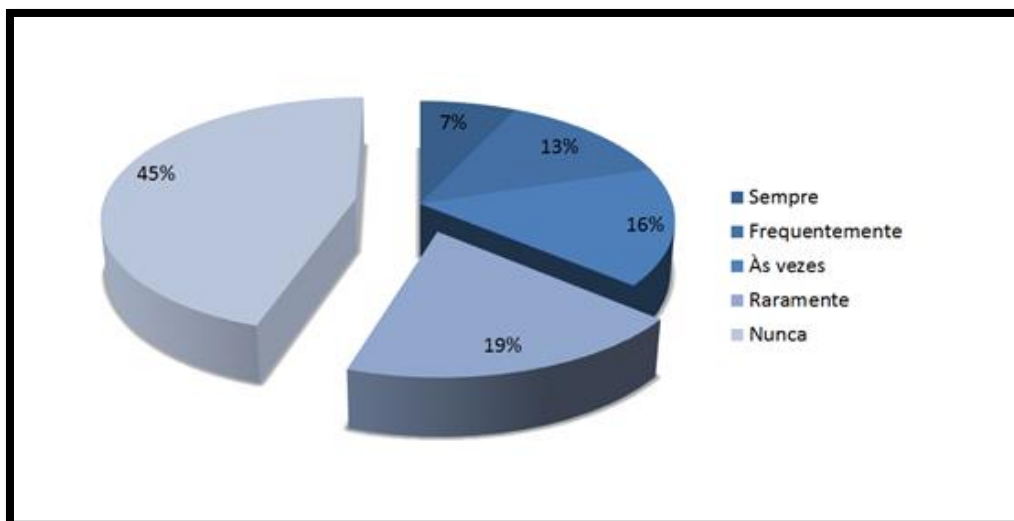


Figura 9: Média de uso de funcionalidade de sistemas (STANDISH GROUP, 2002)

5.4 Garantia

Ao abordar a questão de garantia, percebemos que o tema é mais comum na esfera pública, vide leis como a 9.609/98 artigo 7º, que aborda explicitamente garantia e não estabelece um prazo para a validade técnica do software quanto aos vícios. Ainda há divergências no entendimento de que o projeto de desenvolvimento de software estaria alinhado a esta lei. Deve haver prestação de garantia com depósito e retenção até que a garantia do produto expire. Após isso, o valor é devolvido ao fornecedor.

Para melhor endereçamento da questão em âmbito privado, é recomendado que a garantia seja explícita (direta) no contrato, independentemente do tamanho e/ou complexidade do produto. Uma prática recomendada no gerenciamento de escopo (PMI, 2004) é aceitar formalmente uma entrega e no gerenciamento de integração, encerrar o projeto ou fase. Esse é um bom momento para indicar, nos documentos contratuais, o início da garantia técnica.

A duração da garantia deve considerar diversos aspectos, entre eles, o custo e o benefício de ampliar esse período. Quanto maior o período de garantia, maior o tempo de disponibilidade do fornecedor e, consequentemente, maior o valor cobrado pelo projeto. Por outro lado, em se tratando de desenvolvimento de software, questões relacionadas a arquitetura da solução podem ficar ocultas por muito tempo se o serviço tiver uma elevação gradual da solicitação por recursos, por exemplo, números de usuários simultâneos. Uma vez identificadas que características do negócio indicam que tais problemas podem não ocorrer

nos primeiros meses, deve-se tentar um entendimento do prazo mais adequado para o período de garantia.

5.5 Envolvimento do Cliente

O envolvimento do cliente é menosprezado nos modelos tradicionais de contratação de desenvolvimento de software. No entanto, em contraste a isso, os modelos ágeis reforçam a necessidade da participação constante do cliente no decorrer do projeto. A necessidade de maior atenção a esse aspecto fica evidente na pesquisa realizada nas seções seguintes. Resta investigar o motivo pelo qual essa peça fundamental nos projetos não é mais bem explorada nas relações contratuais.

Era comum, nos modelos tradicionais, o envolvimento do cliente se dar apenas em momentos pontuais do projeto. Normalmente, no início do projeto, para o levantamento e especificação de requisitos, e no final, para homologação e validação do sistema produzido. Esporadicamente poderia ser solicitada alguma aprovação ou revisão no meio do projeto, mas essa prática não era muito comum nesse tipo de projeto. E é justamente essa falta de envolvimento um dos principais causadores de retrabalho, atraso e aumento no custo do projeto, uma vez que, ao final do projeto, o cliente percebe que o que foi desenvolvido não era exatamente o que queria, e, no melhor dos casos, percebe melhorias e modificações que poderiam ter sido sugeridas no início do projeto, se houvesse um maior envolvimento de sua parte.

Sabendo disso, os métodos ágeis reforçam essa necessidade de envolvimento, e, por isso, torna-se interessante a criação de cláusulas que aumentem a participação do cliente no decorrer do projeto, o que torna esse aspecto uma característica direta. Por exemplo, uma cláusula que garanta a presença de um representante do cliente pelo menos em uma reunião quinzenal com o fornecedor já pode significar uma redução de problemas no futuro.

5.6 Método de Estimativa

Estimativas estão na essência de todo projeto. Desde um pequeno reparo residencial até projetos envolvendo dezenas de milhares de pessoas e quantias enormes de recursos financeiros, sempre haverá a necessidade de estimar. As questões mais comuns são: quanto vai custar e em quanto tempo ficará pronto? Isso tudo alinhado ao que será feito (escopo). Durante muito tempo foi encarado em projetos como tríplice restrição, onde características

de uma área influenciam diretamente nas características de outra. Aqui trataremos de questões de prazo, visto que a precificação, anteriormente abordada, trouxe um ponto inicial de debate sobre o tema custo.

Desde estimativas mais tradicionais, como estimativa paramétrica e APF (Análise de Ponto de Função), até abordagens mais ágeis como *T-shirt sizing* (classificação em P, M e G) e Pôquer do Planejamento (onde o time de desenvolvimento classifica as histórias de acordo com a sequência de Fibonacci), o tema é recursivo em todo debate sobre a gestão do tempo em projetos. Dessa forma, como se pode perceber, o método de estimativa é visto como uma característica direta no âmbito desse trabalho.

É sabido que praticamente todos os contratos, em algum momento, abordarão a questão de tempo. Prazo para uma entrega parcial, um protótipo, uma funcionalidade ou até mesmo a conclusão do projeto. Os métodos de estimativa apresentados – além de outros que não foram citados – são ferramentas interessantes que podem se alinhar a todos os modelos de organização. No entanto, a cultura, os processos e ambiente dessas organizações são fatores determinantes na escolha de uma dessas técnicas.

As abordagens alinhadas a métodos ágeis têm mostrado grande valor em projetos de desenvolvimento de software. Há relatos de aumento na ordem de 400% na velocidade de entregas de pontos por *Sprint* (SUTHERLAND, 2014). Uma vez determinado o valor que cada funcionalidade do projeto traz para o negócio, a equipe decide a forma como o trabalho será estimado, dividido e planejado. Essa forma precisa ser estimável, como definido no amplamente aceito modelo INVEST (Independente, **N**egociável, **V**aliosa, Estimável, *S*mall [Pequena] e **T**estável). Uma vez que o trabalho é conhecido e estimado usando uma das abordagens citadas, finalmente a questão sobre quando determinada entrega será feita pode ser respondida, restando conhecer, a partir de dados históricos, a velocidade/produtividade das equipes envolvidas. Nesse momento, a estimativa análoga é utilizada de forma empírica pelo próprio time.

5.7 Risco para o Cliente

O risco para o cliente é uma característica indireta que surge a partir das demais cláusulas definidas no contrato. A inclusão de certas cláusulas pode aumentar o nível de risco para o cliente (contratante). Por exemplo, ao usar modelos de precificação Tempo & Material (*T&M*), o nível de risco para o contratante certamente sobe, visto que não é possível prever

com exatidão o valor a ser pago no final do mês e nem uma previsão para conclusão do projeto. Outro exemplo é o baixo envolvimento do cliente do projeto. Com pouca participação do contratante, o risco para o mesmo é aumentado devido à altíssima probabilidade de alterações no final do projeto, que acarretam em aumento de custos e prazos.

Em resumo, todos os aspectos que puderem incorrer diretamente em aumento de custos para o contratante, aumentar a chance de atrasos nas entregas, reduzir a qualidade do software ou algum outro impacto negativo para o cliente, podem ser considerados como aspectos do contrato que aumentam o risco para o cliente.

5.8 Risco para o Fornecedor

Por outro lado, o nível de risco assumido pelo fornecedor (contratado), que também é um aspecto indireto derivado das outras cláusulas do contrato, é outro fator que deve ser considerado. Dentre os exemplos de ações que podem representar um aumento do nível de risco para o fornecedor, pode-se citar o modelo de precificação com pagamento após a entrega do sistema, a exigência de longos períodos de garantia, modelos de rescisão de contrato unilaterais sem multa associada, dentre outros.

É possível concluir, portanto, que todos os aspectos dos contratos que possam criar a possibilidade de o fornecedor não ser pago, fazê-lo gastar mais recursos que o planejado, obrigá-lo a reduzir sua margem de lucro, gerar atrasos no pagamento ou mesmo impactar diretamente em seus processos e padrões de trabalho podem representar um aumento no risco para o fornecedor.

5.9 Popularidade

A popularidade de um tipo de contratação, ou seja, o seu nível de adoção no mercado atual, pode parecer algo irrelevante. No entanto, para muitas organizações essa característica pode influenciar sua adesão ao modelo em questão. Essa característica também é indireta, visto que não é algo que se pode alterar diretamente através de cláusulas no contrato.

Em organizações mais tradicionais, governamentais ou, ainda, aquelas com maior aversão ao risco, as contratações tendem a optar pelos modelos mais consagrados, usados em larga escala, mesmo que os resultados destes contratos não sejam necessariamente os melhores. Nesse tipo de relação contratual, os modelos mais populares são os de Preço Fixo

(garantido ou com remuneração de incentivo ou com ajuste econômico de preço) e Tempo e Material (T&M) (PMI, 2013).

Por outro lado, organizações com maior foco na inovação e empresas privadas de pequeno/médio porte, que têm uma maior aceitação em uso de tecnologias e práticas modernas, são candidatas aos modelos de contratos ágeis.

