



Universidade Federal da Paraíba
Centro de Informática
Departamento de Informática
Programa de Pós-Graduação em Informática

Uma Abordagem para Seleção de Equipes Tecnicamente Qualificadas para Implementação de Projetos de Software

Vinicius Souza dos Santos

João Pessoa – PB
Janeiro de 2014



Universidade Federal da Paraíba
Centro de Informática
Departamento de Informática
Programa de Pós-Graduação em Informática

Uma Abordagem para Seleção de Equipes Tecnicamente Qualificadas para Implementação de Projetos de Software

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba como requisito parcial para obtenção do título de Mestre em Informática

Orientador: Glêdson Elias da Silveira

Orientando: Vinicius Souza dos Santos

João Pessoa – PB

Janeiro de 2014

S237u Santos, Vinicius Souza dos.

Uma abordagem para seleção de equipes tecnicamente qualificadas para implementação de projetos de software / Vinicius Souza dos Santos.- João Pessoa, 2014.

153f. : il.

Orientador: Glêdson Elias da Silveira

Dissertação (Mestrado) – UFPB/CI

1. Informática. 2. Alocação de equipes. 3. Seleção de equipes. 4. Software distribuído – desenvolvimento. 5. Lógica Fuzzy. 6. Representação de conhecimento.

UFPB/BC

CDU: 004(043)

RESUMO

Devido às necessidades atuais e grande demanda de produtos de *software* com cada vez mais qualidade, várias abordagens para desenvolvimento de *software* foram desenvolvidas. Entre estas, as Linhas de Produtos de *Software* surgiram como uma abordagem promissora para a melhoria da qualidade de *software* e redução de custos e prazos devido a sua alta capacidade de reúso. Outra abordagem promissora que tem sido adotada pelas organizações, como consequência da globalização, é o Desenvolvimento Distribuído de *Software*, pois a partir dela é possível encontrar mão de obra qualificada e custos mais atrativos em empresas espalhadas pelo mundo. Ao adotar conjuntamente essas abordagens, especialistas de domínio podem ser encontrados em várias equipes terceirizadas ou mesmo equipes de subsidiárias ou filiais de todo o mundo. Todavia, mesmo utilizando-se de abordagens que visam melhorar a qualidade dos produtos de *software* criados, não se pode garantir a qualidade dos mesmos se as equipes não possuem os conhecimentos necessários e específicos para implementação dos diversos módulos de *software*. Assim, surge uma questão primordial: quais equipes são tecnicamente mais qualificadas para desenvolver cada módulo identificado num projeto de *software*? Considerando apenas dez equipes e dez módulos de *software*, é possível identificar 10^{10} combinações entre módulos e equipes, ou seja, 10.000.000.000 de arranjos possíveis. Claramente, esse é um processo complexo para ser realizado pelo gerente de projetos utilizando apenas sua experiência, uma vez que um grande número de combinações pode ser identificado com um número relativamente pequeno de equipes e módulos de *software*. Assim, qualquer decisão *ad-hoc* tomada pode ser complexa, ineficiente e sujeita a erros.

A fim de fornecer suporte para escolha de equipes distribuídas de desenvolvimento de *software*, podendo inclusive o projeto estar integrado em um cenário de Linha de Produtos de *Software*, este trabalho apresenta uma abordagem de apoio à decisão que identifica um conjunto de equipes qualificadas tecnicamente para o desenvolvimento de cada módulo de *software*, baseado em requisitos técnicos dos módulos e capacidades técnicas das equipes. A abordagem proposta está estruturada em quatro etapas. A primeira tem a intenção de produzir um artefato que representa os requisitos técnicos esperados para desenvolver cada módulo de *software*. Com base em tais exigências, a segunda etapa tem por objetivo representar as capacidades técnicas de todas as equipes candidatas. Em seguida, a terceira etapa visa à definição de uma política que rege a seleção de equipes. Por fim, a quarta etapa cruza as informações coletadas nas fases anteriores em um algoritmo de lógica *fuzzy* e realiza a seleção das equipes. Como saída, para cada módulo de *software* é gerado uma recomendação que indica o conjunto de equipes candidatas que possuem as qualificações técnicas mais adequadas para implementar os módulos de *softwares* segundo a política de seleção adotada.

ABSTRACT

Due to the current needs and high demand for software products with more quality, various approaches for software development have been proposed. Among such approaches, Software Product Line has emerged as a promising approach for improving software quality and reducing costs and development time due to its high capacity for promoting software reuse. As another promising approach, Distributed Software Development has been adopted by organizations as a consequence of globalization, making possible to find skilled software professionals and more attractive costs in organizations spread throughout the world. By adopting such approaches together, domain experts can be found in several outsourcing or even insourcing software development teams around the world. However, even adopting such approaches to improve the software quality, is not possible to guarantee the software quality if software development teams do not have the necessary and specific knowledge to implement each module. So, a key question arises: which teams are more technically qualified to develop each module identified in a software project? Considering only ten software development teams and ten software modules, it is possible to identify 10^{10} combinations of modules and teams, in other words, 10.000.000.000 possible arrangements. Clearly, this is a complex process to be done by project managers based on their experience, since a large number of combinations can be identified with a relatively small number of teams and software modules. Thus any ad-hoc decision making can be complex, inefficient and error-prone.

In order to provide support for recommending qualified global software teams in Software Product Line projects, this dissertation presents a decision support approach that identifies a set of technically qualified software development teams for each software module. To achieve such a goal, the proposed approach is based on technical requirements of software modules and technical capabilities of software development teams. The proposed approach is structured in four stages. The first stage is intended to produce an artifact that represents the technical requirements expected to develop each software module. Based on such requirements, the second stage is intended to represent the technical capabilities of all candidate software development teams. Then, the third stage aims to define a policy that guides the selection of such teams. Finally, the fourth stage maps the information collected in earlier stages into a fuzzy logic algorithm and performs the selection of technically qualified teams. As output, for each software module, a recommendation is generated that indicates a set of candidate teams that have the most appropriate technical skills to implement the software module.

SUMÁRIO

RESUMO	III
ABSTRACT	IV
SUMÁRIO	V
LISTA DE FIGURAS	VII
LISTA DE TABELAS	VIII
LISTA DE ABREVIATURAS E SIGLAS	XI
1. INTRODUÇÃO	12
1.1 Motivação	14
1.2 Objetivo do trabalho	16
1.3 Organização do texto	17
2. FUNDAMENTAÇÃO TEÓRICA	18
2.1 Gerenciamento e seleção de recursos humanos	18
2.2 Lógica <i>fuzzy</i>	23
2.3 Considerações finais	26
3. FRAMEWORK DE RECOMENDAÇÃO DE ALOCAÇÃO DE EQUIPES	28
3.1 Composição do <i>Framework</i>	29
3.2 Análise da Arquitetura	30
3.3 Avaliação técnica	31
3.4 Avaliação não técnica	32
3.5 Avaliação do desempenho das equipes	33
3.6 Considerações Finais	34
4. ABORDAGEM DE SELEÇÃO TÉCNICA DE EQUIPES	35
4.1 Descrição dos módulos	37
4.2 Descrição técnica das equipes	40
4.3 Definição de políticas de seleção	49
4.3.1 Política de qualificação equivalente	52
4.3.2 Política de equipes mais hábeis	53
4.3.3 Política de qualificação padrão	54
4.3.4 Política de provisão de treinamento	55
4.4 Avaliação técnica das equipes	56
4.5 Cálculo do valor de reputação	62
4.6 Considerações Finais	63
5. ESTUDO DE CASOS	64
5.1 Configurações utilizadas	65
5.2 Linha de Produto de Comércio Eletrônico	67

5.2.1 Primeira Iteração (Engenharia de domínio)	67
5.2.1.1 Estudo com Equipes do Conjunto A	70
5.2.1.2 Estudo com Equipes do Conjunto B	76
5.2.2 Segunda iteração (Engenharia de aplicação)	83
5.2.2.1 Estudo com Equipes do Conjunto A	86
5.2.2.2 Estudo com Equipes do Conjunto B	90
5.3 Multi-MOM.....	93
5.3.1 Estudo com Equipes do Conjunto A	96
5.3.2 Estudo com Equipes do Conjunto B	100
5.4 Considerações Finais	104
6. TRABALHOS RELACIONADOS.....	105
6.1. ICARE	105
6.2. WebAPSEE-Planner	106
6.3. Apoio à alocação de RH baseado em satisfação de restrições	107
6.4. Task Allocation Optimizer	109
6.5. MRES.....	110
6.6. Multi-Criteria Task Assignment in Workflow Management Systems	111
6.7. Considerações Finais	111
7. CONSIDERAÇÕES FINAIS.....	117
7.1. Contribuições	118
7.2. Limitações.....	119
7.3. Trabalhos Futuros.....	120
REFERÊNCIAS	122
APÊNDICE A	127
APÊNDICE B	138
APÊNDICE C.....	143

LISTA DE FIGURAS

Figura 1. Representação de conjunto <i>fuzzy</i>	23
Figura 2. Blocos de um sistema <i>fuzzy</i> [Jantenz, 98].	24
Figura 3. <i>Fuzzificação</i> de valores <i>crisp</i> em termos vagos.	25
Figura 4. Visão geral das fases do <i>framework</i>	30
Figura 5. Etapas referentes à seleção técnica de equipes.	35
Figura 6. Conjuntos <i>Fuzzy</i> de representação dos módulos de <i>software</i>	38
Figura 7. Exemplo de questionário.....	42
Figura 8. Conjuntos <i>Fuzzy</i> de representação das habilidades das equipes.	48
Figura 9 Conjuntos <i>Fuzzy</i> de representação da adequabilidade das equipes.....	50
Figura 10. Cálculo de adequabilidade.	57
Figura 11. Configurações dos conjuntos <i>fuzzy</i>	65
Figura 12. Visão dos componentes variantes	68
Figura 13. Arquitetura de domínio da LPS E-Commerce	69
Figura 14. Arquitetura da aplicação B2B	83
Figura 15. Arquitetura da aplicação B2C.	84
Figura 16. Componentes opcionais da LPS de E-Commerce.....	84
Figura 17. Arquitetura de domínio do Multi-MOM.	94

LISTA DE TABELAS

Tabela 1. Tabela de implementação.	38
Tabela 2. Exemplo de política de seleção	51
Tabela 3. Política de qualificação equivalente.	53
Tabela 4. Política de equipes mais hábeis.	54
Tabela 5. Política de qualificação padrão.	54
Tabela 6. Política de provisão de treinamento.	55
Tabela 7. Primeira versão do <i>Mapemaneto equipes × módulos</i>	59
Tabela 8. Versão final do <i>Mapemaneto equipes × módulos</i>	61
Tabela 9. Tabela de implementação de M0 (E-commerce – domínio).	69
Tabela 10. Tabela de implementação de M1 (E-commerce – domínio).	70
Tabela 11. Tabela de implementação de M2 (E-commerce – domínio).	70
Tabela 12. Valores <i>Crisp</i> das habilidades técnica das equipes (E-commerce - domínio).	71
Tabela 13. Valores <i>Crisp</i> da complexidade esperada das tecnologias (E-commerce – domínio).	71
Tabela 14. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto A).	72
Tabela 15. Versão final do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto A).	72
Tabela 16. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - equipes mais hábeis - Conjunto A).	75
Tabela 17. Versão final do Mapeamento equipes × módulos (E-commerce – domínio – equipes mais hábeis - Conjunto A).	75
Tabela 18. Valores <i>Crisp</i> das habilidades técnica das equipes (E-commerce – domínio – Conjunto B).	77
Tabela 19. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto B).	78
Tabela 20. Versão final do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto B).	78
Tabela 21. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - equipes mais hábeis - Conjunto B).	80

Tabela 22. Versão final do Mapeamento equipes × módulos (E-commerce – domínio – equipes mais hábeis - Conjunto B).	81
Tabela 23. Tabela de implementação de M1 (E-commerce – aplicação).	85
Tabela 24. Tabela de implementação de M2 (E-commerce – aplicação).	85
Tabela 25. Tabela de implementação de M3 (E-commerce – aplicação).	86
Tabela 26. Valores <i>Crisp</i> das habilidades técnica das equipes (E-commerce - aplicação).	86
Tabela 27. Valores <i>Crisp</i> da complexidade esperada das tecnologias(E-commerce - aplicação).	87
Tabela 28. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de equipes mais hábeis – Conjunto A).	87
Tabela 29. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – equipes mais hábeis - Conjunto A).	87
Tabela 30. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de provisão de treinamento – Conjunto A).	89
Tabela 31. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – política de provisão de treinamento - Conjunto A).	89
Tabela 32. Valores <i>Crisp</i> das habilidades técnica das equipes (E-commerce - aplicação).	90
Tabela 33. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de equipes mais hábeis – Conjunto B).	91
Tabela 34. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – equipes mais hábeis - Conjunto B).	91
Tabela 35. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de provisão de treinamento – Conjunto B).	92
Tabela 36. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – política de provisão de treinamento - Conjunto B).	93
Tabela 37. Tabela de implementação de M0 (Multi-MOM)	95
Tabela 38. Tabela de implementação de M1 (Multi-MOM)	95
Tabela 39. Tabela de implementação de M2 (Multi-MOM)	95
Tabela 40. Tabela de implementação de M3 (Multi-MOM)	95
Tabela 41. Tabela de implementação de M4 (Multi-MOM)	96
Tabela 42. Valores <i>Crisp</i> das habilidades técnica das equipes.	96
Tabela 43. Valores <i>Crisp</i> da complexidade esperada das tecnologias(Multi-MOM).	96

Tabela 44. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação equivalente – Conjunto A).	97
Tabela 45. Versão final do Mapeamento equipes × módulos (Multi-MOM – política de qualificação equivalente - Conjunto A).	97
Tabela 46. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação padrão – Conjunto A).	99
Tabela 47. Versão final do Mapeamento equipes × módulos (Multi-MOM – política de qualificação padrão - Conjunto A).	99
Tabela 48. Valores <i>Crisp</i> das habilidades técnica das equipes (Conjunto B).	100
Tabela 49. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação equivalente – Conjunto B).	101
Tabela 50. Versão final do Mapeamento equipes × módulos (Multi-MOM – política de qualificação equivalente - Conjunto B).	101
Tabela 51. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação padrão – Conjunto B).	102
Tabela 52. Versão final do Mapeamento equipes × módulos (Multi-MOM – política de qualificação padrão - Conjunto B).	103
Tabela 53. Comparação dos trabalhos relacionados.....	115

LISTA DE ABREVIATURAS E SIGLAS

DBC	Desenvolvimento Baseado em Componentes
DDS	Desenvolvimento Distribuído de <i>Software</i>
EA	Engenharia de Aplicação
ED	Engenharia de Domínio
LPS	Linhas de Produto de <i>Software</i>
POO	Programação Orientada a Objetos
RH	Recursos Humanos
SOA	Arquitetura Orientada a Serviço
SRE	Sistema de Recomendação de Especialista
PSEE	Ambiente de Engenharia de <i>Software</i> Centrado em Processos
WfMS	Sistema de Gerenciamento de <i>Workflow</i>

Capítulo 1

Introdução

Com a globalização, o mercado de *software* vem apresentando uma demanda muito grande por sistemas de *software*, o que por sua vez gera grande concorrência e colaborações entre empresas a fim de atender às necessidades do mercado [Herbsleb, et al., 2001]. Assim, cada vez mais, é exigido que os sistemas de *software* desenvolvidos tenham menos erros, sejam mais flexíveis e entregues o mais rápido possível. Ao mesmo tempo, os desenvolvedores tentam atingir níveis mais altos de produtividade e qualidade de seus produtos. Para atender a estas necessidades, as empresas vêm adotando técnicas de reúso de *software* a fim de agilizar o processo de construção do *software* e diminuir seus custos. A Engenharia de *Software* vem ao longo dos anos desenvolvendo várias técnicas de reúso como o uso de bibliotecas, Programação Orientada a Objetos (POO), *Design Patterns*, Desenvolvimento Baseado em Componentes (DBC), Arquitetura Orientada a Serviço (SOA), entre outras. Dentre tantas abordagens de reúso, Linha de Produtos de *Software* (LPS) vem se apresentado como uma abordagem de destaque, pois é capaz de promover um reúso sistemático e previsível por tratar o reúso como parte fundamental do processo de construção do *software* [Clements, et al., 2002].

Tradicionalmente, os processos de LPS são divididos em duas etapas: Engenharia de Domínio (ED) e Engenharia de Aplicação (EA) [Linden, et al., 2007]. Na etapa de ED é construído o núcleo da LPS, onde são planejados quais produtos devem ser produzidos pela linha de produtos. Também é construída a arquitetura de referência, que contém os componentes comuns a todos os produtos e alguns componentes específicos de alguns produtos previamente identificados. Os produtos são instanciados na fase de EA, onde são levantadas as necessidades específicas de um produto e os componentes são utilizados para instanciá-lo. Caso o núcleo da linha de produto não ofereça todas as funcionalidades requisitadas pelo produto, novos componentes podem ser implementados e integrados ao núcleo da LPS [Hunt, M. J, 2006].

Percebe-se que, em LPS, o reúso de *software* é intrínseco à abordagem, ou seja, não se pode implementar uma LPS sem reúso, pois a própria abordagem está fundamentada no conceito de reutilização. Por conta dessa característica e das consequentes vantagens como redução de custos, redução de *time-to-market*, aumento da qualidade entre outros [Pohl, et al., 2005], esta abordagem vem ganhando adesão perante as empresas. Todavia, iniciar uma LPS não é uma tarefa fácil visto que um bom planejamento e alto investimento são necessários para se criar uma arquitetura de referência. Além disso, para que a linha tenha sucesso é necessário que a arquitetura de referência reflita bem os conceitos envolvidos no domínio da aplicação. Portanto, *experts* no domínio da aplicação são necessários. Contudo, nem sempre é fácil dispor de tais *experts* numa mesma empresa.

Outra abordagem que vêm sendo utilizada no cenário globalizado para atender as demandas do mercado é o Desenvolvimento Distribuído de *Software* (DDS). No DDS, equipes espalhadas pelo globo trabalham de forma colaborativa para o desenvolvimento de um projeto de *software* [Marquardt, et al., 2001]. Assim como a LPS, essa abordagem também produz uma série de benefícios, tais como: redução de custo, redução do *time-to-market*, aumento da qualidade de *software*, etc [Cataldo, 2008] [Sangwan, et al., 2006]. Um dos grandes benefícios do DDS se mostra na possibilidade de achar equipes mais qualificadas e *experts* no domínio de uma aplicação em um lugar remoto [Karolak, 1999]. Percebe-se então que ambas as abordagens possuem características e vantagens complementares e que podem ser utilizadas conjuntamente para agregar ainda mais vantagens ao processo de desenvolvimento de *software*. De fato, Paulish [Paulish, 2003] reitera essa tendência quando afirma que a própria técnica de LPS oferece suporte ao desenvolvimento de produtos para mercados globais.