Embora alguns modelos de contratos modernos apresentem situações inegáveis de ganha-ganha, com redução de risco para ambas as partes, tais modelos enfrentam resistências relacionadas a aspectos culturais e um arcabouço jurídico que dá sustentação às relações contratuais. O ‘medo do novo’ e de se responsabilizar por mudar uma cultura que atravessa os anos acaba sendo um fator determinante para organizações se manterem apegadas aos modelos tradicionais. Poucos têm a coragem de assumir a responsabilidade de mudar essa cultura: se problemas acontecerem no contrato tradicional, não há culpa, porque os problemas já eram previstos, visto que a adoção do modelo é comum e recorrente. Entretanto, se ocorrerem problemas ao utilizar um contrato novo, a culpa provavelmente vai cair em quem trouxe a novidade.

Em tese, a estrutura legal dos contratos ágeis é a mesma dos modelos tradicionais. A grande diferença é a abordagem utilizada, o entendimento do processo ágil, suas entregas e consequentemente, como detectar e inserir esse contexto nos contratos.

Por último, mas não menos importante, os contratos refletem desejos e, sobretudo, medos em relação ao futuro numa relação entre organizações e pessoas. Um projeto de sucesso não será garantido com um bom contrato, mas em relações de colaboração, transparência e confiança. À medida que a confiança entre cliente e fornecedor cresce, o modelo comercial totalmente sustentado em contratos enfraquece, e, dessa forma, se valida a citação do início do trabalho que embasou e motivou essa pesquisa: “Colaboração com Cliente MAIS QUE Negociação de Contratos” (BECK et al., 2001).

5.10 Vantagem para Usuário Final

A característica mais importante numa relação para entrega de valor deveria ser essa, ou seja, o usuário final perceber uma vantagem no produto ou serviço adquirido. Aspectos como poder usar o software o mais rápido possível, ter liberadas as funcionalidades mais importantes primeiro, usar um sistema mais enxuto e com melhor usabilidade buscam

garantir essa percepção de uma relação vantajosa, que não deve significar desvantajosa para as outras partes – cliente e fornecedor.

Essa característica pode ser considerada indireta, pois a construção dessa vantagem é feita pelo conjunto das outras características aqui abordadas, como as mudanças, métodos de estimativa e precificação e assim por diante.

6 Pesquisa sobre as Características mais Importantes

No capítulo anterior, foram apresentadas algumas características dos modelos de contratos. Foram explicados detalhes sobre dez atributos presentes em contratos de desenvolvimento de software, como, por exemplo, forma de precificação, período de vigência do contrato, método de estimativa e riscos para ambas as partes. Mas será que eles são realmente os atributos mais importantes em um modelo de contrato? Quais serão as características mais importantes a se tratar em um modelo de contrato para desenvolvimento de software? Para responder essa pergunta, foi realizada uma pesquisa utilizando a metodologia de questionário de pesquisa citada na seção 2.2, pensando em, uma vez identificadas quais são as características consideradas mais relevantes, ajudar pesquisadores e empresas a escreverem contratos mais modernos e representativos, contendo regras e cláusulas que sejam de interesse de todas as partes envolvidas na contratação. São apresentados nas seções seguintes os resultados da pesquisa executada, mostrando graficamente a relevância para cada setor e perfil entrevistado. Em seguida, é feita uma análise dos dados obtidos e são explicadas as conclusões obtidas com o questionário.

6.1 Resultados da Pesquisa

Após a execução da pesquisa (divulgação do questionário) durante duas semanas, foram obtidas 32 respostas no *Dot Voting* criado e no formulário de pesquisa no Google Forms, sendo das 32:

- 17 de empresas públicas e 15 de empresas privadas.
- 18 trabalham do lado fornecedor e 14 do lado cliente.

O volume de resposta foi considerado muito satisfatório, visto que os entrevistados estão diretamente envolvidos em contratação de desenvolvimento de software. O perfil dos respondentes é gerencial, executivo ou comercial, e todos são de empresas de TI. Todos tem conhecimento sobre contratação de desenvolvimento de software e todos conhecem (no mínimo superficialmente) métodos ágeis.

Os resultados foram analisados de todos os pontos de vista: com relação à relevância das características dos contratos para o setor público, para o setor privado, para o lado fornecedor, para o lado contratante e uma visão geral das respostas. Abaixo é feita uma

demonstração específica para cada resultado obtido, mostrando o questionamento feito para identificar se restrições contratuais de fato afetam o trabalho das equipes nos projetos. Porém, antes é apresentado um questionamento realizado no início da pesquisa com o objetivo de validar o problema pesquisado.

6.1.1 Você já Teve seu Trabalho Limitado/Prejudicado por Restrições Contratuais?

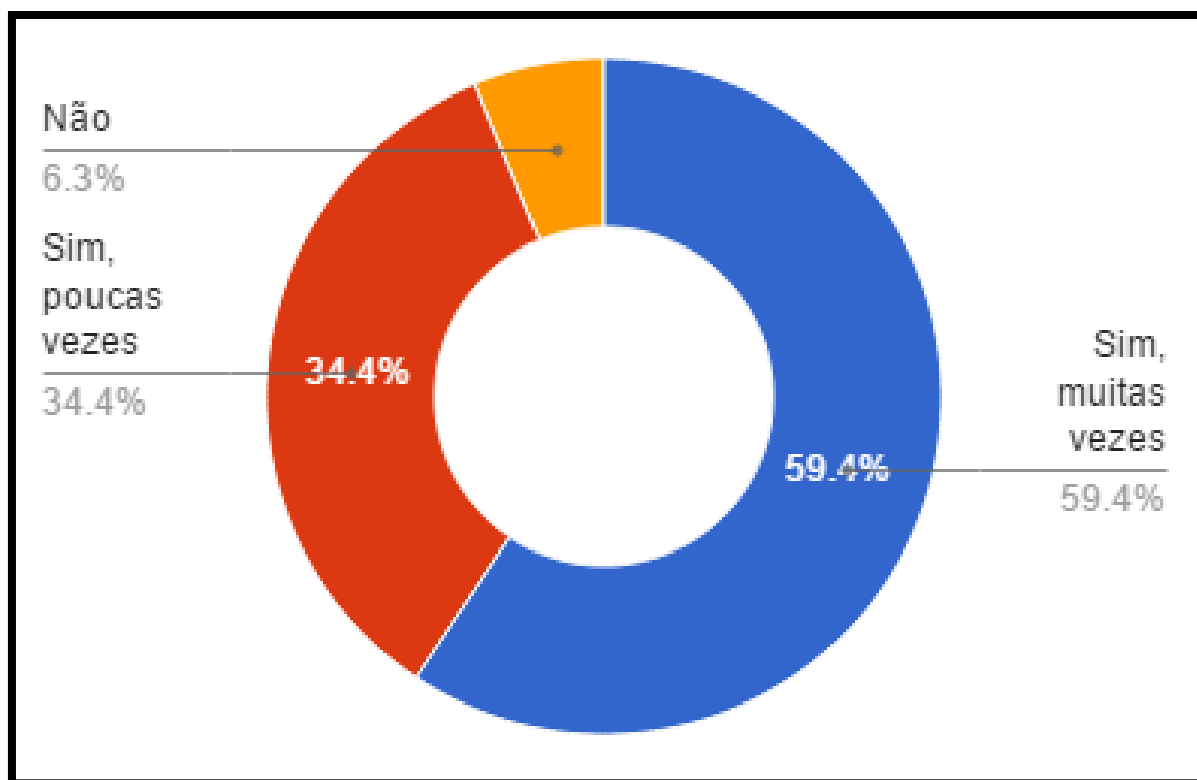


Figura 10: Gráfico sobre limitações/restrições contratuais

Como esperado, a resposta confirmou as expectativas. Dos 32 entrevistados, 19 (59,4%) afirmaram já ter passado por essa experiência muitas vezes, 11 (34,4%) disseram que já passaram, porém poucas vezes, enquanto apenas 2 (6,3%) não foram afetados por restrições contratuais (**Figura 10**).

Esse resultado confirma a importância de um contrato bem escrito e adequado à realidade da organização, já que quase 94% dos gestores e funcionários envolvidos em contratação entrevistados já foram afetados por causa de contratos mal escritos ou inadequados.

6.1.2 Relevância Total das Características

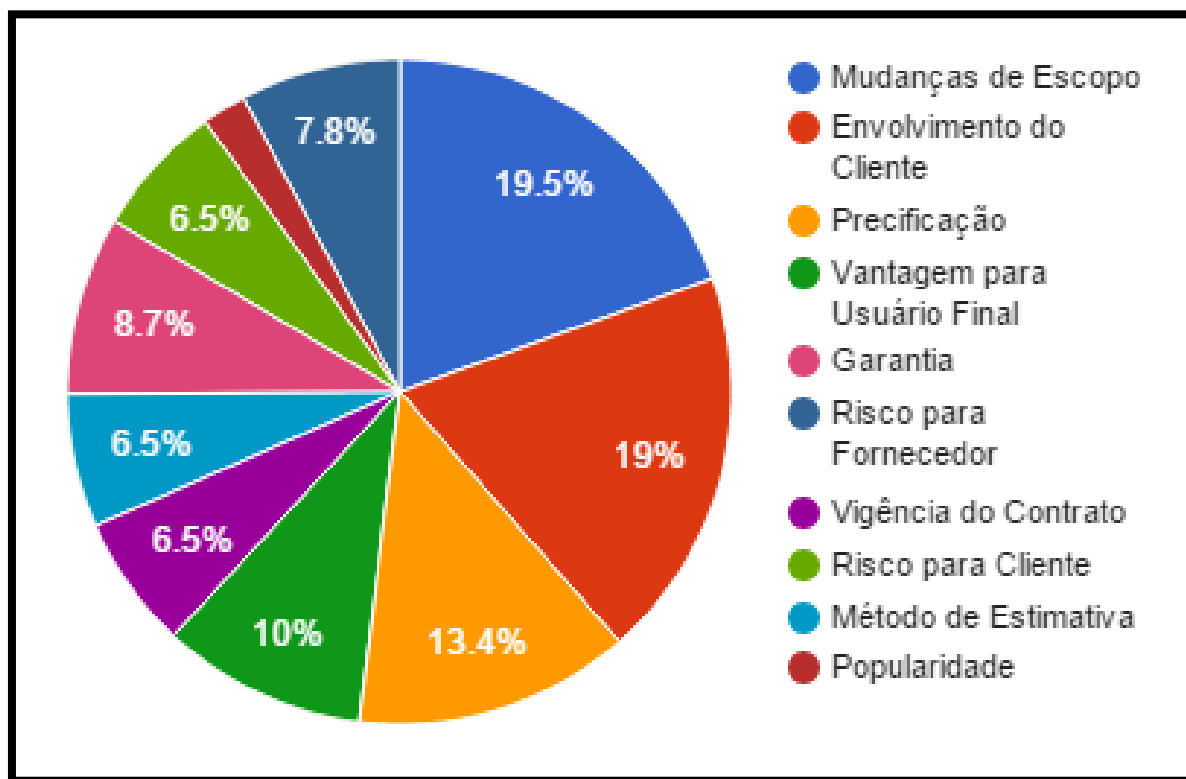


Figura 11: Gráfico total de relevância das características

Fazendo uma análise do resultado geral somando todos os votos (**Figura 11**), percebe-se que o resultado mais votado e considerado mais relevante foi *Mudanças de Escopo*, com 45 votos (19,5%), que superou o segundo colocado *Envolvimento do Cliente* por apenas 1 voto, terminando com 44 (19%). *Precificação* mostrou-se a terceira característica mais votada na pesquisa, com 31 votos (13,4%), seguida da *Vantagem para o Usuário Final*, que teve 23 votos (10%). *Garantia*, surpreendentemente, terminou como quinta mais votada, tendo 20 votos (8,7%). *Risco para o Fornecedor* se mostrou mais relevante que o *Risco para o Cliente* (7,8% contra 6,5%), e *Métodos de Estimativa* e *Vigência do Contrato* tiveram votação pouco expressiva, com apenas 15 votos (6,5%). Por fim, *Popularidade* foi a característica menos votada, tendo somente 5 votos (2,2%).

6.1.3 Relevância das Características para o Setor Público

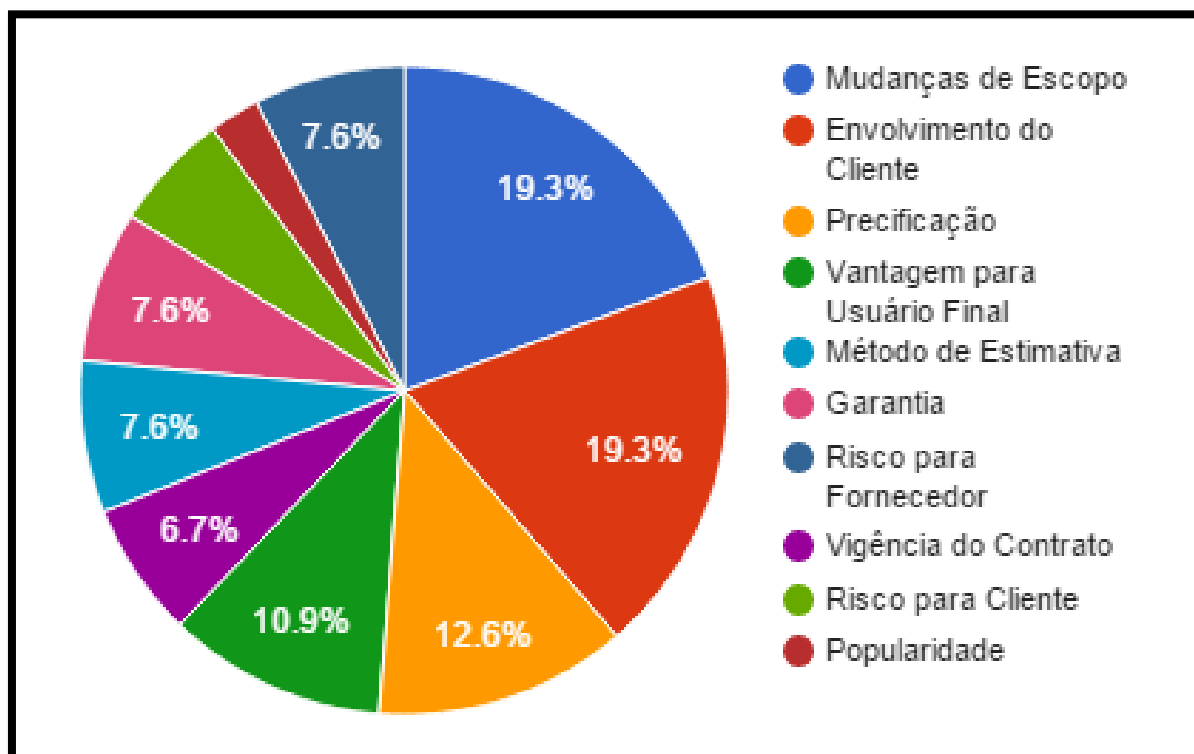


Figura 12: Gráfico da relevância das características para o setor público

Ao avaliar as respostas no Dot Voting preenchido por colaboradores do setor público (tanto contratantes quanto contratados), percebe-se uma forte preocupação com cláusulas relacionadas a *Mudanças de Escopo* e ao *Envolvimento do Cliente* no projeto (19,3%, 23 votos). *Precificação* também foi muito votada, com 12,6%, 15 votos, seguida de perto pela *Vantagem para Usuário Final* com 10,9%, 13 votos. As demais características tiveram uma votação muito próxima, ao redor dos 7% (entre 9 e 7 votos), com exceção da *Popularidade*, que teve apenas 2,5% dos votos (3 votos) (**Figura 12**).

6.1.4 Relevância das Características para o Setor Privado

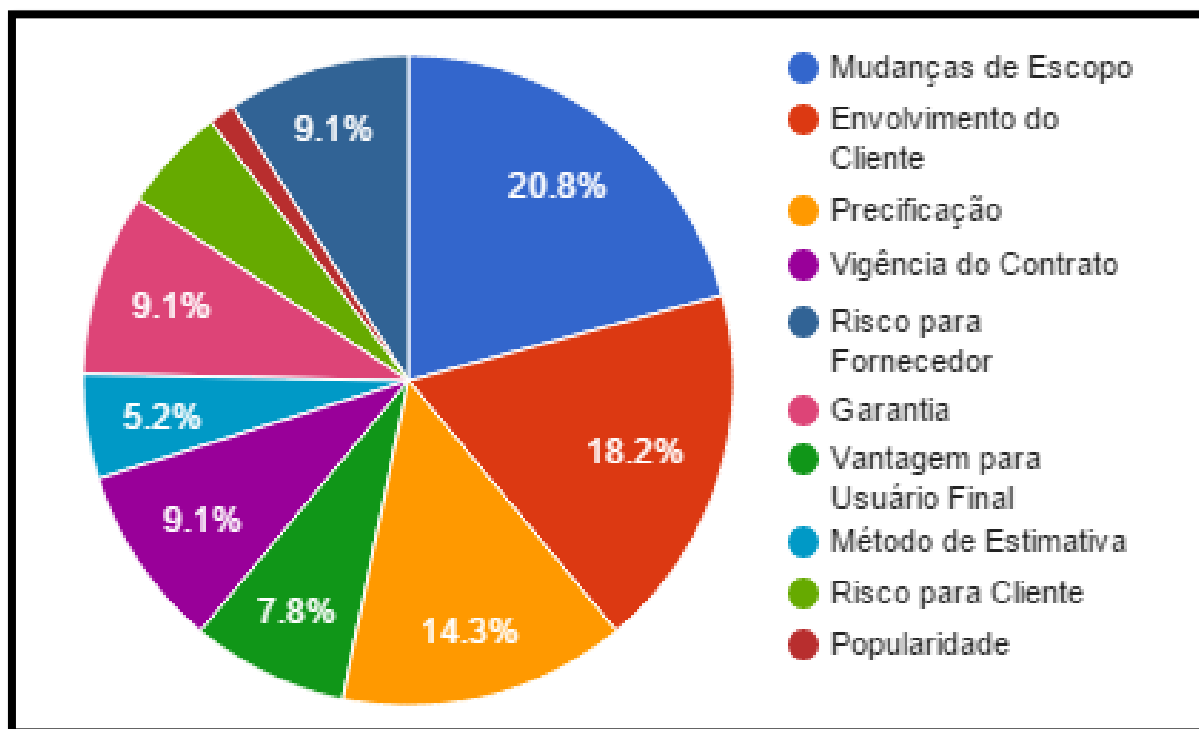


Figura 13: Gráfico da relevância das características para o setor privado

Com relação às respostas obtidas nos questionários de funcionários do setor privado (contratantes e contratados), é possível perceber que os mesmos três aspectos são os mais votados (*Mudanças de Escopo* 20,8% (16 votos), *Envolvimento do Cliente* 18,2% (14 votos) e *Precificação* 14,3% (11 votos)). No entanto, nota-se uma preocupação maior com *Vigência do Contrato*, *Garantia* e *Risco para o Fornecedor* (todos com 9,1%, 7 votos) (**Figura 13**).

6.1.5 Relevância das características para o lado Cliente

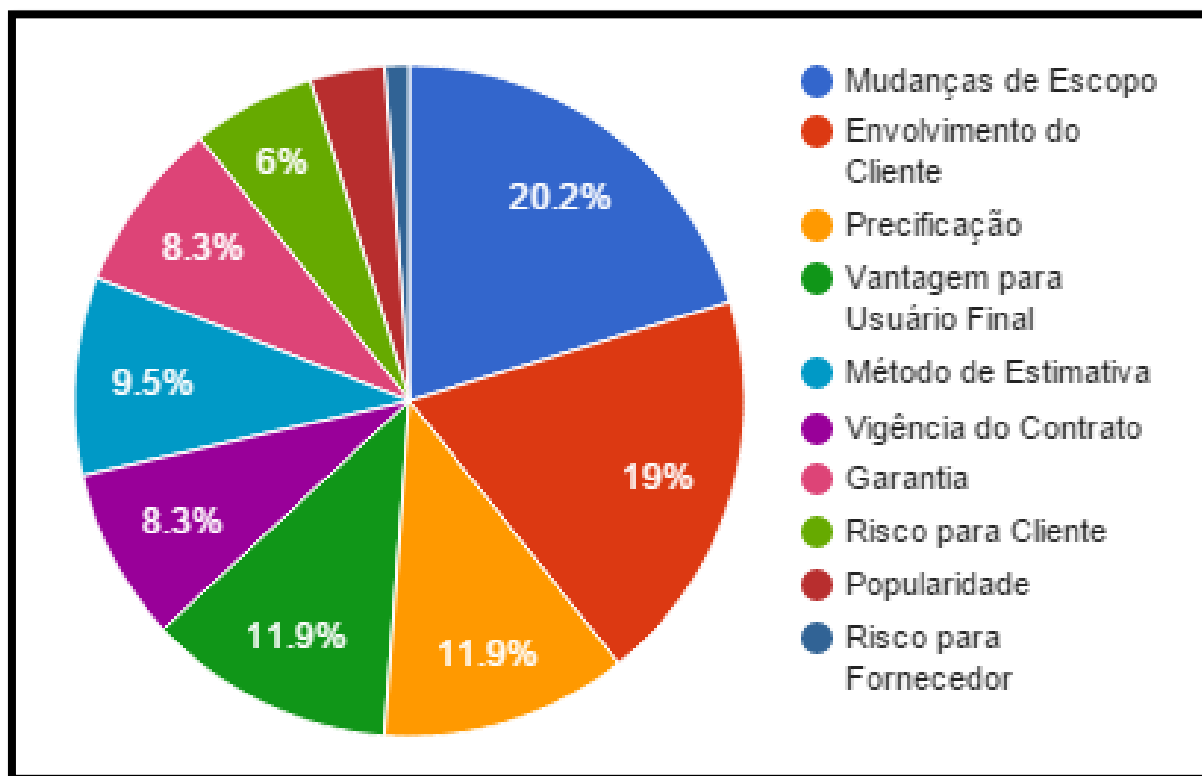


Figura 14: Gráfico da relevância das características para o lado Cliente

Avaliando os questionários de respondentes que trabalham do lado cliente (tanto no setor público quanto privado), continua-se percebendo a mesma tendência, com *Mudanças de Escopo* em primeiro com 20,2% (17 votos) e *Envolvimento do Cliente* com 19% (16 votos). Nota-se, porém, uma votação mais expressiva de *Vantagem para o Usuário Final*, que empata com *Precificação* com 11,9% (10 votos). É interessante perceber nesse resultado o desinteresse dos clientes pelo *Risco para o Fornecedor*, que teve apenas 1,2% (1 voto) dos votos. Também é relevante perceber que o próprio risco (para o cliente) não teve uma votação expressiva – apenas 6% (5 votos) dos votos (**Figura 14**).