Embora em cenários de DDS geralmente se encontrem profissionais especializados e mais qualificados, apenas isso não basta para garantir o sucesso de um projeto de *software*. A boa comunicação entre os profissionais participantes de um mesmo projeto é um dos grandes desafios dessa abordagem, como evidenciado por [Carmel, 1999]. A falta de comunicação ou uma comunicação mal feita entre as equipes pode gerar uma série de problemas e desentendimentos que são refletidos na qualidade dos *softwares* produzidos, já que o bom desempenho das pessoas envolvidas no projeto é um dos motivos de sucesso de um processo de desenvolvimento de *software*. Pressman [Pressman, 2004] reforça esta ideia mencionando que a gestão de projetos de *software* só poderá ter sucesso se o gerente de projetos não esquecer que este é um empreendimento intensamente humano e complementa afirmando que

a qualidade do fator humano num projeto de *software* apresenta um impacto direto na qualidade dos *softwares* produzidos.

Dado este cenário, percebe-se que a correta seleção dos recursos humanos (RH) que participarão de um projeto é fundamental para obter as vantagens descritas, pois as equipes distribuídas: *i)* devem possuir as qualificações técnicas necessárias e conhecimento de domínio adequado para a construção de uma LPS e seus produtos; *ii)* devem possuir a maior afinidade possível entre si, seja por falarem a mesma língua, por disporem do mesmo horário de trabalho ou qualquer outro motivo que amenize os problemas de comunicação advindos do DDS. A seleção de RH apresenta um impacto direto na eficiência do desenvolvimento do projeto [Ebert, et al., 2001]. Assim, é de grande importância a existência de uma abordagem de suporte a decisão que guie os gerentes de projetos na seleção de RH adequado para as atividades de um projeto de *software*.

Neste sentido, o objetivo principal deste trabalho é fornecer uma abordagem que lide com os aspectos de qualificação técnica das equipes (item *i*), ou seja, recomendação de equipes distribuídas tecnicamente qualificadas para a implementação de projetos de *software*, inclusive projetos de LPS. Vale salientar que este trabalho é parte integrante de um *framework* de alocação de equipes distribuídas para implementação de projetos de LPS [Santos et al. 2010] [Pereira, et al., 2010] que dentre suas várias etapas lida com os aspectos de qualificação técnica das equipes (item *i*), bem como os aspectos não-técnicos (item *ii*) que impactam na qualidade da comunicação entre as equipes. É importante destacar que os aspectos não-técnicos não fazem parte do escopo deste trabalho, bem como a independência do fato do projeto estar diretamente inserido no contexto de LPS e DDS (contexto principal do *framework* de alocações de equipes distribuídas para implementação de projetos de LPS), visto que a abordagem de seleção proposta pode ser utilizada com qualquer tipo de projeto de *software*.

1.1 Motivação

Segundo pesquisa realizada em 2009 pelo *Standish Group* [Standish Group, 2009], foi constatado que apenas 32% dos projetos analisados foram realizados com sucesso. Em 44% foi verificado atraso, estouro de orçamento e/ou foram entregues com menos funcionalidades do que as planejadas. Por fim, 24% foram cancelados ao longo do desenvolvimento. É evidente que a maioria dos projetos de *software* sofreu algum tipo de dano. Muitas vezes esses danos são acarretados pela falta de gerenciamento das pessoas, como afirma Wohlin

[Wohlin, 2004] quando comenta que o principal fator envolvido no problema são as pessoas que desenvolvem o *software*.

Segundo a norma NBR ISO 10006 (2000) [ABNT, 2000], as pessoas são determinantes da qualidade e do sucesso do projeto. A norma também diz que se deve selecionar pessoal suficiente, com competências apropriadas que correspondam às necessidades do projeto. Geralmente cabe ao gerente de projetos fazer parte do processo de seleção de RH para as atividades de um processo de desenvolvimento de *software*. Isso é confirmado segundo a mesma norma quando esta determina que o gerente de projetos deve fazer parte da seleção dos membros-chave para o desenvolvimento do projeto.

Apesar do DDS apresentar-se como uma solução para prover equipes qualificadas e *experts* de domínio na construção de projetos de LPS e suas várias aplicações, geralmente há muitos componentes de software a serem implementados. Assim, em um cenário que possua apenas dez atividades de implementação a serem realizadas e dez equipes de desenvolvimento disponíveis, o número de combinações possíveis de seleção de quais equipes devem realizar quais tarefas chega a 10^{10} ou 10 bilhões. Nesse universo de soluções possíveis, é certo que há tanto soluções boas como soluções péssimas que atrasam o projeto e que comprometem a qualidade do produto de *software*.

Para um gerente de projetos é impossível analisar cada solução em busca das mais adequadas às suas necessidades sem nenhum tipo de apoio, e, por conta disso, muitas vezes ele baseia suas escolhas apenas em sua experiência e instinto. Dessa forma, boas soluções podem vir a ser desprezadas, acarretando em perdas significativas para o projeto de *software*.

A dificuldade desse processo é também agravada pela sua própria natureza, ou seja, para que a seleção de equipes tecnicamente qualificadas possa acontecer é necessário que primeiro os conhecimentos técnicos de cada equipe sejam capturados para assim poderem ser avaliados. Contudo, representar tal conhecimento não é trivial já que o mesmo se apresenta como algo bastante subjetivo, sendo insuficiente representar com dados booleanos, como possui ou não conhecimento, e também sendo contra intuitivo representar através de números que expressam uma precisão inexistente na natureza de quantificação de conhecimento.

Assim é evidente que o gerente de projetos deve dispor de algum processo de apoio à decisão para auxiliá-lo quanto ao processo de seleção de RH. Vale salientar que um sistema de apoio à decisão não deve tomar a decisão pelo gerente, mas lhe prover informações que o

auxiliem, deixando a cargo do gerente, com sua experiência, a decisão final [Rus, et al., 2002].

1.2 Objetivo do trabalho

O objetivo principal desse trabalho é desenvolver uma forma de auxílio aos gerentes de projetos no que diz respeito à seleção de equipes distribuídas tecnicamente qualificadas para atividades de implementação em um projeto de *software*, inclusive projetos de LPS. Mais especificamente, a abordagem proposta visa auxiliar o gerente de projetos nos seguintes aspectos: *i)* caracterizar as necessidades de cunho técnico para implementação de módulos de *software* identificados num projeto de *software*; *ii)* caracterizar os conhecimentos e atributos técnicos de equipes globalmente distribuídas; *iii)* caracterizar políticas de seleção que considerem as necessidades específicas dos diferentes projetos de *software*; *iv)* avaliar a seleção das equipes para o desenvolvimento dos módulos de *software* através de uma determinada política de seleção. Para a etapa de caracterização das equipes são levados em consideração atributos que tenham relevância com o conhecimento técnico que as equipes possuem, como habilidade em determinada tecnologia e conhecimento acerca de um dado domínio. Quanto aos módulos de *software*, são caracterizados os requisitos técnicos esperados que uma equipe deve apresentar para ser capaz de desenvolvê-los. Como o projeto pode se tratar de um projeto de LPS, é importante que conhecimentos acerca do domínio de aplicação também sejam levados em consideração. Para lidar com a complexidade de representação de conhecimento comentada anteriormente, conjuntos *Fuzzy* [Zadeah, 1965] são utilizados, possibilitando que a subjetividade inerente ao processo seja apropriadamente levada em consideração.

A partir dos atributos técnicos levantados é proposto um processo de avaliação da qualificação técnica das equipes em relação aos módulos de *software* baseando-se em inferência provida pela utilização de um algoritmo de Lógica Nebulosa (*Fuzzy*) [Zadeah, 1965] [Cox, 1995]. Para tal inferência são propostas regras de seleção determinadas por políticas comuns em um projeto de *software*. Tais regras são responsáveis por gerar as recomendações, auxiliando o gerente de projetos no processo de seleção das equipes. Apesar de políticas de seleção serem definidas previamente neste trabalho, com base em recomendações da literatura [Pressman, 2009], o gerente de projetos deve ter a liberdade de criar novas regras e alterar políticas existentes. Assim, esta abordagem pode se adequar às peculiaridades apresentadas pelos diversos projetos de *software*. Ao final é gerada uma lista

de equipes tecnicamente qualificadas para a implementação de cada módulo de *software* identificado no projeto, ordenado pelo nível de adequabilidade das mesmas em relação aos módulos de *software* aos quais foram selecionadas, de acordo com a política de seleção utilizada.

1.3 Organização do texto

O trabalho está estruturado da seguinte forma. No Capítulo 2 são apresentados os conceitos básicos para o entendimento da abordagem proposta neste trabalho, introduzindo ao leitor fundamentos sobre gerenciamento e seleção de recursos humanos e lógica *fuzzy*, conceitos que serão explorados nos capítulos posteriores. O Capítulo 3 apresenta o *framework* do qual este trabalho faz parte, abordando cada fase do mesmo (recomendação de módulos de *software*, recomendação técnica de equipes e recomendação de alocação de equipes), indicando onde o presente trabalho se encontra no *framework*. No Capítulo 4 é apresentada a proposta deste trabalho, mostrando os elementos que a compõe: atributos técnicos utilizados para mensurar conhecimento, a forma de extrair informações das equipes sobre o conhecimento das mesmas, cálculo para atribuir o conhecimento das equipes acerca das tecnologias identificadas para produção de módulos de *software*, construção de regras de seleção e cálculo de adequabilidades das equipes para implementação dos módulos de *software*. O Capítulo 5 apresenta estudos de casos que realizam a avaliação do trabalho, mostrando os resultados obtidos e analisando o que significam, descrevendo para cada resultado os pontos mais importantes. O Capítulo 6 apresenta os trabalhos relacionados e faz uma comparação entre eles e o presente trabalho, evidenciando os pontos fortes e fracos das abordagens. Finalmente, o Capítulo 7 apresenta as considerações finais, as contribuições, as limitações e as possibilidades de trabalhos futuros.

Capítulo 2

Fundamentação teórica

Neste capítulo são descritos conceitos fundamentais relacionados ao desenvolvimento deste trabalho. Inicialmente são apresentados conceitos acerca de gerenciamento e seleção de recursos humanos, visto que o trabalho visa à seleção de equipes. Em seguida, são apresentados os conceitos fundamentais sobre lógica *fuzzy* e sua utilização em sistemas computacionais, visto que os algoritmos utilizados na construção da solução proposta para recomendações de equipes qualificadas se baseiam na utilização da lógica *fuzzy*.

2.1 Gerenciamento e seleção de recursos humanos

A seleção de pessoas para o desenvolvimento de projetos de *software* se caracteriza como uma atividade relacionada ao gerenciamento de RH. A gerência de RH de um projeto inclui os processos que organizam e gerenciam a equipe do projeto [PMI, 2004]. Para que as organizações consigam obter sucesso em seus projetos de *software*, elas precisam gerenciar as pessoas alocadas de forma eficiente, visto que a qualidade dos produtos a serem desenvolvidos depende diretamente da capacidade técnica e competências destes.

Segundo Wastell [Wastell et al., 1999], no processo de desenvolvimento de *software*, a interação entre os membros da equipe é crucial para a obtenção do sucesso. Ele ainda comenta que o entendimento do processo de desenvolvimento de *software* tem como foco o esforço do gerente de projetos em definir quais tarefas devem ser executadas, por quem devem ser executadas, quando devem ser executadas e por quais ferramentas devem ser apoiadas.

O PMI [PMI 2004] reconhece o gerenciamento de RH como uma das áreas de conhecimento envolvidas na gerência de projetos. Como o gerenciamento de RH de qualquer projeto inclui tarefas que têm por objetivo organizar e gerenciar a equipe do projeto, o PMI define os seguintes processos:

- **Planejamento de recursos humanos** – Identificação e documentação de funções, responsabilidades e relações hierárquicas do projeto, bem como criação de plano de gerenciamento de pessoal;
-

- **Contratação ou mobilização da equipe do projeto** – Obtenção dos recursos humanos necessários para terminar o projeto;
- **Desenvolvimento da equipe do projeto** – Melhoria de competências e interação de membros da equipe para aprimorar o desempenho do projeto;
- **Gerenciamento da equipe do projeto** – Acompanhamento do desempenho de membros da equipe, fornecimento de *feedback*, resolução de problemas e coordenação de mudanças para a melhoria do desempenho do projeto.

A norma NBR ISO 10006 (2000) [ABNT, 2000], que tem por finalidade definir normas para garantir a qualidade do gerenciamento de projetos, também define processos relacionados aos recursos humanos. Estes processos são:

- **Definição da estrutura organizacional do projeto** – Definição de uma estrutura organizacional para o projeto, baseada no atendimento às necessidades de projeto, incluindo a identificação das funções e definindo autoridades e responsabilidades;
- **Alocação da equipe** – Seleção e nomeação de pessoal suficiente com a competência apropriada para atender as necessidades do projeto;
- **Desenvolvimento da equipe** – Desenvolvimento de habilidades individuais e coletivas para aperfeiçoar o desempenho do projeto.

Schnaider [Schnaider, 2003] define um processo de gerenciamento de RH composto por três sub-processos: Planejamento da Alocação de Recursos Humanos, Monitoramento da Alocação de Recursos Humanos e Avaliação de Recursos Humanos. O Planejamento da Alocação de Recursos Humanos, como o próprio nome sugere, tem como objetivo planejar a alocação dos recursos humanos necessários para a execução do projeto de *software*. Nesse sub-processo são definidas as seguintes atividades:

- **Definição de Perfis de Competência** – Definir perfis de competência necessários à execução de cada atividade do projeto. Um perfil de competência consiste na definição de um conjunto de atributos que incluem conhecimentos, habilidades, formações acadêmicas, etc., visando à seleção e posterior alocação dos profissionais mais adequados para o projeto;
 - **Seleção de Equipe** – Selecionar quais profissionais serão alocados ao projeto. Os recursos humanos são selecionados para alocação de acordo com cada perfil de competência especificado para o projeto;
-

- **Contratação de Profissionais** – Tem por objetivo solicitar à área de recursos humanos a contratação de um profissional a ser alocado em um determinado projeto de *software*. Vale salientar que esta atividade é executada apenas quando não existem profissionais na organização que possuam o perfil de competência exigido pelo projeto de *software*, seja porque profissionais com este perfil já estejam alocados em outros projetos ou por realmente não existir na empresa alguém que condiz com o perfil desejado;
- **Capacitação de Profissionais** – Esta atividade é uma alternativa à contratação de profissionais. Caso não haja profissionais disponíveis com o perfil desejado, pode-se ao invés de contratar novos profissionais, capacitar os existentes para que estes adquiram os conhecimentos necessários que são exigidos pelos perfis de competência.

Já para Dingsoyr [Dingsoyr et al., 2001], os processos de gerenciamento de recursos humanos consistem em:

- **Busca de competências necessárias** – Definição das competências em si, procura dos profissionais da organização que possuem as competências definidas e seleção dos profissionais desejados;
- **Alocação de recursos humanos** – Alocação, deslocamento e retirada de profissionais;
- **Desenvolvimento de competências** – Avaliação dos recursos humanos para capacitação dos mesmos.

Note que todas as abordagens de gerenciamento de recursos humanos citadas tratam da **identificação de competências ou habilidades** dos profissionais e posterior **seleção** destes, ambas atividades foco deste trabalho. Segundo [Schnaider, 2003], a identificação de competências é o primeiro passo a ser feito para cada profissional que poderá vir a ser selecionado para participação de um projeto.

Observando a norma NBR ISO 10006 [ABNT, 2000], percebe-se que esta recomenda que a competência necessária dos recursos humanos seja definida em termos de educação, conhecimento e experiência. A norma ainda sugere que, para a identificação das competências dos profissionais a serem incluídos no projeto, devem ser levados em consideração aspectos relacionados aos interesses pessoais, relacionamentos interpessoais e pontos fortes e fracos. Essa recomendação é feita, pois o conhecimento das características e experiências pessoais

pode ajudar na identificação da melhor distribuição de responsabilidades entre os membros da organização do projeto.

Dingsoyr [Dingsoyr et al., 2001] recomenda que as competências necessárias ao projeto sejam agrupadas e categorizadas. O agrupamento e categorização das competências facilitam consultas, análises ou modificações posteriores do conjunto de competências exigidas num projeto de *software*. Ele ainda sugere que para cada competência identificada seja atribuído um nível de graduação e um nível mínimo necessário de conhecimento exigido. O nivelamento das competências permite que os diferentes graus de *expertise* existentes para uma mesma competência possam ser padronizados.

Plekhanova [Plekhanova, 1999], espelhando-se no *Rational Unified Process* (RUP), sugere que além das competências, os cargos das pessoas também sejam considerados como critérios para a sua alocação em atividades de um processo.

Silva [Silva, 2007] faz uma observação interessante quando comenta que se um projeto é similar a algum projeto já desenvolvido pela organização, seria útil ter acesso às listas de competências identificadas para este último, pois estas listas podem ajudar o gerente de projetos na identificação das competências necessárias para o seu próprio projeto. Ele ainda ressalta que as lições aprendidas com alocações de profissionais em outros projetos, sejam estas bem ou mal sucedidas, são de fundamental importância, pois evitam que erros anteriores sejam repetidos e que casos de sucesso não sejam esquecidos.

Quanto à seleção dos profissionais, uma vez que a seleção de recursos humanos representa um fator tão determinante nos processos de desenvolvimento de *software*, a atribuição de tarefas adequadas às capacidades de cada profissional é de grande importância para que os níveis de produtividade e qualidade desejados em cada atividade possam ser atingidos [Silva, 2007]. A seleção deve levar em conta as características de cada projeto, para que assim os profissionais sejam selecionados visando à melhor adequação destes com as atividades que devem ser desempenhadas [Schnaider, 2003]. Atribuir pessoas com competências inferiores às exigidas pela atividade pode fazer com que esta não seja concluída ou que sofra atrasos ou termine com baixa qualidade. Por outro lado, selecionar pessoas com competências muito além das exigidas pela atividade pode resultar em um aumento desnecessário no custo do projeto ou até mesmo na frustração do profissional por pensar que está sendo subutilizado dentro do projeto [Barreto, 2005].

Segundo a norma NBR ISO 10006 [ABNT, 2000], é recomendado que o gerente do projeto tome como base as seguintes informações para seleção dos recursos humanos da organização que estão disponíveis no período desejado: as descrições das tarefas; as competências necessárias; e as respectivas referências de experiências anteriores. Ainda de acordo com a norma, sempre que dificuldades na seleção de recursos humanos sejam identificadas devido a requisitos de competências, recomenda-se que seja estabelecido um intervalo de tempo suficiente para recrutamento e treinamento de profissionais adicionais. Assim, é entendido que o gerente poderá, além de contratar novos profissionais para a empresa, contratar profissionais terceirizados ou buscar profissionais de subsidiárias da mesma empresa em outro local. Dentre as técnicas mais comuns no que diz respeito à seleção de pessoas podem ser mencionadas a análise de currículos, entrevistas, dinâmicas de grupo, reuniões, análise de lições aprendidas e páginas amarelas corporativas. Inclusive, segundo Basili [Basili et. al., 2001], esta última é um excelente recurso, pois apesar de saber que existe algum profissional capacitado na empresa, às vezes despende-se muito tempo para identificá-lo e encontrá-lo. O uso das páginas amarelas permite organizar e manter controle sobre as competências dos recursos humanos presentes na empresa e onde encontrá-los.

Como pôde ser observado, a seleção de RH é um processo de extrema importância para o sucesso de um projeto. Contudo, a seleção de RH pode se tornar muito complicada sem a existência de algum tipo de apoio à identificação de competências e tomada de decisão de quais recursos devem ser selecionados. O apoio à decisão já foi aplicado com sucesso em várias áreas como logística, manufatura, saúde, etc. [Ruhe, 2003]. Contudo, tomar decisão é normalmente um processo complexo e não sistemático, já que tipicamente se baseia nas experiências pessoais [Rus, et al., 2002]. O apoio à tomada de decisão se apresenta como um dos grandes objetivos da engenharia de *software*, já que tem o intuito de melhorar a habilidade das pessoas em fazer boas decisões técnicas [Herbsleb, et al., 2006] que tragam impacto positivo ao processo de desenvolvimento de *software*.

Esse apoio à decisão se torna ainda mais necessário no processo de seleção de RH, pois além da complexidade já mencionada para a seleção de profissionais, a identificação de competências não é uma tarefa trivial visto a subjetividade das informações contidas nesta atividade. Por exemplo, os processos descritos dizem que as competências e habilidades devem ser identificadas e até mensuradas para que haja uma seleção adequada dos profissionais. Contudo mensurar as capacidades ou habilidades de um profissional é extremamente subjetivo. Visando a lidar com essa subjetividade, o trabalho proposto se utiliza

de uma abordagem de lógica nebulosa ou *fuzzy*, que consegue tratar de forma satisfatória este problema. Assim, a seguinte seção traz alguns conceitos fundamentais para seu entendimento.

2.2 Lógica *fuzzy*

Lógica *Fuzzy* ou Lógica Nebulosa teve seu início num artigo publicado por Zadeah [Zadeah, 1965] onde o mesmo discutia sobre imprecisões do mundo real que não eram lidas adequadamente ao tentar se formalizar um sistema. Desse modo, o autor propôs um conceito de vanguarda chamado conjuntos *fuzzy*. Desde então, sistemas de lógica *fuzzy* vêm sendo adotados em vários problemas onde não existe precisão quanto aos elementos de entrada do sistema ou quando não se conhece completamente o comportamento de um sistema.

O objetivo primordial da lógica nebulosa é generalizar o conceito representado pela teoria dos conjuntos ordinários clássicos, aproximando-os da imprecisão e do aspecto vago do raciocínio humano [Kosko, 1992].

Para exemplificar um problema de imprecisão, suponha um copo com certa quantidade de água presente nele. Para um humano é praticamente impossível dizer com exatidão o valor em mililitros (ml) de água contida num copo simplesmente olhando para ele. Normalmente alguém comentaria que “o copo está meio cheio” ou “o copo está um pouco vazio”, que são valores imprecisos e bastante relativos. Com a utilização de lógica *fuzzy* é possível utilizar estes termos vagos como “meio cheio” ou “pouco vazio” e quantificá-los com valores de pertinência perante certo conjunto.

O conceito fundamental por trás da lógica nebulosa se encontra nos conjuntos de pertinência *fuzzy*. Diferentemente da lógica booleana binária, onde algo sempre é completamente falso ou completamente verdadeiro, estes conjuntos indicam que um elemento pertence a ele em certa proporção, chamada grau de pertinência. Então um dado elemento poderia estar contido tanto no conjunto falso como no conjunto verdadeiro simultaneamente, possuindo um grau de participação efetiva que mensura o quanto este elemento pertence ao conjunto. A representação de um conjunto *fuzzy* é dada como descrito na Figura 1:

$$T = \left\{ \frac{\mu_{C_T}(x_0)}{x_0} + \frac{\mu_{C_T}(x_1)}{x_1} + \dots + \frac{\mu_{C_T}(x_n)}{x_n} \right\}$$

Figura 1. Representação de conjunto *fuzzy*.

Nessa representação, os sinais matemáticos de soma e divisão não expressam suas funcionalidade matemáticas, são apenas separadores dos elementos de entrada e saída. Assim, cada elemento é separado pelo sinal de +. Os “dividendos” x_i representam os elementos de entrada do conjunto, enquanto os “divisores” $\mu_{C_T}(x_i)$ representam o valor de pertinência do elemento no conjunto T , onde μ é chamado de função de pertinência que mapeia o valor do mundo real (chamado valor *crisp*) para um valor de pertinência no conjunto *fuzzy*. Os valores de pertinência dos conjuntos estão sempre contidos no intervalo contínuo $[0, 1]$ onde 0 denota nenhuma pertinência e 1 denota total pertinência.

Imaginando o exemplo citado do copo com água, um conjunto de pertinência pode representar o quão cheio de água um copo está, pensando nas entradas como a quantidade de água no copo e a saída a representação de quão cheio o copo está. Do mesmo modo, outro conjunto significando quão vazio o copo está pode ser formado. Note que desse modo certa quantidade de água indicará tanto que o copo está vazio quanto cheio, contudo com valores de pertinência diferentes em cada conjunto, assemelhando-se a ideia do pensamento humano (“meio cheio”, “pouco vazio”, etc), se distanciado da lógica booleana que indicaria apenas se o copo está cheio ou vazio.

Estes conceitos podem ser utilizados para criar sistemas computacionais chamados de sistemas *fuzzy*. Geralmente todo sistema *fuzzy* é estruturado como apresentado na Figura 2:

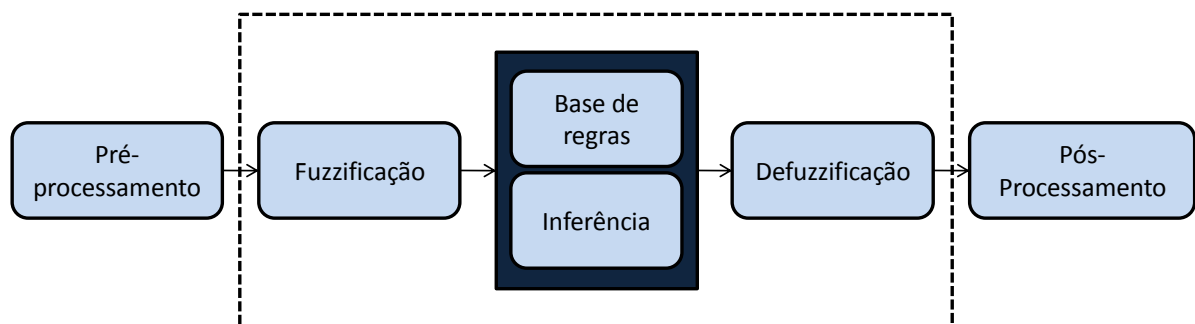


Figura 2. Blocos de um sistema *fuzzy* [Jantenz, 98].

O pré-processamento é uma etapa que consiste em ajustar os dados de entrada para que sejam interpretáveis pelo sistema *fuzzy*. Desse modo, essa etapa trata de normalizar ou transformar os valores segundo alguma escala, tratar algum tipo de ruído, ou até, no caso de sistemas que utilizam sensores, capturar as informações do mundo real (temperatura, iluminação, etc.) e transformar essas medidas em algo interpretável pelos computadores.

A fuzzificação é de fato a primeira etapa realmente envolvendo os conceitos *fuzzy*. Essa etapa mapeia os valores *crisp* de entrada para conjuntos *fuzzy*, atribuindo os respectivos valores de pertinência através das funções de pertinência definidas. Em sistemas *fuzzy*, geralmente, a fuzzificação é configurada através de gráficos que mapeiam os valores *crisp* para termos vagos como ‘pouco’, ‘muito’, ‘bom’, ‘médio’, etc. Com a utilização de termos, os sistemas *fuzzy* facilitam as configurações necessárias a serem feitas bem como regras que devem ser seguidas, pois assim os valores matemáticos são escondidos do usuário e o mesmo passa a lidar com termos que são utilizados normalmente no mundo real. A Figura 3 mostra um gráfico contendo termos vagos e os valores de pertinência:

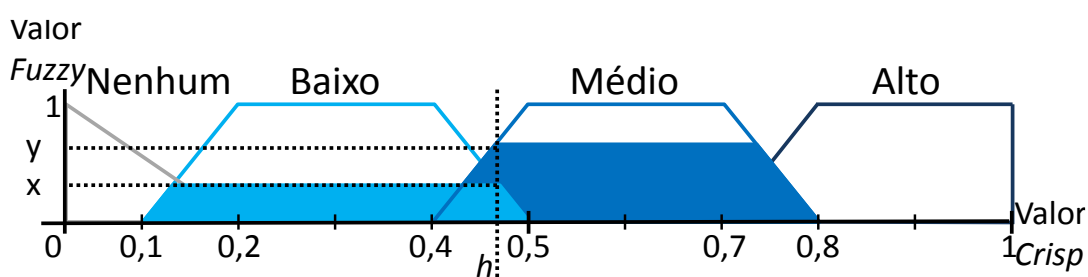


Figura 3. Fuzzificação de valores *crisp* em termos vagos.

Note que o eixo horizontal contém os valores *crisp* do mundo real e o eixo vertical indica sua pertinência nos conjuntos associados. Os valores do eixo horizontal são dependentes de cada problema associado, já os do eixo vertical sempre estão contidos no intervalo contínuo $[0, 1]$. Cada figura geométrica representa um conjunto *fuzzy*, assim um valor *crisp* pode ser mapeado para mais de um conjunto. Cada conjunto é representado por um termo vago. O valor de pertinência é calculado segundo as funções do gráfico no qual o ponto *crisp* o corta verticalmente. No exemplo ilustrado, o valor *crisp* h é fuzzificado para os termos (conjuntos) Baixo e Médio com pertinências x e y respectivamente.

A partir desses conjuntos é gerada uma Base de Regras do tipo IF-THEN que farão a inferência com base nos valores definidos, onde nos consequentes e antecedentes são utilizados os termos vagos definidos. Vários termos podem ser inclusive utilizados, sendo associados pelos operadores AND, OR, XOR, entre outros. Tomando como exemplo o contexto de seleção de RH regras podem ser feitas para definir o modo de seleção, como por exemplo:

IF competência IS alta OR competência IS media THEN seleção IS alta

IF competência IS baixa OR competência IS nenhuma THEN seleção IS baixa

Note que as regras indicam o nível de aceitação para selecionar recursos com as competências indicadas. A etapa de inferência consiste em analisar as regras e mapear os consequentes das mesmas para uma representação *fuzzy* semelhante à apresentada na Figura 3.

Assim que as regras são avaliadas e os resultados são gerados, os mesmos ainda estão representados segundo os conjuntos *fuzzy* e níveis de pertinência nos mesmos. Contudo para vários sistemas, é necessário que os valores voltem a ser representados segundo valores *crisps* para que possam ser utilizados. Dessa forma, a etapa de defuzzificação objetiva transformar os resultados de volta em valores *crisps*.

Vários métodos de defuzzificação existem. Dentre estes métodos, sete descritos em [Hellendoorn, 1993] são os mais conhecidos e utilizados. Entre esses métodos se encontram:

- *Max membership principle*
- *Centroid Method*
- *Weighted average method*
- *Mean Max membership*
- *Center of sums*
- *Center of largest area*
- *First (last) of maxima*

Alguns desses métodos sugerem achar o ponto onde a pertinência seja mais alta e atribuí-lo com o valor *crisp* correspondente (*Max membership principle*). Outros métodos buscam os valores de pertinência mais altos que aparecem primeiro ou por último (*First (last) of maxima*). Outros consistem em calcular as áreas de pertinência encontradas derivando um valor que se concentre relativamente no meio da figura gerada pelas áreas de pertinência (*Centroid Method*, *Weighted average method*, *Mean Max membership*, *Center of sums*, *Center of largest area*). Estes últimos são geralmente os mais utilizados. A escolha de qual método utilizar depende do contexto no qual o problema está inserido.

Por fim, o pós-processamento visa à formatação dos resultados para que possam ser utilizados. Por exemplo, mudança de escalas nos valores, normalização ou até novos cálculos necessários antes do valor ser utilizado.

2.3 Considerações finais

Neste capítulo foram apresentados os conceitos fundamentais sobre gerenciamento e seleção de recursos humanos bem como os conceitos necessários para o entendimento da utilização de

lógica *fuzzy*. Quanto ao gerenciamento de RH pretendeu-se neste capítulo apresentar como é realizado geralmente a identificação de competências dos profissionais e a seleção dos mesmos, bem como as dificuldades e problemas inseridos neste contexto, assim demonstrando a motivação para a criação da abordagem proposta. Quanto à lógica *fuzzy*, pretendeu-se apresentar o seu funcionamento em sistemas computacionais para que o leitor possa entender sua utilização no trabalho proposto.

Capítulo 3

***Framework* de Recomendação de Alocação de Equipes**

A abordagem de seleção de equipes tecnicamente qualificadas descrita neste trabalho é parte integrante de um *framework* que tem como objetivo principal dar suporte a um gerente de projetos para executar o processo de alocação de equipes distribuídas às tarefas de implementação de uma LPS. A proposta original do *framework* é apresentada em [Pereira, et al., 2010]. Para atingir seu objetivo, nas diferentes fases do *framework*, são geradas várias sugestões que apoiam as decisões do gerente de projetos no que diz respeito à melhor forma de dividir as tarefas a serem realizadas, bem como a escolha daqueles que irão realizá-las. No contexto do *framework*, toda tarefa diz respeito apenas à tarefa de implementação de *software*. Demais tarefas comumente envolvidas em projetos de *software* tais como levantamento de requisitos, elaboração de arquitetura, testes de *software*, entre outras, estão fora do escopo do trabalho.

Como o cenário ao qual o *framework* pretende dar suporte possui o envolvimento direto de equipes distribuídas pelo mundo, a principal preocupação na divisão de tarefas encontra-se nas dificuldades geradas pela comunicação entre estas equipes. O fator comunicação é problema inerente a qualquer abordagem de DDS [Carmel, 1999] [Gumm, 2006] [Herbsleb, 2007]. Vários trabalhos tendem a solucionar os problemas decorrentes da comunicação em DDS de uma forma reativa, ou seja, partindo do pressuposto que um conjunto de equipes distribuídas pré-existentes devem se comunicar entre si. Estes trabalhos propõem abordagens que tentam diminuir os problemas de comunicação através do aumento da percepção entre as equipes [Ramasubbu, et al., 2005] [Prikladnicki, et al., 2008] [Chaves, et al., 2008].

A proposta do *framework* aqui descrito é de minimizar as necessidades de comunicação de uma forma proativa. Assim, ao invés de resolver os problemas decorrentes da comunicação após a ocorrência dos mesmos, tenta-se evitá-los ou diminuí-los a partir da análise de características, tanto do software quanto das equipes distribuídas, que venham a

influenciar na comunicação entre as equipes. Contudo, vale ressaltar que sempre existirá algum nível necessário de comunicação, visto que as partes de *software* implementadas por cada equipe precisam ser posteriormente integradas para que a plataforma ou núcleo da LPS seja construído (no caso da engenharia de domínio) ou para que alguma aplicação seja instanciada (no caso da engenharia de aplicação).

As entidades que compõem o *framework*, suas fases, bem como entradas e saídas das mesmas serão descritas nas subseções seguintes.

3.1 Composição do *Framework*

Considerando um conjunto $E = \{e_1, e_2, e_3, \dots, e_x\}$ de equipes que podem vir a fazer parte do projeto de *software*, o *framework* é dividido em três tipos de recomendações que devem ser geradas ao longo de sua utilização e que identificam:

- i) um conjunto de módulos de software $M = \{m_1, m_2, m_3, \dots, m_n\}$ que possuam pouca dependência entre si e, portanto, possam ser desenvolvidos separadamente por equipes diferentes;
- ii) um mapeamento $EM = \{(m, se) \mid m \in M \wedge se \subseteq E\}$ que identifica, para cada módulo de *software* m , um subconjunto de equipes se composto de equipes que possuem qualificações técnicas adequadas para a implementação deste módulo de *software*;
- iii) um conjunto de alocações $A = \{(m, e) \mid m \in M \wedge e \in E \wedge (m, se) \in EM \wedge e \in se\}$, onde cada alocação provê suporte à tomada de decisão do gerente de projetos em como delegar cada módulo de *software* m para uma equipe específica e .

É importante salientar que as recomendações são apenas sugestões e não soluções mandatórias. O intuito é proporcionar ao gerente de projetos recomendações com base em análises fundamentadas nas características das entidades referentes a cada fase de recomendação, provendo assim soluções que tenham em média uma qualidade melhor do que soluções feitas apenas com base na experiência do gerente de projetos. Contudo, a experiência do gerente, assim como informações que vão além das características levantadas pelo *framework*, deve ser levada em conta no sentido de que a partir das recomendações propostas o gerente possa fazer os devidos ajustes para que as mesmas possam atender melhor às necessidades específicas de cada projeto.