6.1.6 Relevância das características para o lado Fornecedor

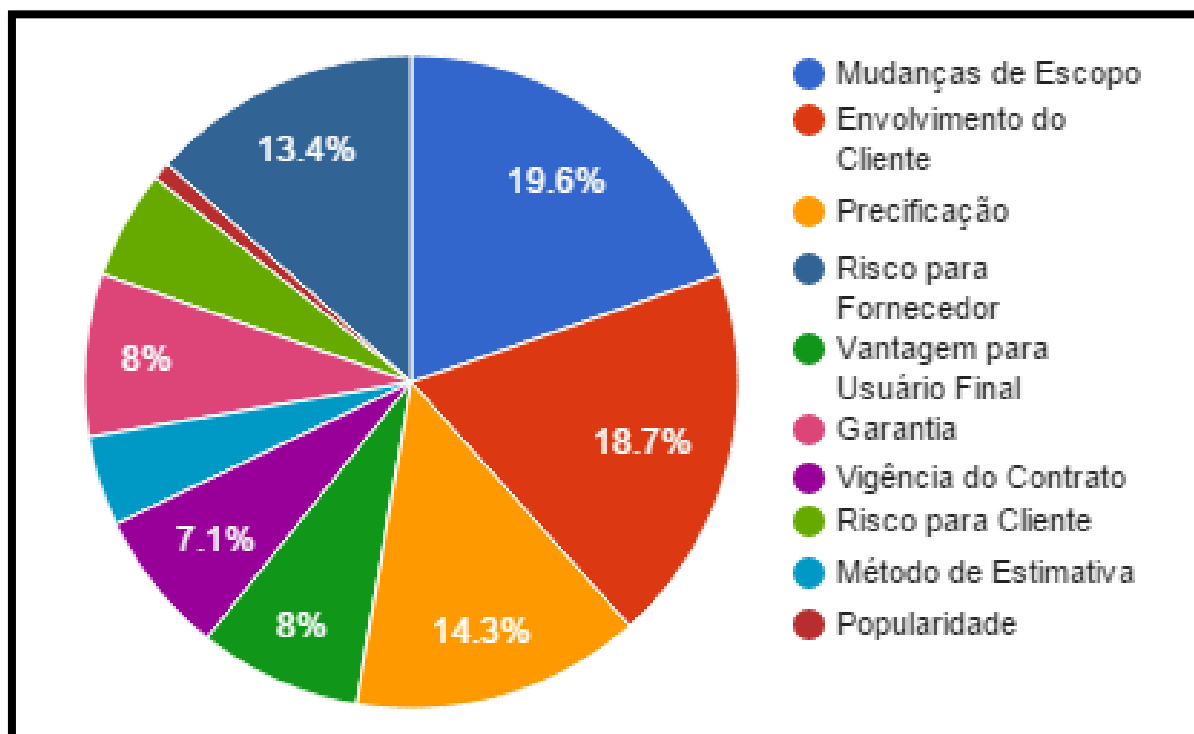


Figura 15: Gráfico da relevância das características para o lado Fornecedor

Ao analisar as respostas vindas do lado fornecedor (somando setor público e privado), nota-se, apesar da mesma tendência dos 3 primeiros colocados (*Mudanças de Escopo* 19,6% (22 votos), *Envolvimento do Cliente* 18,7% (21 votos), *Precificação* 14,3% (16 votos)), que os fornecedores têm uma preocupação muito maior com riscos, principalmente os que envolvem o seu lado (*Risco para o Fornecedor*), com expressivos 13,4% (15 votos). A votação nos outros aspectos ficou parelha, ao redor dos 7% (entre 9 e 5 votos), com exceção novamente de *Popularidade*, que teve apenas 0,9% (1 voto) dos votos (**Figura 15**).

6.2 Análise do Resultado

Pode-se perceber, após análise dos dados obtidos com o questionário, que a necessidade de prever e facilitar mudanças de escopo através de cláusulas no contrato é reconhecida pela grande maioria. É, de fato, sabido que os requisitos irão mudar e como tratar essa mudança deve estar previsto no contrato. Outra conclusão obtida é o consentimento quase geral de que o envolvimento do cliente é necessário e deve ser forçosamente previsto em contrato. A precificação, ou seja, o acordo em relação à forma de pagamento, também figurou entre os mais votados, como era de se esperar.

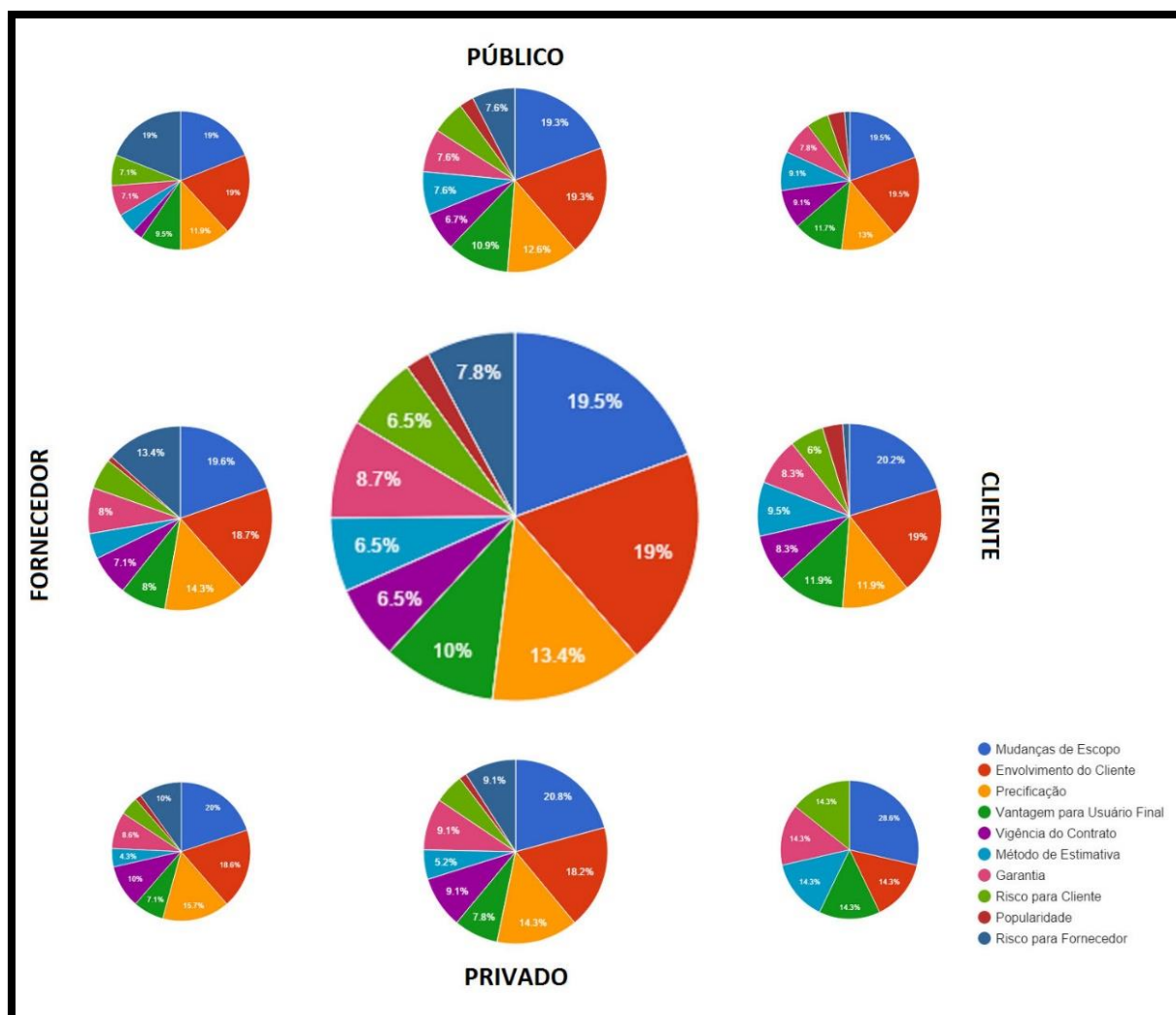


Figura 16: Consolidado de todos os gráficos

A **Figura 16** é um consolidado que permite uma análise conjunta de todos os gráficos. No centro, encontra-se o gráfico que soma todos os votos (inclusive dos que optaram por não se posicionar entre público, privado, cliente e fornecedor). No eixo horizontal, estão localizados os gráficos dos votos de Fornecedores (na esquerda) e Cliente (na direita). Analogamente para o eixo vertical, onde estão localizados os votos de funcionários de empresas Públicas (acima) e empresas Privadas (abaixo). Nos vértices do quadrado formado pelos gráficos, encontram-se as interseções entre os eixos. Por exemplo, acima na direita encontra-se o gráfico de votantes que trabalham do lado cliente **E** são de empresas públicas. O mesmo raciocínio se aplica aos demais gráficos nas extremidades.