Com o objetivo de gerar os três tipos de recomendações citadas anteriormente, o *framework* é dividido em três fases como ilustra a Figura 4. Além das fases de recomendações, o *framework* provê etapas para caracterização de requisitos técnicos de cada módulo de *software* e dos aspectos técnicos e não-técnicos de cada equipe. O *framework* conta também com uma última fase que tem como objetivo avaliar o desempenho das equipes no tocante à tarefa de implementação realizada por elas. Essa informação será útil para novas recomendações em projetos futuros. Além das fases, podem ser vistos na figura os artefatos que servem de entrada para cada uma das fases, bem como o artefato final *Alocação* que identifica o conjunto $A = \{(m, e) \mid m \in M \wedge e \in E \wedge (m, se) \in EM \wedge e \in se\}$ de alocações gerado após o término da execução do *framework*.

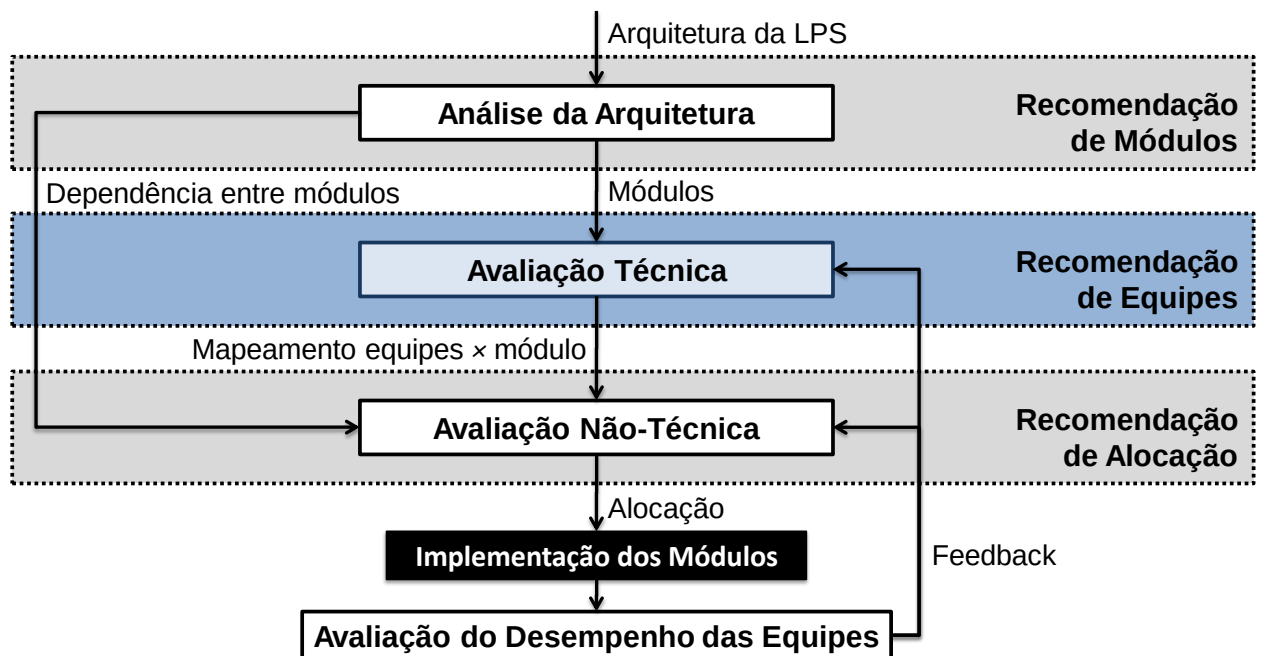


Figura 4. Visão geral das fases do *framework*.

3.2 Análise da Arquitetura

Apoiando-se na ideia de que a modularidade na estrutura de um *software* resulta na modularidade das subsequentes tarefas de projeto [Herbsleb, 2007], a fase de *Análise da arquitetura* busca analisar uma arquitetura de uma LPS e identificar dependências entre componentes. O intuito é identificar módulos de *software* que sejam os mais independentes possíveis entre si. Assim, são geradas as recomendações de módulos de *software* representado pelo conjunto de módulos $M = \{m_1, m_2, m_3, \dots, m_n\}$, onde cada elemento do conjunto possui o menor nível de dependência possível com os demais elementos. Partindo da premissa de que

quanto mais independentes forem os módulos de *software*, menor é a necessidade de comunicação entre as equipes que os implementarão, dois objetivos principais são traçados nessa fase: *i)* identificação dos módulos de *software* que devem ser implementados e integrados pelas diferentes equipes; *ii)* medição do grau de dependência entre os módulos de *software*.

Para que se possa dar início à fase de análise e identificação dos módulos é necessário que se passe como entrada para esta fase a *arquitetura baseada em componentes* do projeto de LPS. Idealmente essa arquitetura deve representar os componentes, as interações entre os componentes baseados nas interfaces requeridas e providas, nas operações requeridas e providas e no diagrama de *features*. Essa arquitetura pode ser uma arquitetura do domínio da LPS ou de alguma aplicação a ser instanciada.

Como o *framework* se baseia no princípio de que dependências entre módulos têm um impacto direto no nível de comunicação requerido pelas equipes, cada módulo de *software* é definido como um conjunto de componentes onde as dependências internas (entre os componentes do mesmo módulo) são maiores que as dependências externas (entre componentes de diferentes módulos).

Para a formação dos módulos de *software* (agrupamento), a fase de *Análise da Arquitetura* provê métricas arquiteturais para guiar a avaliação de dependência entre os componentes, bem como um algoritmo baseado na metaheurística *Simulated Annealing* para encontrar uma solução de agrupamento satisfatória [Pereira, 2011].

3.3 Avaliação técnica

Neste ponto se encontra o foco deste trabalho. Com os módulos de *software* já definidos, os engenheiros de *software* podem prover informações acerca dos requisitos técnicos de cada módulo identificado na *arquitetura baseada em módulos*. Essa descrição tem o intuito de prover informações necessárias para a posterior análise técnica das equipes distribuídas, a fim de verificar qual delas possui os atributos técnicos mínimos necessários para que possam implementar os módulos de *software*. São levantadas, portanto, todas as tecnologias (linguagens de programação, ferramentas, processos, plataformas, etc.) e conhecimento de domínios necessário para a implementação de cada módulo de *software* identificado [Santos et. al. 2011].

Com os requisitos de implementação identificados, é gerado um questionário destinado às equipes de desenvolvimento cujo objetivo é identificar o conhecimento de cada equipe acerca das tecnologias que devem ser utilizadas para implementação dos módulos de *software*. Como cada equipe pode se envolver em diferentes projetos ao longo do tempo, as informações são atualizadas toda vez que uma equipe se engaja num novo projeto. Assim, todas as informações contextuais necessárias para avaliação de uma equipe sempre estarão atualizadas.

O *framework* considera as seguintes informações para a caracterização do conhecimento técnico das equipes: *anos de trabalhos*, *número de projetos desenvolvidos*, *títulos* (certificações e especializações) e *reputação técnica* da equipe.

Assim que todas as equipes e do conjunto $E = \{e_1, e_2, e_3, \dots, e_x\}$, de equipes que podem vir a fazer parte do projeto de *software*, tiverem sido tecnicamente descritas, elas passam a estar aptas ao processo de avaliação da qualificação técnica que elas possuem para a implementação dos módulos de *software* descritos no projeto. Para essa avaliação é utilizado um algoritmo de lógica *fuzzy* que a partir dos atributos levantados calcula o conhecimento que cada equipe possui em cada tecnologia e utilizando uma política de seleção configurada pelo gerente de projetos, avalia cada equipe e retorna o mapeamento $EM = \{ (m, se) \mid m \in M \wedge se \subseteq E \}$ representado como uma lista ordenada das equipes mais adequadas tecnicamente para a implementação de cada módulo de *software* identificado.

3.4 Avaliação não técnica

A fase de *Avaliação não-técnica* consiste em duas etapas: obtenção de informações não-técnicas das equipes e, como sugere o nome da fase, avaliação não-técnica das equipes. Primeiramente são levantadas das equipes informações que possam vir a causar impacto nos aspectos de comunicação entre equipes. Como essas informações independem dos projetos a serem desenvolvidos, os atributos referentes aos aspectos não-técnicos podem ser levantados no início do ciclo do *framework* ou paralelamente a qualquer etapa anterior a etapa de *Avaliação não-técnica*, onde estas informações serão efetivamente avaliadas.

Os atributos não-técnicos podem ser classificados em quatro diferentes tipos sendo eles: *geográficos*, *temporais*, *culturais* e *de reputação*. Os atributos geográficos representam a localização da equipe no mundo, relacionando-as com país, estado, cidade ou até continente onde se encontram. Os atributos temporais estão relacionados com horário de trabalho das equipes, tanto ao fuso horário de onde se encontram como aos horários que as equipes

trabalham durante o dia e em quais dias das semanas elas trabalham. Com essas informações, é possível criar janelas de tempo que relacionam duas ou mais equipes e verificar a interseção de tempo em que elas trabalham simultaneamente (na visão temporal). Essa análise pode apontar o tipo de comunicação (síncrona ou assíncrona) adequada entre as equipes. O aspecto cultural tem o intuito de caracterizar as línguas que as equipes podem utilizar para se comunicar em termos de quão bem elas falam, escutam, escrevem e leem nessas línguas. Por fim, o atributo de reputação serve para avaliar quão bem foi o relacionamento de uma equipe com as demais equipes durante a implementação dos módulos. Desse modo as recomendações feitas podem ser avaliadas e no futuro recomendações melhores podem ser sugeridas.

Depois de levantadas todas as informações não-técnicas, a fase de *Avaliação não técnica*, utiliza os artefatos *dependência entre módulos* e *mapeamento equipe × módulos* para gerar as melhores combinações de equipes visando a melhor comunicação entre equipes que irão implementar módulos que possuam dependências entre si.

Devido ao esforço computacional requerido para avaliar todas as possíveis recomendações, a etapa de *alocação de equipes* utiliza-se de métodos heurísticos com base em algoritmos genéticos. Com o uso desse tipo de algoritmo, não é necessário analisar todas as possíveis soluções para o problema [Linden, 2008], o que no cenário do *framework* demandaria um custo computacional considerável devido ao grande número de possibilidades de alocações que podem ser geradas. Assim o uso de algoritmo genético cria um número limitado de alocações, as avalia e verifica se elas são boas o suficiente e aceitáveis. Se não forem, novas alocações são geradas a fim de melhorar as anteriores. Esse processo é feito até que as recomendações geradas atinjam um nível aceitável de qualidade. Ao final do processo são geradas as melhores soluções de alocação encontradas permitindo ao gerente selecionar a melhor pra o contexto de seu problema [Ribeiro, 2013].

3.5 Avaliação do desempenho das equipes

Após a implementação de todos os módulos m do conjunto $M = \{m_1, m_2, m_3, ..., m_n\}$, duas avaliações de desempenho das equipes distribuídas tornam-se necessárias a fim de atualizar e refinar os modelos de descrição técnica e não-técnica das equipes para que, em próximos projetos, recomendações cada vez melhores possam ser geradas.

Na visão técnica, o gerente de projeto irá avaliar as equipes de acordo com o desempenho que as mesmas obtiveram na implementação dos módulos de *software* a elas atribuídos. São observados fatores como tempo gasto, qualidade do *software* e número de

defeitos. Essa avaliação se faz necessária, pois, como a avaliação inicial do conhecimento das equipes é feita com base em informações fornecidas pelas próprias equipes, a avaliação do desempenho pelo gerente pode gerar outra perspectiva do desempenho real das equipes, fazendo com que avaliações futuras envolvendo essas equipes levem em conta o desempenho prático das mesmas.

Do ponto de vista não técnico, as equipes são avaliadas entre si de acordo com a interação que as equipes tiveram durante o processo de desenvolvimento dos módulos de *software* identificados. Portanto, essas avaliações são feitas pelas próprias equipes onde as mesmas pontuam as equipes com as quais interagiram, indicando se a comunicação foi boa ou não. Desse modo, o *framework* pode, em alocações futuras, verificar quais equipes possuem boas interações.

3.6 Considerações Finais

Neste capítulo foi apresentada a visão geral do *framework* ao qual a abordagem proposta neste trabalho se integra. Foram vistas cada uma de suas fases, bem como os respectivos artefatos de entrada e saída de cada fase. Vale ressaltar que o *framework* não impõe soluções ótimas para qualquer projeto no qual o mesmo seja utilizado. O intuito é proporcionar ao gerente de projetos uma metodologia que, baseada na análise de características que influenciam a comunicação entre equipes, unido com a análise de adequabilidade técnica dessas equipes para a implementação dos módulos de *software* identificados, possam ser geradas soluções que sejam em média melhores que uma escolha aleatória ou baseada apenas na experiência dos gerentes de projetos.

O *framework* gera diferentes tipos de recomendações, onde em cada uma delas o gerente de projetos pode refinar a seu critério as soluções recomendadas, caso as mesmas não sejam totalmente adequadas para seu contexto. Desse modo, o *framework* pode se adequar a grande maioria dos projetos, tornando-o configurável para atender as necessidades específicas de cada projeto.

Capítulo 4

Abordagem de seleção técnica de equipes

Este capítulo apresenta a abordagem de recomendação de seleção de equipes qualificadas tecnicamente para a implementação de módulos de *software*. O objetivo final da abordagem proposta é apoiar o gerente de projetos a selecionar equipes distribuídas mais adequadas dentre um conjunto de equipes identificadas como tecnicamente qualificadas para implementar os módulos de *software* do projeto de *software*, inclusive de uma LPS. Com o objetivo de gerar tais recomendações, são necessárias quatro etapas principais que podem ser vistas na Figura 5. Essa figura representa a etapa *Avaliação técnica* vista na Figura 4 do Capítulo 3.

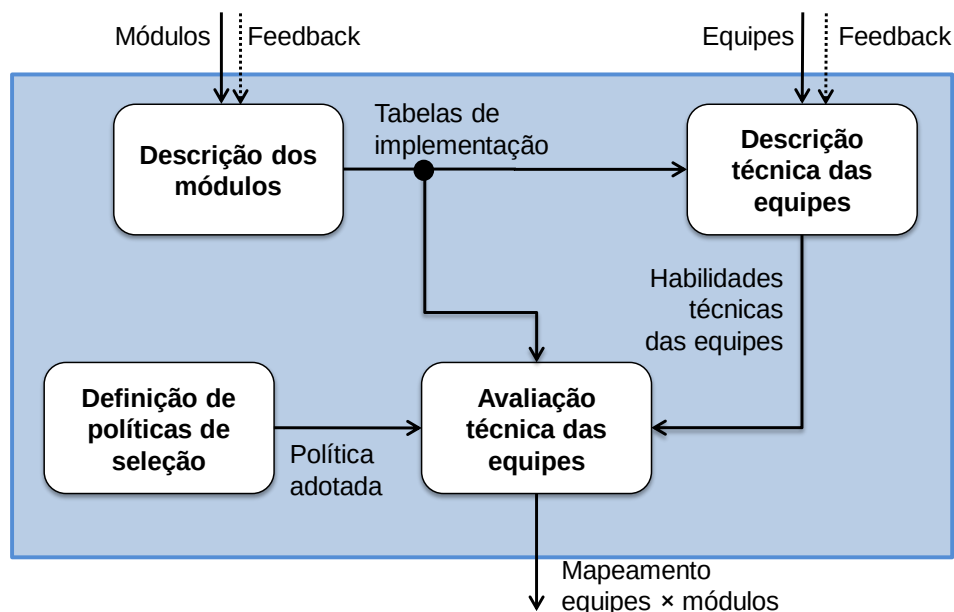


Figura 5. Etapas referentes à seleção técnica de equipes.

A etapa de *Descrição dos Módulos* é o primeiro passo desta abordagem. Ela recebe como entrada um conjunto de módulos de *software* a ser desenvolvido no projeto de *software*. Estes módulos são então descritos segundo os aspectos das tecnologias (OS, plataformas, ferramentas, linguagens de programação, processos, etc.) necessárias para implementá-los, bem como os domínios de aplicação nos quais estes módulos estão inseridos. Com estes

dados, um conjunto de artefatos de descrição técnica de módulos, sendo um por módulo, é criado como saída para as próximas etapas. A partir do conhecimento de todas as tecnologias necessárias para implementar todos os módulos do *software*, o gerente de projetos pode então reunir informações das equipes de desenvolvedores sobre os seus conhecimentos acerca das tecnologias identificadas. Isso é feito na etapa de *Descrição técnica das equipes*. Esta etapa recebe como entrada a descrição técnica de cada módulo, os atributos técnicos das equipes, necessários para medir o seu nível de conhecimento em cada tecnologia identificada na descrição técnica dos módulos e informações sobre o desempenho na implementação de módulos anteriores que a equipe tenha desenvolvido utilizando esta abordagem de seleção técnica de equipes. Após todos os dados necessários para inferir o nível de conhecimento das equipes acerca das tecnologias serem coletados e processados, esta etapa gera a descrição técnica das equipes como a saída para a próxima etapa.

Neste ponto, os módulos de *software* e as tecnologias necessárias para implementá-los, bem como as equipes e os seus níveis de conhecimento sobre estas tecnologias, foram todos coletados, reunidos e descritos nas respectivas etapas já mencionadas. Então a avaliação das equipes pode ser realizada tão logo que o gerente de projetos defina um conjunto de regras que irão conduzir a forma como a seleção das equipes deve ser feita. Estas regras são definidas na etapa de *Definição de políticas de seleção* e podem ser definidas de forma a dar preferência para as equipes mais experientes, para dar preferência às equipes cujo nível de conhecimento seja o mais próximo do conhecimento exigido pelas tecnologias identificadas nos módulos, dar preferência àquelas equipes que possuem o mais baixo nível de conhecimento, caso o gerente deseje prover treinamento para as equipes, ou qualquer outra política que possa ser concebida para melhor atender às necessidades específicas de cada projeto. O gerente de projetos pode predefinir um conjunto de políticas e escolher uma para cada processo de avaliação das equipes.

Com todas as informações detalhadas e uma política escolhida, a etapa de *Avaliação técnicas das equipes* pode prosseguir com a avaliação das equipes e módulos de *software*. Esta etapa recebe como entrada as saídas das demais etapas da abordagem. O processo de avaliação cruza todas as equipes com todos os módulos e, seguindo a política adotada, determina quais equipes são as mais adequadas para o desenvolvimento de cada módulo de *software* identificado, gerando como saída o artefato final *Mapeamento equipes × módulos*, que para cada módulo apresenta uma lista ordenada com as equipes mais adequadas

tecnicamente para implementá-lo, inclusive indicando o nível de adequação destas equipes (o nível de adequação é uma métrica variando entre 0 e 1).