Dentre as surpresas do resultado da pesquisa, pode-se citar: (1) a alta preocupação com o usuário final, que não é algo tão comum de se perceber em clientes e fornecedores; (2)

a baixa votação de características como estimativas e vigência do contrato, que são comumente vistas em contratos atuais, mas possivelmente foram percebidas como menor prioridade que as outras (o que não quer dizer que não tenham sua importância); (3) a preocupação com o risco para o fornecedor superar a preocupação com o risco para o cliente, o que surpreendeu, porque usualmente os clientes têm grandes preocupações com riscos.

Dados os resultados obtidos, foi possível confirmar que o problema existe: muitas pessoas são mesmo afetadas pelas restrições contratuais e trabalham sob contratos que não favorecem seu projeto, equipe, processo ou modo de trabalhar. Para resolver esse problema, o estabelecimento de novos contratos se mostra necessário e muito vantajoso para a maioria das organizações. Os contratos têm mesmo que evoluir junto com as metodologias de desenvolvimento de software, e as características avaliadas e descritas podem ajudar a criação de um contrato que espelhe melhor realidade das organizações.

Um detalhe relevante a ser apontado é que as características “Mudanças de Escopo”, “Envolvimento do Cliente” e “Vantagem para Usuário Final”, primeira, segunda e quartas colocadas no gráfico total das características (**Figura 11**), podem ser vistas diretamente citadas no Manifesto Ágil (BECK et al., 2001), conforme relação abaixo:

- **“Mudanças no Escopo”** relacionado a “Responder a mudanças MAIS QUE seguir um plano”;
- **“Envolvimento do Cliente”** relacionado à “Colaboração com cliente MAIS QUE negociação de contratos”;
- **“Vantagem para Usuário Final”** relacionado à “Software em Funcionamento MAIS QUE documentação abrangente”.

Esse relacionamento mostra e valida a pesquisa, confirmando que a mentalidade dos entrevistados já possui uma forte identificação com características ágeis.

7 Análise dos Modelos de Contratos Frente aos Resultados da Pesquisa

Nesse capítulo, é feita uma análise comparativa considerando todas as informações levantadas nos capítulos anteriores. As características priorizadas de acordo com o ranqueamento obtido como resultado do Capítulo Seis são observadas caso a caso em cada um dos modelos de contrato descritos no Capítulo Quatro. Dessa forma, é possível verificar com mais clareza e facilidade quais são os contratos que tratam as características de melhor forma e quais tem mais restrições às mesmas. Nas seções seguintes, é mostrado um quadro que resume toda a pesquisa, mapeando cada característica em cada contrato, e pontuando se tal característica, inserida no contexto daquele contrato, assume um valor positivo, neutro ou negativo. Em seguida, justifica-se individualmente as categorizações feitas no mapeamento e conclui-se sobre quais contratos são mais interessantes com base no *ranking* das características.

7.1 Análise da Matriz Comparativa

Com os modelos de contratos identificados, e as principais características de modelos ranqueadas, resta então mapear como cada uma das características se comporta dentro dos modelos descritos. Esse mapeamento é, enfim, mostrado na Matriz Comparativa (**Figura 17**). É importante ressaltar que as células da matriz estão pintadas de verde, amarelo e vermelhas para representar se o atributo avaliado dentro do contexto do contrato em questão é positivo, neutro ou negativo, respectivamente. Outra informação relevante é a sequência exibida das características: elas são apresentadas em ordem decrescente, da esquerda para a direita, com relação ao ranqueamento realizado.

Ranking	1	2	3	4	5	6	7	8	9	10	
Contratos \ Características	Mudanças	Envolvimento do Cliente	Precificação	Vantagem para Usuário Final	Garantia	Risco para Fornecedor	Vigência do Contrato	Método de Estimativa	Risco para Cliente	Popularidade	
Contratos Tradicionais	Contrato Tradicional de Preço Fixo	Difíceis e caras (-)	Muito baixo (-)	Vantagem Ilusória (0)	Não há (-)	Costuma haver (+)	Alto (-)	Fixa / Multas (-)	Comum APF (0)	Teoricamente Baixo (0)	Muito popular (+)
	Tempo & Material	Difíceis e caras (-)	Baixo (-)	Relativo (0)	Improvável (0)	Não é explícito (0)	Baixo (+)	Fixa / Multas (-)	Livre (0)	Muito Alto (-)	Muito popular (+)
	Contrato de Lucro Fixo	Difíceis e caras (-)	Não obrigatória (-)	Vantagem para ambos (+)	Improvável (0)	Não é explícito (0)	Compartilhado (0)	Fixa / Multas (-)	Livre (0)	Compartilhado (0)	Pouco popular (-)
	Contrato com Cláusula de Bônus e Penalidade	Difíceis e caras (-)	Não obrigatória (-)	Vantagem para ambos (+)	Improvável (0)	Não é explícito (0)	Alto (-)	Fixa / Multas (-)	Livre (0)	Baixo (+)	Pouco popular (-)
Contratos Ágeis	Contrato Ágil de Preço Fixo	Facilitadas (+)	Constante e Necessária (+)	Fixo (0)	Alta (+)	Não é explícito (0)	Médio (0)	Fixa / Multas (-)	Livre (0)	Baixo (+)	Média (0)
	Tempo & Material Ágil	Facilitadas (+)	Constante e Necessária (+)	Relativo (0)	Alta (+)	Não é explícito (0)	Baixo (+)	Fixa / Multas (-)	Livre (0)	Muito Alto (-)	Média (0)
	Contrato com Preço Fixo por Unidade de Trabalho	Aceitas (+)	Constante e Necessária (+)	Relativo (0)	Alta (+)	Não é explícito (0)	Alto (-)	Fixa / Multas (-)	APF, Story Points... (0)	Baixo (+)	Popular (+)
	Contrato "Money for Nothing, Change for Free"	Fáceis e gratuitas (+)	Constante e Necessária (+)	Vantagem para ambos (+)	Alta (+)	Não é explícito (0)	Baixo (+)	Flexível (+)	Comum Story Points (0)	Baixo (+)	Pouco popular (-)
	Contrato Progressivo	Facilitadas para iterações seguintes (+)	Constante e Necessária (+)	Livre (+)	Alta (+)	Não é explícito (0)	Compartilhado (0)	Flexível (+)	Livre (0)	Compartilhado (0)	Média (0)
	Contrato "No Cure, No Pay"	Facilitadas (+)	Constante e Necessária (+)	Relativo (0)	Alta (+)	Forte (+)	Alto (-)	Flexível (+)	Livre (0)	Baixo (+)	Pouco popular (-)
Contrato com Objetivo de Custo	Aceitas com cuidado (0)	Constante e Necessária (+)	Vantagem para ambos (+)	Alta (+)	Não é explícito (0)	Compartilhado (0)	Fixa / Multas (-)	Livre (0)	Compartilhado (0)	Pouco popular (-)	

Figura 17: Matriz Comparativa entre Contratos

Analisando individualmente cada uma das relações, verifica-se:

- **Contrato Tradicional de Preço Fixo**

- Mudanças: Difíceis e caras, já que o escopo é fechado e definido no início do projeto.
- Envolvimento do Cliente: Muito baixo, visto que, em tese, já se sabe tudo que deve ser feito do sistema e não parece ser necessária sua participação durante o projeto.
- Precificação: O preço é fixado, mas a vantagem é ilusória porque a probabilidade de algo dar errado quando o escopo, o tempo e o custo estão fixos é altíssima.
- Vantagem para o Usuário Final: Não há, porque o usuário final receberá, somente após muitos meses, um produto abarrotado de funcionalidades de pouca utilidade.
- Garantia: É comum nesse tipo de contrato encontrar cláusulas que definem uma garantia para o cliente, obrigando o fornecedor a prover manutenção e garantir o funcionamento do sistema durante um certo período.
- Risco para o Fornecedor: Alto, já que ele tem que entregar um escopo fixo em um prazo fixo, com custo fixo. Para isso, caso qualquer imprevisto ocorra, ele precisa abrir mão de sua margem de lucro ou da qualidade do produto.
- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: É comum observar Pontos de Função como estimativa desse tipo de contrato, mas isso pode variar.
- Risco para o Cliente: Teoricamente baixos, porque ele sabe o prazo, o custo e o que vai receber. No entanto, há chances altas de ser necessária a realização de mudanças, o que gera aumento de custo e alteração no escopo.
- Popularidade: É um dos contratos mais comumente usados no mercado.

- **Contrato Tempo & Material**

- Mudanças: Com escopo fechado, as mudanças são difíceis e caras, pois todo o sistema já foi levantado.
- Envolvimento do Cliente: É baixo, pois o sistema já foi totalmente levantado, sendo assim pouco necessário que o cliente se envolva novamente.
- Precificação: Pode ser bom ou ruim ter um pagamento por Tempo & Material. Por um lado, é simples e direto. Por outro, é arriscado se não houver controle das horas realizadas.
- Vantagem para o Usuário Final: Não há incentivos para entregar o software com antecedência, nem para entregar as funcionalidades mais importantes ou um software mais enxuto e usável. Por isso, é improvável que haja vantagem para o usuário final.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: É baixo, já que ele recebe pelas horas que trabalha, como descrito na seção 4.1.2.
- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: Muito alto, pois o controle fica todo na mão do fornecedor, que recebe pelas horas de trabalho. A melhor forma para mitigar esse risco é definindo um limite de horas.
- Popularidade: É um modelo de contrato altamente popular e comum no mercado.

- **Contrato de Lucro Fixo**

- Mudanças: Com escopo fechado, mudanças de escopo nesse tipo de contrato são difíceis, pois podem afetar o prazo fixo acordado e impactar em problemas para os dois lados.

- Envolvimento do Cliente: Não é obrigatório, pois não há cláusulas que obriguem essa atividade, visto que sua participação não afeta o resultado final nesse modelo de contrato.
- Precificação: É vantajoso para ambos porque o fornecedor vai sempre receber seu lucro todo, e o cliente vai pagar somente os custos do projeto se o projeto se estender após o prazo estipulado.
- Vantagem para o Usuário Final: Não há incentivos para entregar o software com antecedência, nem para entregar as funcionalidades mais importantes ou um software mais enxuto e usável. Por isso, é improvável que haja vantagem para o usuário final.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: O risco é compartilhado entre as partes, porque se o fornecedor entregar o projeto com antecedência, há vantagens para os dois, mas se ele atrasar, também há desvantagens para os dois.
- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: O risco é compartilhado entre as partes, porque se o fornecedor entregar o projeto com antecedência, há vantagens para os dois, mas se ele atrasar, também há desvantagens para os dois.
- Popularidade: Modelo de contrato pouco popular no mercado.

- **Contrato com Cláusula de Bônus e Penalidade**

- Mudanças: Com escopo fechado, mudanças de escopo nesse tipo de contrato são difíceis, pois podem afetar o prazo fixo acordado e impactar em problemas para os dois lados, já que o modelo é construído ao redor do prazo fixo (bônus e penalidade dependem desse prazo).
- Envolvimento do Cliente: Não é obrigatório, pois não há cláusulas que obriguem essa atividade, visto que sua participação não afeta o resultado final nesse modelo de contrato.

- Precificação: É vantajosa para ambos, porque o fornecedor vai sempre receber um bônus se entregar o produto com antecedência, e tem uma penalidade se o projeto passar do prazo (indenizando o cliente).
- Vantagem para o Usuário Final: Não há incentivos para entregar o software com antecedência, nem para entregar as funcionalidades mais importantes ou um software mais enxuto e usável. Por isso, é improvável que haja vantagem para o usuário final.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: É alto, porque o cliente pode dificultar o aceite do produto caso ele não tenha uma necessidade real de receber o produto cedo ou até mesmo no prazo.
- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: É baixo, já que ele pode controlar o aceite do produto.
- Popularidade: Modelo de contrato pouco popular no mercado.

- **Contrato Ágil de Preço Fixo**

- Mudanças: São facilitadas porque consideram o escopo flexível. Apenas o macro escopo é levantado inicialmente, e modificações nele são facilitadas durante o ciclo do projeto.
- Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
- Precificação: O preço é fixo, apesar da priorização das funcionalidades. Ainda é necessário que seja pago por tudo que foi contratado.
- Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.

- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: É médio, porque o fornecedor só recebe quando os critérios de aceite forem atingidos. No entanto, a flexibilidade com mudanças reduz o risco em comparação com o modelo tradicional.
- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: Baixo, pois o preço é fixo, assim como o tamanho e o prazo. A flexibilidade do escopo reduz o risco de aumento do custo no final do projeto.
- Popularidade: Média, pois já há empresas no mercado adotando preço fixo ágil.

- **Contrato Tempo & Materiais Ágil**

- Mudanças: São facilitadas porque consideram o escopo flexível. Apenas o macro escopo é levantado inicialmente, e modificações nele são facilitadas durante o ciclo do projeto.
- Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
- Precificação: Pode ser bom ou ruim ter um pagamento por Tempo & Material. Por um lado, é simples e direto. Por outro, é arriscado se não houver controle das horas realizadas.
- Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: É baixo, já que o fornecedor recebe pelas horas que trabalha, como descrito na seção 4.1.2.

- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: Muito alto, pois o controle fica todo na mão do fornecedor, que recebe pelas horas que trabalho. A melhor forma para mitigar esse risco é definindo um limite de horas.
- Popularidade: Média, pois já há empresas no mercado adotando Tempo & Materiais Ágil.
- **Contrato com Preço Fixo por Unidade de Trabalho**
 - Mudanças: São aceitas, pois o escopo é flexível. O pagamento é feito pela unidade de trabalho providamente entregue. Por isso, são permitidas as trocas de funcionalidades com base nessas unidades de trabalho, tendo o cuidado de não afetar o prazo final.
 - Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
 - Precificação: Paga-se pela unidade de trabalho entregue. Caso sejam entregues muitos “pontos” no período, o pagamento será grande. Caso contrário, será pequeno. Um empecilho pode surgir se os critérios de aceite não estiverem claros o suficiente e se a confiança entre as partes não for boa.
 - Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.
 - Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
 - Risco para o Fornecedor: É alto, pois ele corre o risco de não receber ou receber com atraso, já que depende de critérios de aceitação para cada unidade de trabalho entregue.

- Vigência do Contrato: Há cláusulas com multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Recomenda-se APF, por ser paramétrico e possuir muitas regras e padrões, facilitando a precificação de mudanças. Mas é possível também, apesar de incomum, utilizá-lo com outros métodos, como *Story Points*.
- Risco para o Cliente: É baixo, porque tem poder sobre definir se aceita ou não as funcionalidades.
- Popularidade: Modelo de contrato que algumas empresas e órgãos públicos vêm tentando utilizar.

- **Contrato Money for Nothing, Change for Free**

- Mudanças: Fáceis e gratuitas, porque é permitido incluir, alterar ou remover qualquer item do escopo, desde que o tamanho final não ultrapasse o tamanho inicial acordado.
- Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
- Precificação: É vantajosa para o fornecedor, que recebe seu lucro antecipado se conseguir concluir o projeto antecipadamente, e é vantajosa para o cliente, porque deixa de pagar os custos do projeto se aceitar encerrar o projeto com antecedência.
- Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: É baixo, porque sabe que o projeto vai durar até no máximo o mês acordado, tem liberdade para tratar mudanças, sabe exatamente quanto vai receber de lucro e que vai receber os custos do projeto enquanto o projeto estiver em andamento.

- Vigência do Contrato: Flexível, porque o contrato pode ser encerrado em qualquer iteração e isso é favorável a todas as partes envolvidas.
- Método de Estimativa: É comum que sejam explícitas as estimativas nesse contrato através de *Story Points*, porém não é uma regra.
- Risco para o Cliente: Baixo, porque ele pode receber o produto com antecedência, gastar menos do que o previsto, e conhece o prazo máximo de entrega, além de ter menos chance de mudanças posteriores.
- Popularidade: Modelo de contrato pouco visto no mercado.

- **Contrato Progressivo**

- Mudanças: Nesse modelo, onde o contrato deve ser renovado a cada iteração, não se recomenda incluir mudanças dentro da iteração corrente, mas elas são facilitadas e perfeitamente acopladas nas iterações seguintes.
- Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
- Precificação: A forma de pagamento é livre nesse modelo, podendo ser Tempo & Material, preço fixo por iteração, ou até preço fixo por unidade de trabalho. A forma de pagamento pode até variar entre iterações, se for considerado válido.
- Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: Os riscos são compartilhados nesse modelo, já que o fornecedor deve mostrar valor agregado constantemente para que tenha o contrato renovado, e o cliente deve sempre pagar em dia e cumprir com suas obrigações para que o fornecedor tenha interesse na continuidade do projeto.

- Vigência do Contrato: É flexível, já que o contrato deve ser renovado a cada iteração, podendo ser encerrado por qualquer uma das partes no fim de qualquer iteração.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: Compartilhados, conforme descrito em “Risco para o Fornecedor”.
- Popularidade: Popularidade média, já que já podem ser encontradas organizações que utilizam esse modelo.
- **Contrato “No Cure, No Pay”**
 - Mudanças: São facilitadas, porque o escopo é flexível e o pagamento é em Tempo & Materiais. Modificar o escopo de uma iteração não altera significativamente o produto nesse modelo, já que a principal característica desse contrato é que o pagamento só é feito depois que as histórias são comprovadamente entregues.
 - Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
 - Precificação: Relativa, porque o pagamento é em Tempo & Materiais, e só é realizado depois que as histórias passam por rigorosos critérios de aceitação. Dessa forma, em uma relação com pouca confiança entre as partes, isso torna-se um risco considerável.
 - Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.
 - Garantia: Há uma forte garantia, porque há uma cláusula que define que o pagamento de cada entrega fica retido por 30 dias como garantia depois que a entrega entra em produção. Caso apareça algum defeito, ele deve ser corrigido senão o pagamento não é liberado. Isso é válido para todas as iterações.

- Risco para o Fornecedor: É alto, porque o pagamento só é feito após a comprovação da entrega e depois de 30 dias em ambiente de produção. Dessa forma, se o cliente não colaborar para o sucesso do projeto, o fornecedor fica fortemente prejudicado.
- Vigência do Contrato: Flexível, visto que o contrato tem duração indefinida e pode ser encerrado ao final de qualquer iteração.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: É baixo, porque ele tem a segurança de só pagar quando receber o produto de fato, e tem 30 dias de “teste” no ambiente de produção.
- Popularidade: É um modelo pouco comum no mercado atual.

- **Contrato com Objetivo de Custo**

- Mudanças: Aceitas com cuidado, porque há mudanças que podem afetar o objetivo de custo. Essas devem ser analisadas com cuidado para não prejudicar o acordado inicialmente.
- Envolvimento do Cliente: É constante e necessário, porque é fundamental que o mesmo participe da priorização das funcionalidades que serão desenvolvidas e que dê feedback do que já foi entregue.
- Precificação: Vantajosa para ambos, porque se o projeto terminar antes ou abaixo do objetivo de custo, os lucros são divididos. Caso contrário, o prejuízo é repartido igualmente.
- Vantagem para o Usuário Final: É alta, porque o usuário receberá um software com as funcionalidades mais importantes com antecedência e com mais usabilidade.
- Garantia: Nessa modalidade, não fica explícito e nem é comum que haja cláusulas de garantia, mas nada impede que isso aconteça.
- Risco para o Fornecedor: Compartilhado, porque se os lucros ou os prejuízos são repartidos igualmente, e as duas partes tem igual interesse no término antecipado do projeto.

- Vigência do Contrato: Há cláusulas e multas para término antecipado do projeto, dificultando essa atividade.
- Método de Estimativa: Esse modelo de contrato pode ser associado a todos os tipos de estimativa, sem grande impacto em sua execução.
- Risco para o Cliente: Compartilhado, porque se os lucros ou os prejuízos são repartidos igualmente, e as duas partes tem igual interesse no término antecipado do projeto.
- Popularidade: Modelo de contrato pouco comum no mercado.

7.2 Cálculo Ponderado dos Modelos de Contrato em Relação à Pesquisa

Com o mapeamento de como cada característica de contrato se comporta dentro dos modelos de contrato identificados, é possível fazer uma análise ponderada de quão significativo é cada modelo considerando o ranqueamento identificado no questionário de pesquisa. Para essa análise, usou-se como base conceitos de um método de priorização chamado *Theme Screening*, idealizado por Mike Cohn.

O método *Theme Screening* (COHN, 2012) é uma técnica de priorização utilizada para priorizar *backlog* em projetos ágeis, mas também é aplicável a outros tipos de atividades. Primeiro são identificados critérios que serão utilizados para avaliar os itens a serem priorizados. Seleciona-se, em seguida, um critério que tenha importância intermediária para se tornar a “*baseline*”. Ele servirá como parâmetro de comparação, permitindo avaliar se determinado critério é mais importante (+1) ou menos importante (-1) que o critério determinado como *baseline*. Os itens que serão comparados são, então, listados e avaliados conforme os critérios definidos, sendo classificados em +1, 0 ou -1, de acordo com sua importância em relação à *baseline*. Dessa forma, é possível fazer uma priorização dos itens identificados em relação aos critérios estabelecidos.