Ao invés de selecionar as equipes contando apenas com o conhecimento tácito, percepção subjetiva e instinto do gerente do projeto, como geralmente é feito [Acuña 2003], o *Mapeamento equipes × módulos* gerado pela abordagem proposta permite ao gerente de projetos poder escolher com mais segurança e propriedade as equipes mais adequadas ao desenvolvimento de cada módulo de *software* do projeto de *software* de acordo com uma política definida pelo próprio gerente, visando atender da melhor maneira as necessidades do projeto.

As seções a seguir explicam com mais detalhes cada etapa da abordagem proposta da Seleção Técnica de Equipes.

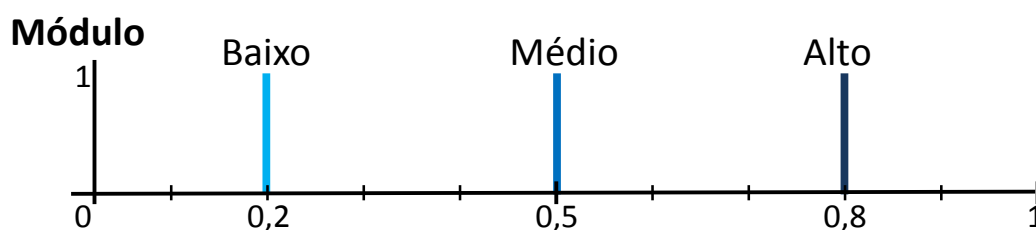
4.1 Descrição dos módulos

Como explicado anteriormente, para tornar possível a avaliação das equipes e módulos de *software*, é necessário obter informações, tanto dos módulos quanto das equipes, necessárias para proceder com a avaliação. Do ponto de vista dos módulos de *software*, na etapa de *Descrição dos Módulos*, é necessário obter uma lista (sendo representada como uma tabela) das tecnologias (SO, plataformas, ferramentas, linguagens de programação, processos, etc.) e domínios de conhecimento envolvidos na construção do módulo cujas equipes devem saber para conseguirem desenvolvê-lo. Como o principal objetivo desta fase é levantar não somente as tecnologias requeridas para implementação dos módulos, mas o nível de complexidade exigido em cada uma delas, cada coluna da tabela representa um nível diferente de complexidade exigida, contendo nas células as tecnologias e conhecimentos de domínio necessários correspondentes a essa complexidade. Este artefato é chamado de *Tabela de implementação*, pois contém os requisitos técnicos que uma equipe deve possuir para implementar o módulo de *software* ao qual a tabela se refere. Um exemplo dessa tabela pode ser visto na Tabela 1.

Tabela 1. Tabela de implementação.

Baixo	Médio	Alto
Eclipse	Hibernate	Java SE
XP	JSF	Java EE
	JSP	

O número de níveis e os termos usados para descrever o nível de complexidade tecnológica podem ser redefinidos pelo gerente de projeto como ele julgar suficiente. Perceba que são utilizados termos difusos para a caracterização dos níveis. Como esperado, esses termos representam conjuntos *fuzzy* que também podem ser refinados pelo gerente de projetos a fim de adaptá-los ao projeto a ser desenvolvido. A configuração padrão desses conjuntos é dada por *singletons* como apresentado na Figura 6.

**Figura 6. Conjuntos *Fuzzy* de representação dos módulos de *software*.**

Como se entende que o gerente de *software*, auxiliado pelo o arquiteto de *software*, define a tabela de implementação, o nível de complexidade de cada tecnologia já é conhecido pelo arquiteto, pois foi ele que modelou a arquitetura do projeto. Dessa forma, esse nível de complexidade não é calculado pela abordagem proposta e por essa razão, preferiu-se utilizar *singletons*, pois sempre uma tecnologia estará associada a apenas um termo ou conjunto por vez e sempre pertencerá inteiramente a esse conjunto. Isso torna desnecessário o uso de conjuntos no formato triângulo ou trapézio, por exemplo, já que o foco se torna apenas no termo difuso em si. Os valores *crisp* 0.2 (Baixo), 0.5 (Médio) e 0.8 (Alto) foram utilizados, pois são valores que iniciam os respectivos conjuntos utilizados para o resultado de adequabilidade. Contudo, por se tratar de *singletons*, não faria diferença caso fosse escolhido qualquer outro valor dentro dos conjuntos de pertinência. Caso o gerente de projeto decida aumentar ou diminuir o número de termos para representar a complexidade exigida das tecnologias dos módulos, recomenda-se que dividia o valor *crisp* máximo 1 pelo número de

termos a ser utilizado, para assim obter uma divisão com distâncias iguais entre os valores, apenas por padronização, pois como já mencionado, os valores dos termos *singletons* não influenciam na aplicação, apenas a quantidade de termos a ser utilizado.

Para cada módulo identificado no projeto, uma tabela de implementação deve ser construída. Como os módulos do *software* podem ser originalmente de um projeto de LPS, é esperado que alguns módulos desenvolvidos anteriormente possam ser reutilizados para criar novos módulos em futuras iterações da vida útil do projeto de LPS. Nesses casos, não só as equipes têm de conhecer as tecnologias necessárias para implementar os novos requisitos do módulo, mas também precisam ter conhecimento acerca das tecnologias utilizadas anteriormente na construção do módulo original, a fim de integrar os novos requisitos na construção do novo módulo. É necessário lembrar que um módulo de *software* pode ser reutilizado como base de um novo módulo e este novo módulo por sua vez também é passível de ser reutilizado para a construção de outro módulo e assim consecutivamente. Por esse motivo, fica claro que se torna necessário sempre armazenar um histórico de quais tecnologias foram utilizadas na construção dos módulos de *software*. Portanto, para contar tanto com as novas tecnologias que devem ser utilizadas para implementar o módulo e as tecnologias anteriores que devem ser conhecidas pelas equipes com o propósito de integração, dois tipos de tabelas devem existir: *i)* tabela de implementação, contemplando as novas tecnologias para implementar novos requisitos e tecnologias necessárias para integração desses novos requisitos ao módulo existente e *ii)* tabelas históricos, representando todas as tecnologias que alguma vez foram utilizadas para implementar o módulo de *software* durante a sua existência. Essa última consiste na mesma tabela de implementação que apenas muda seu *status* para servir de histórico. Assim, estes dois tipos de tabelas possuem a mesma representação. Entretanto, a tabela de implementação é a tabela utilizada efetivamente durante o processo de avaliação das equipes, enquanto a tabela histórico serve como uma referência das tecnologias utilizadas no módulo em algum momento. Assim, com o uso de tabelas histórico é possível ao gerente capturar as tecnologias necessárias para integrar o novo módulo com as partes já implementadas anteriormente. Sem contar o fato da tabela poder servir como referência para alguma eventual manutenção, já que possui as tecnologias utilizadas na construção do módulo.

É importante notar que, depois de um módulo de *software* ser implementado é possível que algumas tecnologias previstas na tabela de implementação não tenham sido utilizadas de fato na implantação do módulo de *software*. Da mesma forma, é possível que algumas

tecnologias que não foram planejadas para o desenvolvimento do módulo possam ter sido utilizadas durante seu desenvolvimento. Portanto, a fim de preservar a integridade das tabelas histórico de tecnologias utilizadas, antes de tornar a tabela de implementação numa tabela histórico, o gerente de projetos deve realizar junto com as equipes que implementaram cada módulos de *software* e também com o arquiteto de *software*, uma revisão da tabela, adicionando tecnologias que foram utilizadas mas não planejadas e/ou removendo tecnologias planejadas mas que não foram de fato utilizadas. Apesar de fugir do escopo do trabalho por não ter uso prático na abordagem é interessante comentar que o gerente de projetos pode associar às *tabelas histórico* uma documentação contendo os motivos de algumas tecnologias não terem sido utilizadas ou, no caso contrário, terem sido adicionadas. Com essa informação o arquiteto de *software* pode obter um *feedback* e conseguir planejar mais efetivamente quais tecnologias utilizar na construção de projetos futuros. Isso indiretamente produzirá tabelas de implementação melhores no futuro.

Os artefatos gerados – conjunto de *tabelas de implementação* - constituem a saída da etapa de *Descrição de Módulos*, e serve como entrada para a etapa de *Descrição técnica das equipes* e a etapa de *Avaliação técnica das equipes*. Note que o *feedback* apresentado na Figura 5 como entrada da *Descrição dos módulos* consiste na *tabela histórico*.

4.2 Descrição técnica das equipes

Uma vez que as tabelas de implementação são passadas para a etapa de *Descrição técnica das equipes* é possível elaborar questionários para serem usados para coletar informações das equipes acerca das tecnologias e domínios identificados. Esses questionários, sendo um por módulo, são respondidos pelas equipes de desenvolvimento. Mais precisamente, por cada membro de cada equipe candidata a participar do projeto a ser desenvolvido. Pode-se argumentar que a elaboração dos questionários poderia ser feita independentemente da geração das tabelas de implementação, contudo, se esse fosse o caso, o responsável por desenvolver o questionário tentaria levantar o maior número de tecnologias possíveis, já que nesse ponto o mesmo não saberia que tecnologias seriam utilizadas na construção do *software*. Isso tornaria o questionário maçante e nada prático para ser respondido. Além disso, poderia ocorrer de tecnologias levantadas no questionário não serem de fato utilizadas na construção dos módulos ou que tecnologias utilizadas não tenham sido parte no questionário. Em ambos os casos, isso seria desperdício de tempo e recursos, pois, além do tempo de fabricação dos questionários, as equipes estariam respondendo questões desnecessárias. Pode-

se argumentar também que, sendo esse o caso, as equipes poderiam então apresentar todas as informações sobre elas numa espécie de currículo, sendo então desnecessária a utilização de questionários. Apesar de possível, essa abordagem é mais complicada, pois um modelo de currículo a ser utilizado necessitaria ser construído e as equipes então deveriam seguir esse modelo para a construção de seus currículos. Note que assim a responsabilidade de construção do currículo ficaria a cargo da própria equipe. Isso novamente tomaria tempo das mesmas, o que poderia acarretar também das equipes não seguirem direito o padrão estipulado de currículo.

Através dos dados coletados nos questionários, a experiência de uma equipe em cada tecnologia pode ser calculada, gerando um número entre zero e um, expressando a competência ou *habilidade técnica* estimada (conhecimento que uma equipe apresenta em alguma tecnologia). Um exemplo de questionário, que configura parte do questionário utilizado para os estudos de caso desta abordagem, pode ser visto na Figura 7. O questionário completo utilizado para o estudo de caso da abordagem proposta pode ser visto no Anexo A.

O questionário exemplo apresentado contempla informação acerca de duas tecnologias: linguagem de programação Java e na ferramenta Eclipse, uma IDE para Java. Para cada tecnologia identificada no questionário, até três dados devem ser coletados: *anos de experiência*, *número de projetos desenvolvidos* e *número de títulos* (títulos podem ser certificações ou especializações). Todos estes dados são relativos a cada tecnologia apresentada no questionário, sendo levantados para cada uma das tecnologias identificadas nas tabelas de implementação dos módulos de *software*. Os intervalos e números de níveis utilizados para representar as possíveis respostas podem ser definidos a critério do gerente de projetos. Note que também é possível definir uma pergunta sem definir qualquer nível para as respostas, permitindo que os membros das equipes preencham um valor estimado (no caso *anos de experiência*, *número de projetos desenvolvidos*) ou que estes escrevam os títulos que eles possuem. Neste trabalho não foi utilizado questionário com respostas abertas para os quesitos *anos de experiência*, *número de projetos desenvolvidos*, mas será apresentado como pode ser realizado o processo de cálculos dos valores com essa abordagem para o cálculo do valor de títulos apresentados. Quanto aos *títulos* é possível que algumas tecnologias não ofereçam certificações ou especializações, nesse caso a informação pode ser omitida.

Conhecimento: JAVA Conhecimento e familiaridade na utilização da linguagem Java
Há quantos anos você desenvolve projetos em JAVA? ☐ Nenhum ☐ 1-3 ☐ 3-5 ☐ 5-7 ☐ 7-9 ☐ 9 ou mais.
Quantos projetos você desenvolveu utilizando a linguagem JAVA? ☐ Nenhum ☐ 1-5 ☐ 5-10 ☐ 10-15 ☐ 15-20 ☐ 20-25 ☐ 25-30 ☐ 30 ou mais.
Quais dessas certificações você possui em JAVA? ☐ SCJA ☐ SCJP ☐ SCJD ☐ SCWCD ☐ SCBCD ☐ SCDWJC ☐ SCEA ☐ SMAD
Conhecimento: Eclipse Conhecimento e familiaridade na utilização da IDE Eclipse para JAVA
Há quantos anos você trabalha com o Eclipse? ☐ Nenhum ☐ 1-3 ☐ 3-5 ☐ 5 ou mais.
Quantos projetos você desenvolveu utilizando Eclipse? ☐ Nenhum ☐ 1-5 ☐ 5-10 ☐ 10-15 ☐ 15 ou mais.
...

Figura 7. Exemplo de questionário.

De acordo com Shanteau [Shanteau et. al., 2002], anos de experiência, títulos, certificados, etc. que alguém possui, são comumente utilizados para avaliar se esse alguém é um especialista em determinado domínio. Weiss [Weiss et. al., 2006] explica que um fator importante que determina se alguém é um especialista em um determinado domínio é o fator de *capacidade de discriminação*, descrito como a capacidade de discriminar diferenças sutis em contextos similares. Contudo essa capacidade de discriminação só pode ser vista reunindo

informações através da observação de eventos passados, uma vez que é tipicamente um dado empírico. No entanto, é possível inferir, até certo grau, a capacidade de discriminação de uma pessoa através do *número de projetos* que ela tenha participado. Uma vez que a capacidade de discriminação é adquirida através da participação em diversos projetos diferentes, mas com contextos similares, quanto mais alguém participa de projetos maior é a probabilidade de que essa pessoa presenciou domínios com pequenas diferenças, aumentando assim sua capacidade de discriminação. Desse modo, os atributos técnicos *anos de experiência*, *número de projetos desenvolvidos* e *número de títulos* foram definidos baseando-se nas necessidades específicas da abordagem proposta e na discussão proposta em alguns trabalhos tanto de recomendação de especialistas como de seleção ou alocação de recursos humanos [Shanteau et. al., 2002] [Weiss et. al., 2006] [Petry, 2007] [Barreto, 2005] [Silva, 2007] [Callegari, et al., 2009] [Kagdi, et al., 2008] [Duggan, et al., 2004] [Collofello, et al., 1998] [Ruskova, 2002].

Outro dado utilizado para calcular quanto uma equipe é adequada para desenvolver um módulo de *software* é o atributo *reputação*. Esse dado também foi derivado após a observação de alguns trabalhos supracitados. Esta reputação representa o quão bem uma equipe desempenhou o desenvolvimento de módulos de *software* durante projetos passados, seguindo a utilização da abordagem proposta. Com esta informação de reputação é possível avaliar a qualidade técnica das equipes na prática. O gerente de projeto é responsável por manter essa reputação, aumentando ou diminuindo seu valor de acordo com o desempenho prático das equipes ao final de cada projeto. Perceba então, que este dado não é coletado através dos questionários, mas já iniciado com um valor padrão e alterado pelo gerente depois de cada projeto desenvolvido pelas equipes, portanto servindo como a informação de *feedback* desta etapa. Diferente dos outros atributos de equipe, a reputação não é usada para calcular o valor de *habilidade técnica* que uma equipe possui em uma determinada tecnologia. A reputação é utilizada de fato na etapa de *Avaliação técnica das equipes* onde irá desempenhar um papel importante no cálculo de adequabilidade das equipes para implementação dos módulos.

Depois que todos os questionários são aplicados e os dados coletados, se torna possível calcular o valor de *habilidade técnica* que as equipes apresentam para cada uma das tecnologias identificadas nas tabelas de implementação. Como visto no questionário exemplo, os dados coletados para cada tecnologia (*anos de experiência*, *número de projetos desenvolvidos* e *número de títulos*) podem possuir diferentes representações: valores máximos, quantidade de níveis e variação entre níveis. Para manter todos os dados coletados nos questionários numa mesma magnitude, e também para gerar alguma expressividade, estes

três dados são mapeados para conjuntos difusos. Um conjunto difuso, como definido por Zadeh [Zadeh, 1965], é uma classe de elementos representados por uma função de pertinência que mapeia um valor *crisp* para um grau de pertinência estabelecido entre 0 e 1 neste conjunto. Apesar dos dados utilizados nos questionários serem distintos, a função de pertinência que mapeia os valores respondidos para um grau de pertinência no conjunto *fuzzy* de *grau de habilidade técnica* é obtida da mesma forma para os três tipos de atributos. Essa função é expressa pela Equação 1:

$$T = \left\{ \frac{0/l}{x_0} + \frac{1/l}{x_1} + \frac{2/l}{x_2} + \frac{3/l}{x_3} + \dots + \frac{l/l}{x_l} \right\} \quad (1)$$

Como apresentado no Capítulo 2, esta é a convenção da representação de notação de conjuntos difusos criada por Zadeh [Zadeh, 1965]. Assim, na Equação 1, o sinal + funciona apenas como separador das notações de pertinência. Já o símbolo l corresponde ao número de níveis menos um, utilizado para representar os dados do questionário desenvolvido para os membros da equipe. É utilizado o número de níveis menos um, pois a contagem de elementos se assemelha à contagem de elementos de um vetor, que começa a partir de 0 e não de 1, ou seja, se existem X elementos, a última posição é $X - 1$ já que a primeira posição é 0. O símbolo l também pode representar o valor mínimo que maximiza a função de pertinência (valor que mapeia um valor *crisp* para pertinência total 1), no caso de um modelo de respostas abertas ser escolhido ou no caso do quesito títulos. Por fim, cada x_i representa um dos níveis de resposta atribuído a uma pergunta do questionário, ou seja, representa o valor *crisp* final atribuído à resposta. Finalmente, cada i/l representa o valor de pertinência no conjunto de *grau de habilidade técnica* calculado para os valores *crisps* correspondentes, ou seja, existem tantos i/l quanto o número de opções que podem existir para cada pergunta, já que cada i/l representa uma possível alternativa de uma pergunta do questionário.

Para um melhor entendimento de como funciona esse mapeamento, um exemplo será mostrado. Suponha que o questionário da Figura 7 seja utilizado para coletar apenas os dados sobre a tecnologia Java de um conjunto de equipes. Perceba que o gerente de projetos atribuiu o número máximo de anos de utilização dessa tecnologia como 9, participações de projetos máximo como 30, e, apesar de não constar no questionário enviado para as equipes, o gerente define pesos para cada título definido (peso 0.1 para SCJA; peso 0.2 para SCJP, SCJD e SCWCD; peso 0.3 para SCBCD, SCDWJC e SCEA; e peso 0.4 para SMAD), bem como o peso que maximize a função de pertinência para este atributo, ou em outras palavras, um peso

que signifique que alguém que atinja esse valor, seja considerado como tendo um altíssimo grau de habilidade técnica (no caso do exemplo será utilizado valor um). Caso o gerente de projetos considere que cada título possui a mesma importância, o mesmo não precisa atribuir pesos distintos. Neste caso, basta apenas considerar o valor de cada título como a divisão do número de títulos pelo total de títulos e considerar o valor que maximiza a função de pertinência como o número total de títulos apresentados. Suponha que um membro a da equipe A preencha o questionário atribuindo o valor “() 5-7” para os anos de experiência, o valor “() 20-25” para o número de projetos desenvolvidos e assinale possuir os títulos SCJA e SCJP. Nesse sentido, utilizando-se a Equação 1 para mapear os anos de experiência assinalados para seu homólogo valor de pertinência *fuzzy* no conjunto de grau de habilidade técnica, é tomado o seguinte raciocínio. O termo “() 5-7” representa o nível três (o valor *crisp* “()Nenhum” é considerado como nível zero), portanto x_3 na Equação 1. O total de níveis é 6, portanto l é $(6-1) = 5$. Dessa maneira, a função de pertinência *fuzzy* mapeia o valor *crisp* para $3/5 = 0.6$. Para obter o número de projetos desenvolvidos a lógica segue o mesmo princípio. Como o valor *crisp* “() 20-25” representa o nível 5 (novamente o valor *crisp* “()Nenhum” é considerado como nível zero) e o valor de l é $(8-1) = 7$, a função de pertinência *fuzzy* mapeia o valor *crisp* para $5/7 \approx 0.71$. Finalmente, para os títulos, como o respondente assinalou possuir os títulos SCJA e SCJP, a função agora mapeia não a partir dos níveis, mas da soma de pesos e o valor l passa a corresponder ao peso máximo estipulado. Assim, como os pesos de SCJA e SCJP são respectivamente 0.1 e 0.2, e l é 1, o valor calculado de pertinência passa a ser $(0.1 + 0.2) / 1 = 0.3$. Note que, caso o valor ultrapasse o valor máximo 1 de pertinência de um conjunto *fuzzy*, o valor deverá ser considerado como sendo 1 para se adequar com os resultados das outras perguntas dos questionários.

Com o mapeamento de cada valor *crisp* para um valor de pertinência *fuzzy*, todos os valores passam a possuir mesma magnitude e assim podem ser comparados uns com outros, de modo que quanto mais perto do 0, menor é o grau de habilidade técnica, e quanto mais próximo de 1, maior é o grau de habilidade técnica. Mais importante que comparar os valores calculados, agora é possível agregá-los para assim obter um valor único que represente o grau de habilidade técnica de um membro da equipe em uma determinada tecnologia. Esse valor é calculado através de uma média ponderada como apresenta a Equação 2 que sempre resultará também num valor entre 0 e 1:

$$f(m, t) = \frac{Y_{m,t} \cdot w_y + P_{m,t} \cdot w_p + T_{m,t} \cdot w_t}{w_y + w_p + w_t} \quad (2)$$

Na Equação 2, $Y_{m,t}$, $P_{m,t}$ e $T_{m,t}$ representam respectivamente os anos de experiência, o número de projetos desenvolvidos e o número de títulos que um membro m de uma equipe e apresenta em uma determinada tecnologia t . Os pesos w_y , w_p e w_t são pesos de importância configurados pelo gerente de projetos para cada um destes dados a respeito da tecnologia t . Com a configuração dos pesos, o gerente de projetos é capaz de ajustar quais dados ele considera mais importantes no cálculo da *habilidade técnica*. Nas pesquisas de Shanteau [Shanteau et. al., 2002], reconhece-se que o número de títulos é o dado menos representativo entre os três. Shanteau também considera a capacidade de discriminação (aqui representada pelo número de projetos desenvolvidos), um fator crucial, tão importante quanto os anos de experiência. Então, intuitivamente, é recomendável que os pesos para o número de projetos desenvolvidos e anos de experiência tenham o mesmo valor e sejam maior que o peso dos títulos, embora essa não seja uma regra geral, já que estes valores podem variar para cada uma das tecnologias em diferentes contextos. Neste trabalho são admitidos como padrão para todas as tecnologias os valores $w_y = 4$, $w_p = 4$ e $w_t = 2$. Note que se uma tecnologia não possui títulos associados, o valor de w_t deve passar a ser 0.

Neste ponto, o valor do grau de *habilidade técnica* de um membro de equipe pode ser calculado para cada tecnologia t identificada nas tabelas de implementação. No entanto, o valor necessário para avaliar a adequabilidade de uma equipe para implementar os módulos de *software* é a *habilidade técnica* da equipe como um todo e não valores individuais de cada membro que dela faz parte. Desta forma, um novo cálculo considerando os valores de cada membro e o número de membros da equipe deve ser realizado para obter um valor representante único que mede a *habilidade técnica* da equipe para cada tecnologia. É necessário notar que conhecimentos podem ser complementares, como por exemplo, dois membros de equipe podem compartilhar conhecimento, criando assim mais conhecimento para a equipe. Supõe-se que quanto mais membros existem em uma equipe, mais pessoas existem para compartilhar conhecimento, aumentando assim o nível de conhecimento dessa equipe. Embora seja impossível estabelecer um valor preciso, há um certo número de pessoas em uma equipe que, passando este número, o compartilhamento de conhecimento torna-se inexpressivo, pois já existem muitas pessoas compartilhando o mesmo conhecimento. Assim, quanto mais membros uma equipe apresentar maior será a possibilidade dessa equipe

apresentar um maior conhecimento, contudo a medida que o número de membros aumenta, menor é essa contribuição para o valor final de conhecimento. A Equação 3 retrata esse cálculo:

$$f(e, t) = \left(\frac{\sum_{m=1}^n f(m, t)}{n} \right)^{0.5 \times \frac{n+1}{n}} \quad (3)$$

Na Equação 3 n representa o número total de membros da equipe e que apresentam algum conhecimento na tecnologia t , e m representa cada um desses membros. Embora, as equipes com mais membros tenham mais pessoas compartilhando conhecimento e, por consequência, têm uma chance maior de obter mais conhecimento do que equipes com poucos membros, esta fórmula não descarta equipes com poucos membros. É ainda possível para estas equipes terem um maior valor de *habilidade técnica* do que equipes com mais membros, isso dependerá do grau de *habilidade técnica* de cada membro da equipe. Isso se deve ao fator exponencial apresentado na fórmula que considera que cada membro adicional à equipe ($n+1$) traz consigo um pouco mais de conhecimento para equipe (n). Este fator exponencial foi inspirado em dilatadores *fuzzy*, e corresponde ao termo linguístico “mais ou menos” [Ross, 2004]. Com a utilização do dilatador é possível lidar satisfatoriamente com a imprecisão e o fato de não se ter um valor específico que defina que a partir desse valor as equipes passam a não ganhar um valor de conhecimento compartilhado significativo. Desenvolvendo a equação percebe-se que partindo de equipes que possuem apenas um membro, não há ninguém para agregar conhecimento à equipe, assim a fórmula não pode elevar em nada o conhecimento médio calculado. Perceba que nesse caso, onde $n = 1$, o fator exponencial resulta em 1, assim o valor considerado é apenas o valor de conhecimento puro do único membro da equipe. À medida que se começa a aumentar o número de membros de uma equipe, esse fator exponencial começa a aumentar, mas esse aumento é gradual e cada vez menor, justamente para contar com o fato de que quanto mais membros forem adicionados à equipe menor será o valor de conhecimento que se pode agregar, pois se começa a apresentar interseções de conhecimentos ou conhecimentos repetidos sobre o mesmo assunto e assim o conhecimento compartilhado passar a ser “mais ou menos” o mesmo.

É interessante notar que, embora o grau de *habilidade técnica* das equipes seja representado por um número real entre 0 e 1, essa informação não possui de fato a precisão de um número real, já que este foi calculado com base em níveis estimados de pertinência em um conjunto difuso. Para lidar apropriadamente com essa imprecisão e vagueza, a *habilidade*

técnica calculada é representada através de conjuntos de termos vagos, como *nenhum*, *médio*, *baixo* e *alto*, que passam a representar a *habilidade técnica* da equipe em uma determinada tecnologia. Diferente de usar apenas palavras simples, esses termos vagos são, na verdade, conjuntos difusos que escondem a complexidade numérica do usuário. A representação da capacidade técnica da equipe com os termos *fuzzy* é feita através do processo de *fuzzificação*, que consiste em encontrar a pertinência da *habilidade técnica* de uma equipe em cada um desses conjuntos de dados, usando uma função de pertinência. Desta forma, a imprecisão pode ser tratada sem prejudicar a avaliação das equipes e, em contrapartida, o valor torna-se muito mais compreensível para humanos. Os conjuntos *fuzzy* utilizados para representar a *habilidade técnica* das equipes podem ser vistos na Figura 8.

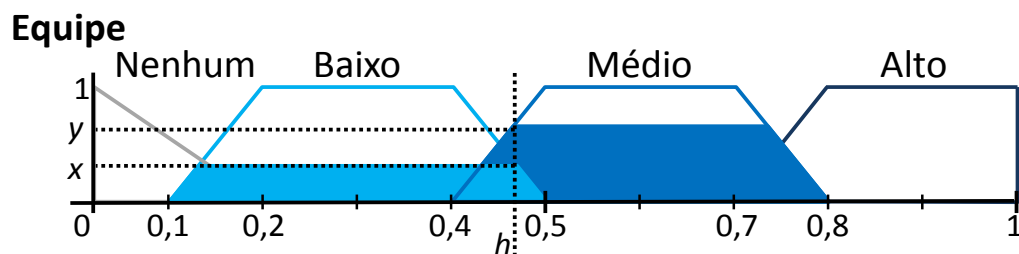


Figura 8. Conjuntos Fuzzy de representação das habilidades das equipes.

O arranjo detalhado na Figura 8, bem como demais configurações padrão apresentadas, foi definido empiricamente através da execução de vários ciclos do protótipo da abordagem de seleção proposta. Contudo, o tamanho dos trapézios e o número de termos podem ser alterados pelo gerente de projetos para atender as necessidades de cada projeto específico. Para descrever como o grau de habilidade técnica pode ser convertido (*fuzzificado*) para uma representação de termos vagos, um exemplo será descrito. Suponha que uma equipe tenha obtido um grau de *habilidade técnica* $h = 0,47$ em uma determinada tecnologia t . Este valor é mapeado para os conjuntos *fuzzy* de representação de habilidades das equipes e os termos que representarão este valor são obtidos pelo cálculo do ponto onde o valor $h = 0,47$ corta cada conjunto. Nesse exemplo, 0,47 situa-se em dois conjuntos: Baixo (valor de pertinência x) e Médio (valor de pertinência y). Como y é maior do que x é possível dizer que a habilidade técnica da equipe é mais média que baixa, ou que está próximo de média, embora os dois termos sejam utilizados simultaneamente para representar a capacidade técnica da equipe, assim como o grau de pertinência em cada um dos conjuntos.

Assim que todas as equipes passarem por todo o processo descrito nesta seção, será obtido todos os valores *fuzzificados* referentes a todas as habilidades técnicas de todas as

equipes em todas as tecnologias identificadas nas tabelas de implementação. Com isso é gerado o artefato *Habilidades técnicas das equipes* que será entrada da etapa de *Avaliação técnica das equipes*.

4.3 Definição de políticas de seleção

Com posse das *tabelas de implementação*, que contém as tecnologias identificadas com os respectivos níveis de complexidade necessários, e das *habilidades técnicas das equipes*, que associa cada equipe a cada uma das tecnologias identificadas nas *tabelas de implementação*, os requisitos necessários para se proceder com o processo de avaliação estão quase prontos. No entanto, antes da avaliação poder ser iniciada, a maneira de proceder com a avaliação deve ser previamente conhecida, em outras palavras, uma política de seleção deve ser seguida a fim de selecionar as equipes que irão implementar os módulos de *software* identificados. Dependendo das necessidades do projeto, diferentes políticas podem ser escolhidas, resultando em critérios diferentes sobre como selecionar as equipes.

Escolher as equipes que apresentam o maior conhecimento acerca das tecnologias necessárias para o desenvolvimento de um módulo de *software* pode ser uma boa opção quando o projeto sofre algum atraso, já que se supõe que as equipes com maior conhecimento terão menos problemas técnicos para desenvolver os módulos de *software*, portanto, não excedendo os prazos ou até mesmo diminuindo o tempo estipulado para desenvolvê-los. Entretanto, essa nem sempre é uma boa escolha. Selecionando sempre a equipe com maior conhecimento técnico para implementar qualquer módulo de *software* pode ser considerado como "desperdício de conhecimento" se o nível de complexidade técnica exigido pelo módulo for muito aquém do conhecimento apresentado pela equipe. Isso pode levar à insatisfação das equipes por não desenvolver atividades desafiadoras. Nesse caso, também pode elevar os custos dos projetos pois as equipes escolhidas apresentam maior conhecimento técnico, que geralmente são as que demandam o maior custo financeiro. Nesse caso, uma política de seleção que tente nivelar os níveis de conhecimento requeridos pelos módulos e apresentados pelas equipes seja mais adequada, já que as equipes serão selecionadas de forma a diminuir a lacuna de conhecimento entre os seus próprios conhecimentos e nível de complexidade esperados dos módulos de *software*. Este tipo de política pode reduzir os custos, mas em contrapartida o projeto pode não ser terminado tão rápido quanto seria se fosse adotada a política de maior conhecimento descrita anteriormente.

Fica claro que o contexto do projeto ou as suas necessidades específicas influenciam na maneira de proceder com o processo de seleção de equipes, sendo necessário que a abordagem proposta permita ao gerente de projetos alterar as políticas de seleção de acordo com a necessidade de cada projeto, também possibilitando ao gerente criar outras políticas, constituindo assim uma abordagem flexível.

As políticas de seleção são representadas com uma estrutura de tabelas, onde as colunas e linhas representam respectivamente as *habilidades técnicas* fuzzificadas das equipes e os níveis de complexidade esperados para o desenvolvimento de um módulo de *software* no tocante a apenas uma tecnologia demandada por ele. As células na tabela representam o nível de adequabilidade que uma equipe deve possuir, avaliando-a para um módulo de *software*. Um exemplo de uma política de seleção pode ser observado na Tabela 2. Perceba que os níveis utilizados para medir a adequabilidade das equipes também são termos vagos e, como esperado, representam conjuntos *fuzzy*. Os conjuntos fuzzy utilizados para caracterizar a adequabilidade das equipes são apresentados na Figura 9.

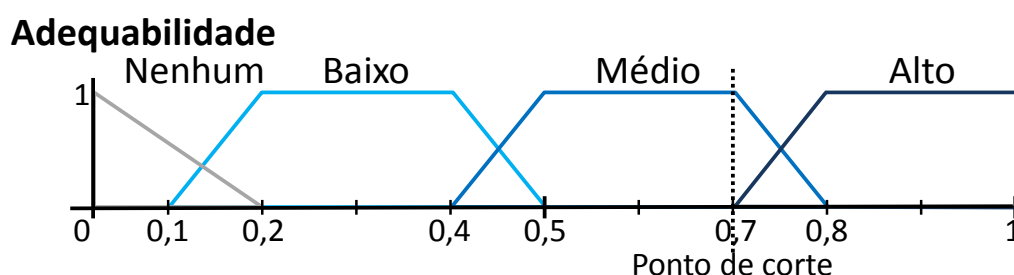


Figura 9 Conjuntos Fuzzy de representação da adequabilidade das equipes

Assim como os demais conjuntos *fuzzy* apresentados até então, o gerente de projeto tem autonomia para modificar os termos, o número de conjuntos, posição e formato das figuras que o representam (são aceitos os formatos *singleton*, triângulo e trapézio). Observe que o gerente de projetos deve definir um ponto no eixo horizontal da representação dos conjuntos *fuzzy* para a representação de adequabilidade. Este ponto será usado posteriormente na etapa de *Avaliação técnica das equipes*, na qual servirá como um ponto de corte, ou seja, apenas equipes com adequabilidade acima deste ponto serão consideradas como equipes adequadas para implementar o módulo. Ao valor padrão de ponto de corte foi atribuído o valor 0,7, pois é ponto que separa equipes com valor de adequabilidade média/alta ou alta das demais equipes.

Tabela 2. Exemplo de política de seleção

		Complexidade de tecnologia dos módulos		
		Baixo	Médio	Alto
Habilidade Técnica das equipes	Nenhum	Nenhum	Nenhum	Nenhum
	Baixo	Alto	Baixo	Baixo
	Médio	Médio	Baixo	Médio
	Alto	Baixo	Médio	Alto

A política de seleção pode ser entendida como uma tabela de regras de condições do tipo **IF-THEN** que relacionam os termos das linhas e colunas numa avaliação que produz o resultado representado na célula correspondente. Perceba que o número de regras que uma política possui é equivalente ao número de linhas multiplicado pelo número de colunas. No caso do exemplo citado, 4×3 ou 12 regras existentes. Um pequeno conjunto de regras representado através de condições **IF-THEN** pode ser visto como:

IF *Complexidade de tecnologia dos módulos* IS **Baixo** AND *Habilidade Técnica das equipes* IS **Nenhum** THEN adequabilidade IS **Nenhum**;

IF *Complexidade de tecnologia dos módulos* IS **Médio** AND *Habilidade Técnica das equipes* IS **Baixo** THEN adequabilidade IS **Baixo**;

IF *Complexidade de tecnologia dos módulos* IS **Alto** AND *Habilidade Técnica das equipes* IS **Alto** THEN adequabilidade IS **Alto**;

Como pode ser visto, sempre que os dados das equipes são cruzados com dados de um módulo de *software*, uma regra da tabela é acionada. Por esta razão, os níveis de adequabilidade presentes na tabela de regras devem ser completamente definidos. Caso algum valor na tabela não seja desejado, pode-se usar o termo **Nenhum** que identificará que uma equipe não possui nenhuma adequabilidade esperada.