O método utilizado nesse trabalho difere do *Theme Screening*, porque não foi estabelecida uma *baseline* para determinar um critério de importância intermediária. Os valores, ou seja, a importância dos critérios foi obtida através do questionário de pesquisa, e com isso foi possível estabelecer um peso para cada critério, que é exatamente o percentual de respostas obtido pelo critério na pesquisa. A partir desse peso para cada critério e dos

contratos identificados, a atividade seguinte foi julgar como cada critério se comportava dentro do contrato e calcular a pontuação de cada contrato.

Nesse cálculo, considera-se que os atributos positivos (pintados em verde) valem $k = +1$ (um ponto), neutros (amarelos) valem $k = 0$ pontos, e negativos (vermelhos) valem $k = -1$ (menos um ponto). Os pesos ($P_1...P_{10}$) de cada característica de contrato foram definidos com base no percentual total de votos obtido na pesquisa (**Figura 11**). Aplica-se, então, a fórmula:

$$Pontuação = \sum_{0 < n < 11} P_n * k$$

Sendo k o valor do atributo e P_n o peso da característica considerada. A matriz da (**Figura 18**) ilustra os cálculos realizados e exibe o resultado do cálculo, calculando a pontuação total de cada contrato na coluna **Pontuação**. É importante ressaltar que o resultado dos cálculos deriva da interpretação dos autores sobre como a característica se comporta dentro dos contratos analisados. O embasamento para avaliar uma característica dentro de um contrato em positiva, neutra ou negativa foi obtido das referências identificadas na revisão sistemática de literatura, entrevistas realizadas e na experiência do autor.

Peso (P1 a P10) (em %)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Pontuação
	19.5	19	13.4	10	8.7	7.8	6.5	6.5	6.5	2.2	
Contratos \ Características	Mudanças	Envolvimento do Cliente	Precificação	Vantagem para Usuário Final	Garantia	Risco para Fornecedor	Vigência do Contrato	Método de Estimativa	Risco para Cliente	Popularidade	
	Contrato Tradicional de Preço Fixo	-19.5	-19	0	-10	8.7	-7.8	-6.5	0	2.2	-51.9
	Tempo & Material	-19.5	-19	0	0	0	7.8	-6.5	0	-6.5	-41.5
	Contrato de Lucro Fixo	-19.5	-19	13.4	0	0	0	-6.5	0	-2.2	-33.8
	Contrato com Cláusula de Bônus e Penalidade	-19.5	-19	13.4	0	0	-7.8	-6.5	0	6.5	-35.1
	Contratos Tradicionais										
	Contrato Ágil de Preço Fixo	19.5	19	0	10	0	0	-6.5	0	6.5	48.5
Contratos Ágeis	Tempo & Material Ágil	19.5	19	0	10	0	7.8	-6.5	0	-6.5	43.3
	Contrato com Preço Fixo por Unidade de Trabalho	19.5	19	0	10	0	-7.8	-6.5	0	6.5	42.9
	Contrato "Money for Nothing, Change for Free"	19.5	19	13.4	10	0	7.8	6.5	0	6.5	80.5
	Contrato Progressivo	19.5	19	13.4	10	0	0	6.5	0	0	68.4
	Contrato "No Cure, No Pay"	19.5	19	0	10	8.7	-7.8	6.5	0	6.5	60.2
Contratos com Objetivo de Custo		19.5	19	13.4	10	0	0	-6.5	0	0	53.2
	Legenda	k:	-1	0	1						

Figura 18: Matriz de Cálculo Ponderado entre Modelos de Contrato

Como pode ser visto na **Figura 19**, os contratos ágeis de modo geral se sobressaem fortemente em relação aos contratos tradicionais, principalmente por causa da diferença nas duas características de maior peso (Mudanças e Envolvimento do Cliente). Dentre todos, o contrato que obteve a maior pontuação com base nas características que a pesquisa identificou como mais importantes foi o modelo *"Money for Nothing, Change for Free"*, seguido pelo Contrato Progressivo. Em oposição, o Contrato Tradicional de Preço Fixo teve a pior pontuação ponderada.

Posição ▾	Nome ▾	Pontuação ▾
1	Contrato "Money for Nothing, Change for Free"	80.5
2	Contrato Progressivo	68.4
3	Contrato "No Cure, No Pay"	60.2
4	Contrato com Objetivo de Custo	53.2
5	Contrato Ágil de Preço Fixo	48.5
6	Tempo & Material Ágil	43.3
7	Contrato com Preço Fixo por Unidade de Trabalho	42.9
8	Contrato de Lucro Fixo	-33.8
9	Contrato com Cláusula de Bônus e Penalidade	-35.1
10	Tempo & Material	-41.5
11	Contrato Tradicional de Preço Fixo	-51.9

Figura 19: Resultado do Cálculo Ponderado entre Contratos

8 Conclusão

As metodologias ágeis e seus conceitos já estão bastante difundidos no mercado de desenvolvimento de software. Muitas empresas utilizam métodos ágeis e a maioria já introduziu algum conceito de agilidade em seu cotidiano. A própria pesquisa realizada nesse trabalho ilustra esse fato. Dentre as características identificadas nos contratos de desenvolvimento de software, as que foram consideradas mais importantes pelos próprios respondentes da pesquisa estão fortemente atreladas às Metodologias Ágeis, conforme descrito na seção 6.2. No entanto, somente essa mentalidade não é suficiente se os contratos em vigor ainda apresentam características tradicionais. Por isso, a utilização dos modelos de contratos ágeis citados nesse trabalho é um passo fundamental para conduzir a adoção das práticas ágeis a um novo patamar, onde cliente, fornecedor e até usuário final têm benefícios diretos advindos das características inovadoras dos métodos ágeis.

8.1 Análise dos Objetivos e Condução do Trabalho

Esse trabalho teve como principal objetivo resolver o problema da falta de alinhamento das contratações de desenvolvimento de software atuais aos princípios ágeis que muitas empresas já utilizam, pretendem utilizar ou até mesmo têm receio do impacto da adoção. No entanto, existe uma restrição para esse objetivo, que é o fato dos métodos ágeis pregarem que “Colaboração com o cliente mais que negociação de contratos”. Essa sentença não invalida a necessidade dos contratos, mas reforça que sua importância deve ser reduzida, deixando somente o que for fundamental e indispensável no contrato.

Dado o problema e sua restrição, a solução inicialmente buscada foi identificar novas formas de contratação na literatura, através da execução da revisão sistemática de literatura. Em seguida, tendo em vista a restrição derivada do manifesto ágil, foi realizada uma pesquisa utilizando um questionário, com o objetivo de entrevistar conhecedores de contratos de desenvolvimento de software e identificar quais eles consideram ser as principais características presentes em um contrato de desenvolvimento de software. A pesquisa atingiu resultados considerados satisfatórios, e permitiu a realização de um ranqueamento dos atributos dos contratos de desenvolvimento de software.

Com as duas pesquisas executadas, tornou-se possível cruzar as informações obtidas e analisar individualmente, para cada modelo de contrato, como cada atributo ranqueado se comporta no modelo observado, e categorizar esse comportamento em “Positivo”, “Negativo” ou “Neutro”. Essa análise tornou possível, a partir de uma análise ponderada em relação aos percentuais obtidos na pesquisa de relevância das características, identificar quais modelos de contratos estão mais alinhados com as características consideradas mais importantes.

Sendo assim, é dada uma possível solução para tratar o problema do desalinhamento entre contratos e metodologias utilizadas. Recomenda-se utilizar os modelos de contratos mais alinhados aos atributos mais relevantes. No entanto, é comum que não seja possível a adoção direta de um modelo de contrato como os desenvolvidos no trabalho. Nesse caso, a melhor solução é customizar o modelo de contrato para que ele se adeque à realidade vivida pelas partes envolvidas na contratação, porém mantendo sempre em mente que o contrato deve ser enxuto, conter apenas o fundamental e focar nas principais características que um contrato deve ter. Com isso, a suposição feita no primeiro capítulo, que questionava se existe alguma forma de contratação que seja benéfica para ambas as partes, pode ser considerada validada.

8.2 Principais Contribuições

Esse trabalho fornece quatro principais contribuições para o desenvolvimento do tema de contratação de desenvolvimento de software ágil. São elas:

- 1) Identificação de modelos de contratos (tradicionais e ágeis) com utilização de revisão sistemática de literatura;
- 2) Identificação das características mais relevantes de um contrato de desenvolvimento de software a partir de um questionário de pesquisa com respondentes conhecedores do tema.
- 3) Análise do comportamento dos modelos de contratos identificados em face às principais características de contratos ranqueadas.
- 4) *Framework* de análise de relevância de contratos com base em características. O *framework* pode ser replicado por qualquer modelo de contrato.

8.3 Limitações do Trabalho

Dentre as limitações do trabalho executado, existem algumas que são mais importantes. O fato de a pesquisa focar grande parte do seu esforço em empresas privadas, deixando um pouco de lado as empresas da esfera pública, pode ser considerada uma das principais limitações do trabalho. Já existe um grande interesse de órgãos públicos (PRIKLADNICKI et al., 2015) em adaptar suas contratações para incluir atributos ágeis em seus editais e termos de referência. Porém, as dificuldades atreladas à legislação brasileira (como a Lei Nº 8.666) foram o grande empecilho para que esta dissertação buscasse trabalhar com contratos que não pudessem ser afetados pela legislação vigente.

Outro fator limitador dos resultados deste trabalho é a falta de experimentação de alguns dos modelos de contratos identificados. Em sua maioria, como pudemos observar pela característica “Popularidade”, os contratos considerados ágeis têm pouquíssima divulgação no mercado, e, conseqüentemente, baixa experimentação. A existência de trabalhos expondo casos de experimentação desses modelos de contratos certamente incentivaria e difundiria a implementação dos mesmos.

Com relação ao questionário de pesquisa, a descentralização dos questionários, resultante da falta de uma ferramenta otimizada para executar o Dot Voting, gerou muita dificuldade no momento da consolidação dos dados, o que pode ter inserido uma margem de erro de alguns votos nos resultados específicos de empresas públicas/privadas (aproximadamente dois votantes preencheram equivocadamente o questionário, e podem ter distorcido o resultado dos gráficos específicos para empresas públicas/privadas). Além disso, a quantidade total de votantes foi boa, porém o objetivo era ter obtido o dobro de respostas. No entanto, dada a semelhança entre todos os gráficos parciais observados, é possível assumir que o resultado final seria muito próximo do obtido.

Por fim, a falta de experiência na execução da revisão sistemática de literatura, a busca manual das referências nas bases de dados e limitações nos filtros das bases de dados (algumas não permitiam a filtragem por data, por exemplo) podem ter prejudicado e impedido um resultado melhor, com ainda mais modelos de contratos diferentes.