As regras são definidas de forma a corresponder às políticas que podem ser adotadas durante o processo de seleção das equipes. Desse modo, foram definidas neste trabalho inicialmente quatro políticas distintas. As políticas definidas foram baseadas a partir de outros trabalhos presentes na literatura [Barreto, 2005] [Silva, 2007] [Callegari, et al., 2009] [Collofello, et al., 1998]. Apesar de serem sugeridas quatro políticas, o gerente de projetos deve possuir autonomia para criar novas políticas ou refinar as políticas existentes. Dessa forma, políticas não previstas na abordagem, mas adotadas pelas empresas, podem ser representadas e utilizadas. As subseções seguintes definem cada uma dessas quatro políticas.

4.3.1 Política de qualificação equivalente

Essa política visa selecionar equipes que possuam as habilidades técnicas mais próximas possíveis da complexidade de implementação esperada para o desenvolvimento dos módulos de *software*. Esse tipo de política é denominada aqui como **política de qualificação equivalente**. Com a adoção desse tipo de política evita-se que equipes com um conhecimento alto sejam alocadas para a implementação de módulos de *software* que não demandem um conhecimento profundo acerca de uma determinada tecnologia. Dessa forma, deixando-as disponíveis para atividades mais complexas. Do mesmo modo, equipes caracterizadas com pouca *habilidade técnica* acerca de algumas tecnologias podem participar dos projetos sendo boas opções para módulos de *software* que demandem baixa *habilidade técnica*. Perceba que na política apresentada, mesmo que uma equipe possua um nível de conhecimento abaixo do que o esperado para o desenvolvimento do módulo de *software*, essa equipe ainda pode ser uma opção considerável desde que os níveis sejam próximos ou equivalentes. Quão mais distantes os níveis, seja pra mais ou pra menos, menor se torna a adequabilidade. O uso dessa política pode trazer um bom custo-benefício ao projeto, já que as equipes são selecionadas de acordo com as necessidades e não por serem as que apresentam maiores habilidades técnicas. Esta política é apresentada na Tabela 3.

Tabela 3. Política de qualificação equivalente.

		Complexidade de tecnologia dos módulos		
		Baixo	Médio	Alto
Habilidade Técnica das equipes	Nenhum	Médio	Baixo	Nenhum
	Baixo	Alto	Médio	Baixo
	Médio	Médio	Alto	Médio
	Alto	Baixo	Médio	Alto

4.3.2 Política de equipes mais hábeis

O segundo tipo de política que pode ser adotada é a **política de seleção de equipes mais hábeis**, que tende a dar preferência às equipes que possuam os maiores níveis de habilidades técnicas apresentados, distanciando-as das demais. Observando as regras geradas para essa política, percebe-se que equipes que possuem conhecimentos mais altos sempre levam vantagem sobre as que possuem conhecimentos abaixo dos apresentados por estas, pois essa política trata apenas de selecionar as equipes que possuem o maior conhecimento, independente do conhecimento exigido para a implementação do módulo de *software*. Esse tipo de política pode ser utilizada para acelerar o projeto, selecionando sempre as equipes que apresentam os níveis de *habilidades técnicas* mais altos. Teoricamente, estas equipes podem realizar o trabalho em menor tempo. Contudo, pode ser que os custos decorrentes dessa decisão sejam maiores cabendo ao gerente de projetos arcar com suas decisões baseando-se nas necessidades específicas do projeto. Tal política pode ser observada na Tabela 4.

Tabela 4. Política de equipes mais hábeis.

		Complexidade de tecnologia dos módulos		
		Baixo	Médio	Alto
Habilidade Técnica das equipes	Nenhum	Nenhum	Nenhum	Nenhum
	Baixo	Baixo	Baixo	Baixo
	Médio	Médio	Médio	Médio
	Alto	Alto	Alto	Alto

4.3.3 Política de qualificação padrão

Como mencionado, o gerente de projetos pode ajustar as políticas a seu critério para que as mesmas se adequem o melhor possível às necessidades de cada projeto. Por exemplo, alterando algumas regras da política de qualificação equivalente apresentada na Tabela 3, pode-se criar uma política que considere igualmente equipes que possuam as habilidades técnicas iguais ou superiores à complexidade esperada do módulo. Deste modo equipes passam a concorrer de igual para igual sempre que as mesmas possuírem o conhecimento mínimo exigido para o desenvolvimento de um módulo de *software*. Essa política pode ser observada na Tabela 5. Note que os termos alterados estão dispostos em negrito.

Tabela 5. Política de qualificação padrão.

		Complexidade de tecnologia dos módulos		
		Baixo	Médio	Alto
Habilidade Técnica das equipes	Nenhum	Nenhum	Nenhum	Nenhum
	Baixo	Alto	Médio	Baixo
	Médio	Alto	Alto	Médio
	Alto	Alto	Alto	Alto

4.3.4 Política de provisão de treinamento

As políticas podem também favorecer o ganho de conhecimento e experiência por parte das equipes. Suponha que ao submeter as equipes e módulos de *software* à política de qualificação equivalente, apresentada na Tabela 3, para alguns módulos de *software* nenhuma equipe tenha sido identificada como qualificada, pois as equipes apresentam pouco ou nenhum conhecimento acerca do domínio ou tecnologias envolvidas. Uma solução possível é utilizar a tabela de regras como uma ferramenta para derivar seleções de equipes cujo intuito seja prover experiência e treinamento para estas. Segundo a norma NBR ISO 10006 (2000) [ABNT, 2000], quando forem verificadas dificuldades na seleção de pessoas devido a requisitos de competência, convém que seja permitido treinamento adicional. Logo, essa política favorecerá a seleção de equipes cujas habilidades técnicas sejam mais baixas do que o esperado para a implementação do módulo de *software*. Note que se o conhecimento for muito aquém, essa equipe passa a não ser apta para um treinamento avançado, pois a mesma pode não apresentar os conhecimentos básicos necessários para tirar proveito do treinamento. Essa política pode ser vista na Tabela 6. Dependendo das necessidades de cada projeto, pode-se criar um projeto fictício com o intuito de verificar o nível de conhecimento técnico apresentados pelas equipes. Nesse caso, o processo de seleção teria a finalidade exclusiva de identificar os pontos fracos das equipes e prover treinamentos para elas.

Tabela 6. Política de provisão de treinamento.

		Complexidade de tecnologia dos módulos		
		Baixo	Médio	Alto
Habilidade Técnica das equipes	Nenhum	Alto	Médio	Baixo
	Baixo	Alto	Alto	Médio
	Médio	Médio	Alto	Alto
	Alto	Baixo	Médio	Alto

4.4 Avaliação técnica das equipes

Neste ponto, todos os requisitos necessários para prosseguir com o processo de avaliação técnicas das equipes estão agora reunidos. Dessa forma, a avaliação das equipes é feita através da seguinte lógica: *i)* Primeiramente cada equipe é avaliada para o mesmo módulo de *software* e em seguida essas equipes são ordenadas pelo nível de adequabilidade calculado para esse módulo. Este passo é repetido para todos os outros módulos de *software* identificados até não restar mais nenhum módulo a ser analisado. *ii)* O atributo de *reputação* explicado na etapa *Descrição técnica das equipes* é utilizado em conjunto com a adequabilidade calculada de forma a reordenar as equipes consideradas adequadas, favorecendo aquelas que possuem níveis mais altos de *reputação*. *iii)* Finalmente, com as equipes ordenadas, aquelas cuja adequabilidade estiver abaixo do ponto de corte estarão fora da lista de seleção, gerando assim a saída desta abordagem, o *Mapeamento equipes × módulos*.

Para explicar o primeiro passo, suponha um módulo de *software* M e uma equipe de desenvolvimento E . Neste ponto, já foi possível calcular a habilidade técnica da equipe E para cada uma das tecnologias identificadas na *tabela de implementação* do módulo M . Escolhendo uma tecnologia t identificada na *tabela de implementação* do módulo M , é possível calcular a adequabilidade da equipe em relação a essa tecnologia através da política de seleção escolhida e representada pela tabela de regras. Considerando o processo de cruzamento das informações dos módulos e das equipes na tabela de regras, note que mais de uma regra pode ser acionada, como explicado anteriormente, gerando mais do que um valor de adequabilidade apenas para esta tecnologia t . É neste ponto que os conjuntos *fuzzy* de representação de adequabilidade desempenham o seu papel. O termo resultante, obtido através da análise das regras da política selecionada, é mapeado para os conjuntos *fuzzy* de representação de adequabilidade com o mínimo dos valores de pertinência entre os conjuntos *fuzzy* referentes à habilidade técnica da equipe E e da complexidade da tecnologia t do módulo M . Como para os módulos a pertinência em qualquer conjunto é sempre 1, o nível de pertinência calculada da adequabilidade equivale ao nível de pertinência da habilidade técnica. Um exemplo gráfico será abordado para uma melhor compreensão.

Considere o seguinte conjunto de regras e os conjuntos *fuzzy* apresentados na Figura 10:

IF *Complexidade de tecnologia dos módulos IS Baixo* AND *Habilidade Técnica das equipes IS Baixo* THEN *adequabilidade IS Alto*;

IF *Complexidade de tecnologia dos módulos IS Baixo* AND *Habilidade Técnica das equipes IS Médio* THEN *adequabilidade IS Médio*;

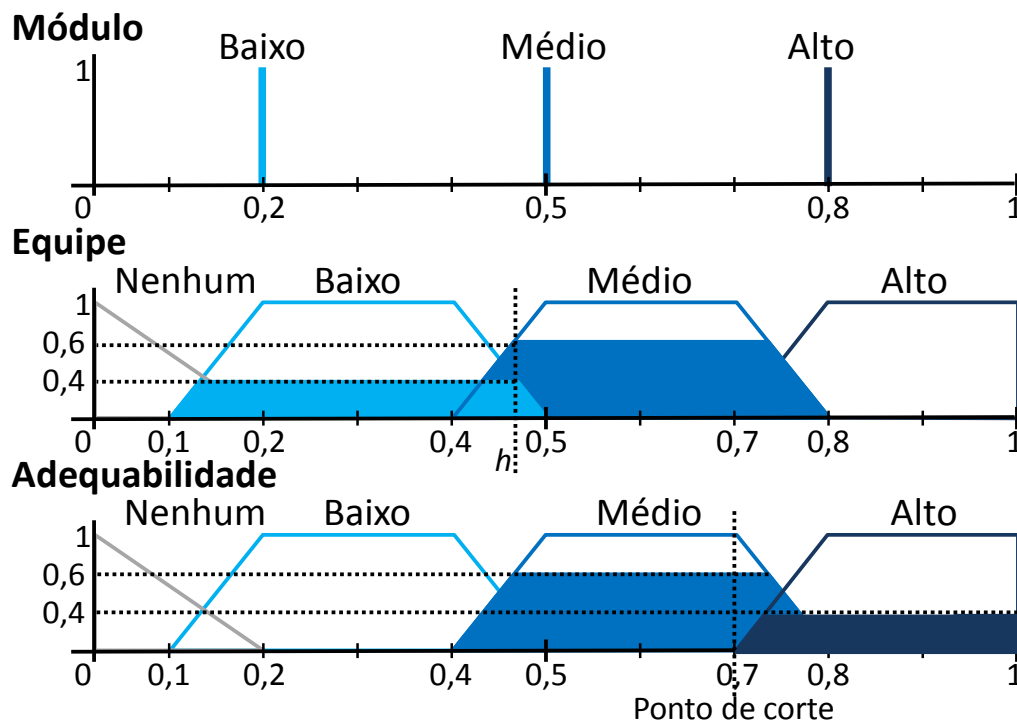


Figura 10. Cálculo de adequabilidade.

Considerando que uma equipe E possui habilidade técnica h numa tecnologia t , observando a figura, percebe-se que o grau de habilidade da equipe E é caracterizada como Baixo (pertinência 0.4) e Médio (pertinência 0.6) simultaneamente. Perceba que, segundo as regras apresentadas, o nível de complexidade esperado para a tecnologia t é Baixo (pertinência 1). Sendo assim a primeira regra informa que se a complexidade é *baixa* e a habilidade técnica da equipe também é *baixa*, a adequabilidade se torna *alta*, portanto a adequabilidade se encontra no conjunto representado pelo termo ‘Alto’. Como o conjunto *baixo* do módulo tem pertinência 1 e o *baixo* da equipe tem pertinência 0.4, será determinado como valor de pertinência da adequabilidade o menor valor entre os dois (assim como definido pela propriedade AND da lógica *fuzzy*), portanto o valor atribuído será 0.4. De fato a figura mostra que existe uma pertinência de valor 0.4 no conjunto ‘Alto’ da adequabilidade. Do mesmo modo, utilizando a segunda regra que informa que se a complexidade é *baixa* e a

habilidade técnica da equipe é *média*, a adequabilidade se torna *média*, e sendo 0.6 o menor dos dois valores de pertinência, a adequabilidade é mapeada para o conjunto ‘Médio’ com pertinência 0.6. Assim, pode-se dizer que a adequabilidade da equipe E para a implementação da tecnologia t do módulo M está entre média e alta ou é mais média que alta.

Note que esse cálculo de adequabilidade é referente a apenas uma das tecnologias da *tabela de implementação*. Desse modo, esse cálculo de adequabilidade deve ser efetuado para todas as tecnologias identificadas na *tabela de implementação* do módulo. Como o objetivo final da abordagem de seleção é apresentar uma lista ordenada das equipes mais adequadas para cada módulo, junto com uma métrica de adequabilidade estabelecida entre 0 e 1, é imperativo que se converta os termos de adequabilidade *fuzzy* em números reais novamente. Isto é feito através do processo de *defuzzificação*. Esse processo gera novamente um valor real com base na área total obtida com a pertinência dos termos de adequabilidade. Este cálculo é feito através da Equação (4.0):

$$Adq^* = \frac{\sum_{k=1}^n \mu_{C_k}(z) \int_z \bar{z} dz}{\sum_{k=1}^n \mu_{C_k}(z) \int_z dz} \quad (4)$$

A Equação 4 é um método de *defuzzificação* chamado *Center of Sums* [Ross, 2010], que denota que $\sum_{k=1}^n \mu_{C_k}(z) \int_z dz$ são as áreas dos conjuntos de adequabilidade e $\bar{z}$ é a distância do centroide (ponto médio entre o ponto inicial de um conjunto e o ponto final, considerando o plano horizontal) de cada uma das respectivas funções de pertinência (graficamente os conjuntos). Em outras palavras, esse método divide a soma das áreas de adequabilidade calculadas multiplicado pela distância ao centroide do conjunto, pela soma das áreas de adequabilidade calculadas. Esse método foi adotado (em detrimento a outros apresentados no Capítulo 2), pois além de gerar um valor real único, relativamente no meio das áreas de adequabilidade calculadas, não apresenta um custo computacional elevado como o *Centroid Method*, mas gera valores mais precisos que o *Weighted Average Method*, ambos métodos que, assim como o *Center of Sums*, tendem a descobrir o valor defuzzificado utilizando cálculo de área das figuras geométricas formadas.

O valor encontrado é usado como uma representação em número real da adequabilidade *fuzzy* calculada. Seu valor está limitado aos valores encontrados no eixo horizontal dos conjuntos de adequabilidade, portanto variando de 0 a 1. Através da utilização de um número real, é possível ordenar as equipes pelo seu nível de adequabilidade calculado.

É necessário reforçar que o processo para descobrir o valor real da adequabilidade foi realizado apenas para uma tecnologia t contida na tabela de implementação do módulo de *software*. Este processo deve então ser realizado para cada uma das tecnologias identificadas. Assim que todos os valores de adequabilidade real para todas as tecnologias contidas na tabela de implementação do módulo forem calculadas, estas são somadas e divididas pelo número de tecnologias da tabela, como pode ser visto na Equação 5:

$$Adeq = \frac{\sum_{k=1}^n Adq_k^*}{n} \quad (5)$$

Na Equação 5, n é o número de tecnologias contidas na tabela de implementação e Adq_k^* são as n adequabilidades reais calculadas. Essa média gera um valor real, único no intervalo de 0 a 1 que representa a adequabilidade técnica de uma equipe em relação à implementação de um módulo de *software* analisado. Assim que todas as equipes passem por esse processo para descoberta da adequabilidade técnica de um módulo, esse processo será repetido até não sobraem mais módulos a serem analisados. Quando isso ocorrer, é gerada a primeira versão do artefato de saída da abordagem, o *Mapeamento equipes × módulos*. Um exemplo desse artefato pode ser observado na Tabela 7.

Tabela 7. Primeira versão do *Mapemaneto equipes × módulos*

M0		M1		M2	
EQUIPE 3	0,8299	EQUIPE 1	0,5602	EQUIPE 4	0,6196
EQUIPE 1	0,6256	EQUIPE 3	0,5570	EQUIPE 3	0,6196
EQUIPE 2	0,4143	EQUIPE 2	0,5000	EQUIPE 1	0,5699
EQUIPE 4	0,0728	EQUIPE 4	0,1093	EQUIPE 2	0,0058

Note que este artefato mostra para cada módulo de *software* todas as equipes candidatas ordenadas pelo valor de adequabilidade técnica calculada. Os valores, bem como a ordem das equipes, vão variar de acordo com a política utilizada.

O *Mapeamento equipes × módulos* passará agora pelo penúltimo passo da abordagem, que consiste em reordenar as equipes levando em consideração, além da adequabilidade técnica calculada, seus atributos de *reputação*. Este dado, como argumentado anteriormente,

representa o quão bem uma equipe realizou o desenvolvimento de módulos de *software* em projetos anteriores. Esta reputação indiretamente trata aspectos de desempenho, consumo de tempo e qualidade prática das equipes. Como o processo de descoberta de conhecimento é inerentemente subjetivo é possível que duas equipes que apresentem habilidades técnicas semelhantes apresentem desempenhos diferentes ao longo do tempo. Assim, o atributo de reputação é responsável para representar estes aspectos, nivelando a subjetividade com a prática real das equipes, uma vez que este valor é adquirido através de observações das equipes por meio de projetos desenvolvidos.

O valor da reputação está contido no intervalo $[0, 1]$, e todas as equipes começam com o valor de reputação 0,5 (valor neutro para a abordagem). Quando uma equipe tem uma reputação maior que 0,5, significa que esta equipe tem demonstrado um bom desempenho através de projetos que desenvolveu. Sendo assim, essa equipe será bonificada, aumentando o valor de adequabilidade ligeiramente. Se a reputação é menor do que 0,5, o processo é inverso, em outras palavras, isso significa que a equipe vem demonstrando um desempenho fraco por meio de projetos que desenvolveu, portanto, esta equipe será penalizada e a adequabilidade técnica será diminuída ligeiramente. Se a reputação é igual a 0,5, isso quer dizer que a equipe tem um desempenho regular ou é a primeira vez que desenvolve um projeto e, portanto, não há nenhuma maneira de saber realmente o desempenho da equipe, portanto, a adequabilidade técnica continuará a ser a mesma, sem alteração em seu valor.