8.4 Trabalhos Futuros

A primeira sugestão de trabalhos futuros é a recomendação da expansão da análise realizada na matriz de comparação para mais modelos de contratos com o objetivo de identificar e incentivar o uso de modelos de contratos cada vez mais modernos e alinhados às práticas ágeis.

Sugere-se também que sejam feitas experimentações com os modelos de contratos citados nesse trabalho, principalmente porque há pouquíssimos registros acadêmicos de experimentações de modelos menos tradicionais realizadas em empresas reais.

Com o objetivo de melhorar a precisão do método de análise de contratos, uma possível melhoria seria remover o critério Popularidade, pois ele teve uma votação com pouquíssima expressividade e não aparenta ter muito sentido nesse tipo de análise.

Uma outra possível melhoria é formar um modelo de contrato “sintético”, que seja composto somente de atributos que foram avaliados em positivo (verdes). Esse modelo pode ser construído juntando as melhores partes de cada contrato e obtendo somente atributos positivos. Uma vez construído, deve ser feita uma análise sobre ele para verificar se ele realmente faz sentido ou possui inconsistências, e, em caso positivo, experimentá-lo.

Outra sugestão é executar uma nova pesquisa, porém, dessa vez, buscando a opinião dos entrevistados sobre os modelos de contratos que obtiveram melhores pontuações nesta dissertação, principalmente o *“Money for Nothing, Change for Free”*. Dessa forma, será possível validar se eles de fato são as melhores opções.

Pretende-se, com o uso de Lógica *Fuzzy*, melhorar a pontuação realizada para comparar os modelos de contratos. A Lógica *Fuzzy* permitiria que a agregação realizada na comparação dos contratos/características seja melhor conceituada, pois da forma atual não há uma ‘grandeza’ ou unidade para a pontuação obtida. Dessa forma, no futuro, recomenda-se que o método de agregação e pontuação seja atualizado para incluir conceitos de Lógica *Fuzzy*.

Finalmente, espera-se que novos modelos de contratos ágeis surjam com combinações de atributos e cláusulas deste trabalho e também de outros que não foram citados aqui, e que, dessa forma, as empresas sigam um caminho rumo a uma realidade onde há plena confiança entre as partes e que todos os envolvidos tenham ganhos reais sem que seja necessário prejudicar a outra parte.

9 Referências

ABRAHAMSSON, P. et al. **Agile software development methods: Review and analysis**, VTT Finland, 2002

AGILE MANIFESTO 2001. **Manifesto for agile software development**. Disponível em: from <http://www.agilemanifesto.org>. Acessado em: 11 de setembro de 2015

ARBOGAST, T., LARMAN, C., VODDE, B. **Practices for Scaling Lean & Agile Development: Large, Multisite, & Offshore Product Development with Large-Scale Scrum**, 2012

AUDY, J. H., **Reunião Diária e Quadro de Tarefas**, acessado em <https://jorgekotickaudy.wordpress.com/2012/03/07/reuniao-diaria-e-quadro-de-tarefas/> em 9 de dezembro de 2015, 2012.

BEAL, A. **Segurança da Informação: princípios e melhores práticas para a proteção dos ativos de informação das organizações**, São Paulo: Editora Atlas, 2005

BECK, K. **Extreme Programming Explained: Embrace Change**, Addison-Wesley Professional, 2000

BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R., MELLOR, S., SCHWABER, K., SUTHERLAND, J., THOMAS, D., **Manifesto for Agile Software Development**, 2001

BEEDLE, M., DEVOS, M., SHARON, Y., SCHWABER, K., SUTHERLAND, J. **SCRUM: An extension pattern language for hyperproductive software development**, 2000

BIOLCHINI, J., MIAN, P.G., NATALI, A.C. C., TRAVASSOS, G.H. **Systematic review in software engineering**, Technical report, System Engineering and Computer Science Dept., COPPE/UFRJ, 2005

BOEHM, B., TURNER, R. **Balancing agility and discipline: A guide for the perplexed**. [S.l.]: Addison-Wesley Professional, 2004

BOEHM, B., TURNER, R. **Management challenges to implementing agile processes in traditional development organizations**. IEEE Software, v. 22, n. 5, p. 30 – 39, out. 2005

BOOK, M., GRUHN, V., STRIEMER, R. **adVANTAGE: A Fair Pricing Model for Agile Software Development Contracting**, 2012

CHAGAS, A.T.R., **O Questionário na Pesquisa Científica**, 2000

CHUNG, K. C.; BURNS, P. B; KIM, H. M. Clinical Perspective: A Practical Guide to Meta-Analysis. **The Journal of Hand Surgery**. 31A (10): 1671, 2006

COHN, M. **User Stories Applied: For Agile Software Development**, 2004

COHN, M., **Theme Screening**, <http://productowner.net/tag/theme-screening>, 2012

CRUZ, C. S., ANDRADE, E. L. P., FIGUEREIDO, R. M. C, **Processo de contratação de serviços de tecnologia da informação para organizações públicas**, Brasília: [s.n.], 2011

DICEMAN, J. **Dot Voting**, www.dotmocracy.org, acessado em novembro de 2015, 2013

ECKFELDT, B., MADDEN, R. **Selling Agile: Target Cost Contracts**, 2005

EDWARDS, I., BICKERSTAFF, R., DUISBERG, A. **Contracting for Agile Software Development Projects**, 2012

FLEXIBLE Contracts, www.flexiblecontracts.com, 2013

FRANCO, C. A. C., TOLEDO, R. P. R. **Terceirização do desenvolvimento de software no Brasil e nos EUA**, Revista do Tribunal de Contas da União, v. 45, p. 18-27, 2013

FRANCO, C. A. C. **Análise da legislação para a terceirização do desenvolvimento de software na administração pública brasileira em relação à norte-americana e britânica**, Dissertação de Mestrado em Informática, UFRJ, 2014

GILB, T. **No Cure no Pay: How to Contract for Software Services**, 2007

GOMES FILHO, A. F., CARDOSO, C. F., DE TOLEDO, R. **Usando Métodos Ágeis para Ensinar Métodos Ágeis.**, Workshop Brasileiro de Métodos Ágeis, Florianópolis, SC. Proceedings of Workshop Brasileiro de Métodos Ágeis, 2014

GUNTHER, H. **Como Elaborar um Questionário**, Série: Planejamento de Pesquisa nas Ciências Sociais, Número 01, 2003

HIGHSMITH, J. **Agile Software Development Ecosystems** Boston, Addison Wesley, 2002

HODA, R., NOBLE, J., MARSHALL, S. **The impact of inadequate customer collaboration on self-organizing Agile teams**. Information and Software Technology, v. 53, n. 5, p. 521–534, 2011

IBM. **Rational Unified Process: best practices for software development teams**, www.ibm.com, TP026B, 2001

KEITH, E. R. **Agile Software Development Processes: A Different Approach to Software Design**, disponível em <http://cf.agilealliance.org/articles/system/article/file/1099/file.pdf>, 2002

KERZNER, H. **Project management: a systems approach to planning, scheduling, and controlling**, 10. Ed, 2009

KITCHENHAM, B. A. **Procedures for Undertaking Systematic Reviews**, Joint Technical Report, Computer Science Department, Keele University, 2004

KITCHENHAM, B.A. **Guidelines for Performing Systematic Literature Review in Software Engineering**, EBSE Technical Report, Keele University, 2007

LAACKONEN, K. **Contracts in Agile Software Development**, 2014

LARMAN, C., VODDE, B. **Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum**, Addison-Wesley Professional, 2010

LAUDON K. C., LAUDON J. P. **Sistemas de Informações Gerenciais**. São Paulo: Prentice Hall, 2004

NERUR, S., MAHAPATRA, R., MANGALARAJ, G. **Challenges of migrating to agile methodologies**, ACM, v. 48, n. 5, p. 72–78, 2005

MASON, D. **Waterfall to Lean, With Agile In Between**, <http://www.ceptara.com/blog/agile-scrum-waterfall-comparison>, acessado em novembro de 2015, 2013

OPELT, A., GLOGER, B., PFARL, W., MITTERMAYR, R. **Agile Contracts: Creating and managing successful projects with Scrum**, 2013

PRIKLADNICKI, R., WILLI, R., MILANI, F. **Métodos Ágeis para Desenvolvimento de Software**, Bookman Editora, 2014

PRIKLADNICKI, R., VACARI, I. **Adopting Agile Methods in the Public Sector: A Systematic Literature Review**, Twenty-Seventh International Conference on Software Engineering and Knowledge Engineering (SEKE 2015), Pittsburgh, p. 709-714, 2015

PROJECT MANAGEMENT INSTITUTE. **Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos**. 3a. ed. Four Campus Boulevard, Newtown Square, PA 19073- 299 EUA: [s.n.], 2004

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

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**, 7. Ed., São Paulo: Bookman, 2009

RANDOLPH, J. **A guide to writing the Dissertation Literature Review**, Practical Assessment, Research & Evaluation, 2009

RICO, D. F., **Agile Record**, “The Necessity of New Contract Models for Agile Project Management”, 2011

SCHWABER, K. **Agile project management with Scrum**, Microsoft Press, 2004

SCHWABER, K., SUTHERLAND, J. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game**, Scrum.org, 2011

STANDISH Group, **CHAOS Manifesto**, 2002

STANDISH Group, **CHAOS Manifesto**, disponível em http://www.versionone.com/assets/img/files/ChaosManifest_2011.pdf, 2011

STANDISH Group, **CHAOS Manifesto**, disponível em <http://www.versionone.com/assets/img/files/ChaosManifesto2013.pdf>, 2013

STEVENS, P. **10 Contracts for your next Agile Software Project**, acessado em novembro de 2015 em <http://developex.com/blog/10-contracts-for-your-next-agile-software-project/>, 2015

SUTHERLAND, J. **Scrum: A Arte de Fazer o dobro do trabalho na metade do tempo**, São Paulo, LeYa, 2014.

SUTHERLAND, J. **Money for Nothing, Change for Free** <http://jeffsutherland.com/Agile2008MoneyforNothing.pdf>, 2008

UCHOA, A. P. **Contratação de desenvolvimento ágil de software pelo governo**, Rio de Janeiro, UFRJ/COPPE, 2013

VERSION ONE **Report: The State of Agile**, disponível em <http://info.versionone.com/state-of-agile-development-survey-ninth.html>, 2015

VAZQUES, C. E., SIMÕES, G. S., ALBERT, R. M. **Análise De Pontos De Função**, 2004