O bônus ou penalidade que será aplicado ao valor da adequabilidade das equipes com base no valor de sua reputação é calculado através da Equação 6:

$$Adequabilidade(m, e) = Adeq_{m,e} * \frac{0,5}{R_e} \quad (6)$$

Na Equação 6, a variável $Adeq_{m,e}$ representa o valor de adequabilidade atual da equipe e para o módulo m e R_e é o valor da reputação da equipe. Note que assim como na Equação 3 o fator exponencial da Equação 4 foi inspirada em modificadores *fuzzy*. Nesse caso unindo um intensificador *fuzzy* [Ross, 2004] que hora tende a aumentar o e hora tende a diminuir o valor calculado de acordo com o valor de R_e . Além disso duas situações devem ser notadas em relação ao cálculo da adequabilidade com reputação. A primeira situação se encontra no caso de R_e ser zero. Nesse caso existiria um problema de divisão por zero, contudo, como será visto mais adiante, o valor de reputação de uma equipe nunca será zero, pois o cálculo de atribuição de reputação das equipes não permite chegar a esse valor. A

segunda situação se encontra no caso da adequabilidade ser um. Nesse caso independentemente do valor de reputação da equipe a adequabilidade seria sempre um, visto que um elevado a qualquer valor é sempre um. Novamente essa situação não é possível na abordagem proposta, visto que mesmo que uma equipe tivesse o valor de adequabilidade calculado com o valor *fuzzificado* ‘Alto’, ao *defuzzificar* esse valor utilizando o método *Center of Sums*, o valor calculado se encontrará em algum lugar perto do centro da figura que representa o maior valor ‘Alto’, ou seja, no máximo tende a chegar em um.

Depois de aplicar este procedimento para todas as equipes, estas são reordenadas e gerado o artefato *Mapeamento equipes × módulos*. Com o intuito de filtrar as equipes que por ventura possuam um nível de adequabilidade muito baixo, é utilizado o ponto de corte definido na etapa de Definição de políticas de seleção. Essa operação consiste simplesmente em eliminar aquelas equipes que não atingiram o ponto de corte. Essa operação gera a segunda versão do *Mapeamento equipes × módulos* que consiste na mesma tabela anterior com exceção das equipes que não atingiram o ponto de corte. Considerando o valor padrão 0.7 de ponto de corte adotado nesta abordagem, a segunda versão do artefato ficaria como apresentado na Tabela 8. Observe que as equipes que permaneceram são aquelas que possuem o valor de adequabilidade técnica igual ou superior ao ponto de corte estabelecido. Com esse passo, as equipes que permaneceram são as que possuem competências técnicas adequadas para a implementação do módulo.

Finalmente o artefato *Mapeamento equipes × módulos* atinge o seu estado final e pode ser apresentado ao gerente de projeto. O gerente pode então usar sua experiência e conhecimento para selecionar as equipes com muito mais propriedade e confiança.

Tabela 8. Versão final do *Mapemaneto equipes × módulos*

M0		M1		M2	
EQUIPE 3	0,7299	EQUIPE 1	0,7352	EQUIPE 1	0,7789
EQUIPE 1	0,7066	EQUIPE 2	0,61235	EQUIPE 4	0,6196
		EQUIPE 3	0,4920	EQUIPE 3	0,5109

4.5 Cálculo do valor de reputação

O valor do atributo *reputação*, como já mencionado anteriormente, representa quão bem uma equipe consegue desempenhar tecnicamente seu trabalho de acordo com a ótica do gerente de projetos. Desse modo, independente das respostas indicadas pelas equipes nos questionários de avaliação de habilidade técnica, o gerente tem como informar à abordagem de seleção sua impressão sobre a qualidade do desenvolvimento dos módulos de *softwares*, estabelecida empiricamente através do gerenciamento dessas equipes na prática. Assim é possível contrabalancear as informações dadas pelas equipes com os respectivos desempenhos reais evidenciados pelo gerente do projeto.

Para avaliar a reputação de uma equipe na prática, é necessário que primeiramente uma equipe seja escolhida, pela abordagem proposta, para o desenvolvimento de um módulo de *software*. Ao final do desenvolvimento do módulo de *software*, o gerente de projetos irá verificar se o desempenho da equipe corresponde com o esperado e irá preencher sua impressão sobre essa equipe com um dos seguintes valores: *muito insatisfeito*, *insatisfeito*, *indiferente*, *satisfeito* e *muito satisfeito*. O preenchimento deve levar em conta o que era esperado dessa equipe de acordo com a solução de seleção de equipes recomendada pela abordagem. Desse modo, caso tenha sido sugerido uma equipe com uma ótima qualificação para o desenvolvimento do módulo de *software*, contudo na prática a mesma obteve um desempenho péssimo, o gerente de projetos deverá preencher, possivelmente, o valor *muito insatisfeito* ou *insatisfeito*. Pensando o contrário, caso tenha sido sugerido uma equipe com uma qualificação baixa, mas a mesma tenha sido selecionada e na prática tenha obtido um ótimo desempenho, o gerente de projetos deve preencher, possivelmente, o valor *satisfeito* ou *muito satisfeito*. Por outro lado, caso a equipe tenha um desempenho prático de acordo com o esperado, como, por exemplo, uma equipe qualificada como boa obtenha um desempenho bom, ou uma equipe com baixa qualificação demonstre baixo desempenho real, o gerente de projetos deve preencher, possivelmente, o valor de *indiferente*, pois a mesma desempenhou suas atividades com a qualidade prevista pela abordagem.

Na prática, a escolha de um valor para classificação da reputação da equipe segue o mesmo esquema apresentado na Equação 1, onde o valor numérico está no intervalo $[0, 1]$, e os valores definidos para cada termo são calculados como: $\{0\}$ (*muito insatisfeito*), $\{0,25\}$ (*insatisfeito*), $\{0,5\}$ (*indiferente*), $\{0,75\}$ (*satisfeito*) e $\{1\}$ (*muito satisfeito*). Quando uma equipe é selecionada pela primeira vez, a mesma ainda não foi avaliada pelo gerente de

projetos e não há como inferir um valor para sua reputação, assim o valor 0,5 (indiferente) é atribuído à equipe. Ao final, o cálculo para a nova reputação é realizado de forma a alterar gradativamente a reputação da equipe. O novo valor é definido com base no valor atual rep_{atual} e o *feedback* do gerente $rep_{feedback}$, como pode ser visto na Equação 7. Para manipular a intensidade da modificação foram definidos dois pesos, w_{rep} que define o peso da reputação atual e w_{feed} que define o peso do *feedback* do gerente. Os valores utilizados como padrão são $\{w_{rep} = 2\}$ e $\{w_{feed} = 1\}$. Com essa configuração padrão se beneficia a reputação atual, mas ainda mantendo o *feedback* do gerente relevante.

$$reputação = \frac{rep_{atual} \times w_{rep} + rep_{feedback} \times w_{feed}}{w_{rep} + w_{feed}} \quad (7)$$

4.6 Considerações Finais

Foi apresentada neste capítulo a abordagem de seleção de equipes distribuídas tecnicamente qualificadas para projetos de *software*. Essa abordagem se propõe como um auxílio à seleção de equipes para um projeto de *software* e, como tal, deve permitir que o gerente de projetos intervenha no processo de avaliação, refinando os atributos para que estes possam condizer com as especificidades únicas de cada projeto.

Esta abordagem está dividida em quatro etapas bem definidas e com artefatos de entrada e saídas também bem definidos de modo que novas etapas possam ser incluídas no futuro caso se julgue necessário. Dentre estas etapas, as etapas de Descrição dos módulos e de Descrição técnica das equipes possuem um papel fundamental, pois a maior parte dos requisitos necessários para que a seleção de equipes funcione é levantada nessas fases. Em especial ao que se refere à descrição das equipes, que é inerentemente um processo bastante subjetivo e portanto apresenta grandes dificuldades quanto à medição e representação de cada atributo.

Outro ponto que vale ser ressaltado é a utilização de políticas de seleção dos profissionais. Apesar de serem apresentadas políticas pré-existentes de seleção, possibilitando que o gerente gere avaliações sem a necessidade de configurações, a abordagem proposta é flexível o bastante para permitir ao gerente de projetos refinar e criar novas políticas de seleção, para que este se torne mais adequado às necessidades do projeto.

Capítulo 5

Estudo de casos

Neste capítulo são apresentados quatro estudos de casos realizados com o propósito de testar e avaliar a abordagem de seleção de equipes desenvolvida, sendo dois projetos de *software* diferentes, cada um com dois grupos diferentes de equipe. Vale à pena lembrar que a abordagem de seleção de equipes, apesar de poder ser utilizada separadamente, está inserida num *framework* de alocações mais abrangente, e que recebe como entrada artefatos de outra fase desse mesmo *framework*, bem como em seu resultado final produz uma entrada para outra fase do mesmo *framework*.

O primeiro estudo de caso a ser apresentado é uma linha de produtos de *software* hipotética, documentada em [Gomaa 2004]: uma linha de produto de comércio eletrônico (E-Commerce) que foi organizada em duas iterações de desenvolvimento (consistindo na engenharia de domínio e engenharia de aplicação) para avaliar o uso da abordagem nas duas fases de uma LPS. Em seguida, é apresentado um estudo de caso de projeto real com participação do arquiteto do projeto na construção de alguns artefatos previstos na abordagem. Este projeto é designado como Multi-MOM [Moraes et. al. 2010], uma linha de produtos de *middleware* para computação móvel. Para ambos os casos foi utilizado dois conjuntos diferentes de equipes a fim de verificar se em ambos os casos equipes semelhantes seriam selecionadas e ranqueadas também de forma semelhante.

Para um melhor entendimento de ambos os estudos, serão mostrados as arquiteturas dos mesmos. Contudo, como essas arquiteturas são provenientes de uma fase anterior do *framework* de alocação, cabe salientar um detalhe: os pontos de variação da LPS no documento arquitetural são representados por um componente genérico que contém mais de um índice, cada um deles correspondendo a um componente variante. Essa representação genérica abstrai as diferenças entre os componentes variantes, sendo adotada apenas para prover uma representação única da arquitetura.

5.1 Configurações utilizadas

Para todos os estudos de caso realizados foram utilizadas as mesmas configurações dos conjuntos e funções de pertinências *fuzzy*, sendo essas apresentadas como configurações padrões. As configurações utilizadas são as apresentadas na Figura 11, onde os valores de entrada para o protótipo da abordagem são as seguintes:

- **Conjuntos *fuzzy* de representação da complexidade esperada de uma tecnologia** – Baixo (0.2, 0.2, 0.2, 0.2); Médio (0.5, 0.5, 0.5, 0.5); Alto (0.8, 0.8, 0.8, 0.8);
- **Conjuntos *fuzzy* de representação da habilidade técnica das equipes** – Nenhum (0.0, 0.0, 0.0, 0.2); Baixo (0.1, 0.2, 0.4, 0.5); Médio (0.4, 0.5, 0.7, 0.8); Alto (0.7, 0.8, 1, 1);
- **Conjuntos *fuzzy* de representação da adequabilidade técnica das equipes para os módulos de *software*** – Nenhum (0.0, 0.0, 0.0, 0.2); Baixo (0.1, 0.2, 0.4, 0.5); Médio (0.4, 0.5, 0.7, 0.8); Alto (0.7, 0.8, 1, 1);

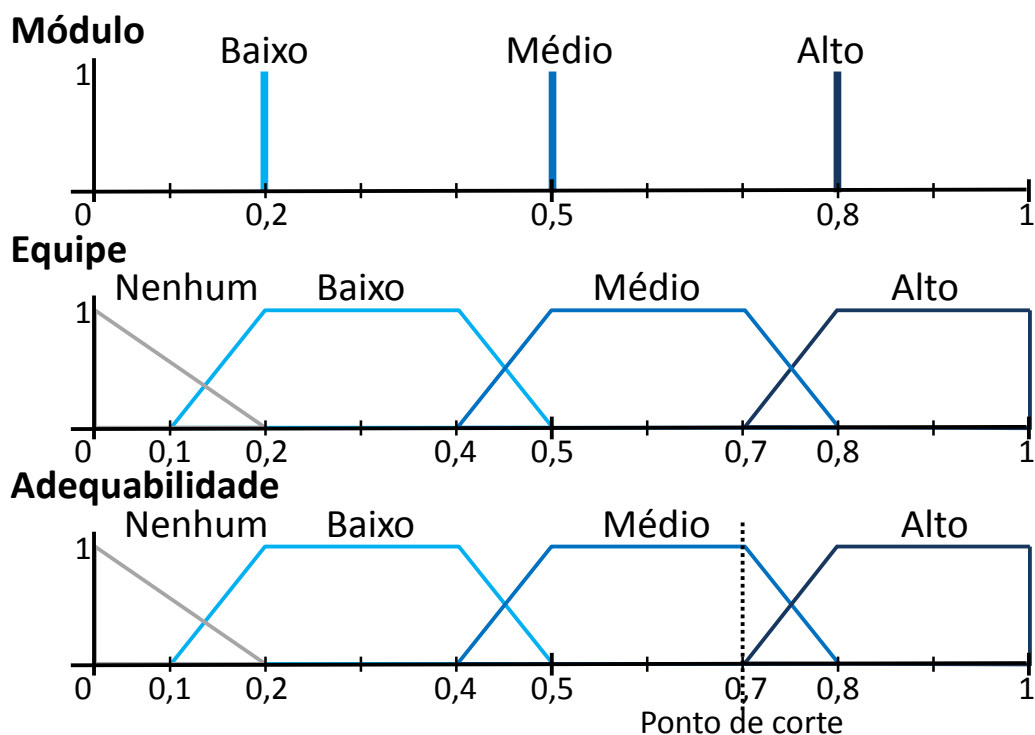


Figura 11. Configurações dos conjuntos *fuzzy*

Para as configurações dos pesos de importância w_y , w_p e w_t , respectivamente representando pesos para os anos de experiência, o número de projetos desenvolvidos e o

número de títulos que um membro m apresenta em uma determinada tecnologia, foram utilizados os valores padrões $w_y = 4$, $w_p = 4$ e $w_t = 2$.

Quanto ao ponto de corte foi utilizado o valor padrão 0.7.

Também foi utilizado para cada um dos estudos de caso o mesmo conjunto de políticas de seleção, apresentadas no Capítulo 4. Como foram utilizadas quatro políticas distintas, foram obtidos, para cada estudo de caso, quatro resultados distintos. Contudo, como essa configuração acaba gerando 24 resultados (2 projetos, sendo um deles subdividido em engenharia de domínio e de aplicação; 2 conjuntos de equipes; 4 políticas), nem todos os resultados serão mostrados para não estender desnecessariamente a leitura do presente trabalho. Assim, serão apresentados os resultados mais significativos, mesclando políticas diferentes para projetos diferentes.

Além disso, vale ser ressaltada a maneira como as equipes de desenvolvimento tomaram parte no estudo de caso. Como para nenhum dos dois estudos de caso existiam equipes reais de desenvolvimento, equipes fictícias precisaram ser montadas. Contudo, os membros de cada equipe são desenvolvedores atuantes no mercado e/ou estudantes de cursos de Ciência da Computação. O procedimento tomado foi o seguinte: como o desenvolvimento dos questionários que visam a capturar atributos técnicos necessita ser feito a partir das tabelas de implementação dos projetos, para o estudo caso hipotético E-Commerce, o próprio autor do presente trabalho se encarregou de levantar um conjunto de tecnologias que o mesmo julgou cabíveis de serem utilizadas num projeto real. Para o estudo de caso Multi-MOM, o arquiteto e criador do Multi-MOM levantou todas as tecnologias envolvidas para cada módulo. Com base nas informações levantadas, no conjunto de tecnologias e domínios a serem utilizados em ambos os estudos de casos, foi desenvolvido um questionário único contemplando todas as informações necessárias para coleta dos atributos técnicos das equipes. Esse questionário pode ser observado no Anexo I. Como mencionado, esse questionário foi respondido por estudantes de Ciência da Computação e desenvolvedores reais de empresas locais. A aplicação do questionário foi feita de forma *online* e 56 pessoas responderam os questionários. Depois o mesmo questionário foi repetido para outro grupo de pessoas, obtendo 123 respondentes.

A partir desse conjunto de pessoas foram desenvolvidos dois conjuntos distintos de equipes. Primeiramente para o conjunto A de equipes foi estipulado que cada equipe teria no máximo 20 membros, assim foi sorteado um número aleatório x entre 1 e 20 que

corresponderia ao número de integrantes de uma equipe. Eram sorteados x participantes que eram retirados da lista e com eles era formada uma equipe. Repetiu-se o procedimento até não sobrares participantes. Assim resultou-se em 7 equipes com 2, 5, 5, 8, 9, 9 e 18 participantes. Para a formação do segundo conjunto de equipes, o conjunto B, foi estipulado que o número máximo de membros por equipe seria 15 e a distribuição de membros fosse de tal forma que obtivesse algumas equipes com o mesmo tamanho e que aumentassem gradualmente de tamanho. Desse modo obtendo-se dois grupos com formação distintas, sendo um formado de modo aleatório (conjunto A) e outro de modo conduzido (conjunto B). Assim foram geradas, para o conjunto B, 3 equipes com 3 membros, 3 equipes com 5 membros, 3 equipes com 8 membros, 3 equipes com 10 membros e 3 equipes com 15 membros. O mesmo procedimento de sorteio de membros foi realizado para escolha de qual membro faria parte de qual equipe.

Finalmente, assim que todas as equipes foram formadas, foi sorteado aleatoriamente para cada equipe um número real no intervalo $[0, 1]$ com precisão de quatro casas decimais, para representar a *reputação* de cada uma das equipes. Isso foi necessário para simular a utilização da aplicação considerando que equipes já tivessem participado de outros projetos e sido avaliadas pelo gerente de projetos.

Agora que já é conhecida toda a configuração utilizada para realização dos estudos de caso, as Seções 5.2 e 5.3 explorarão com detalhes os estudos de caso.

5.2 Linha de Produto de Comércio Eletrônico

A linha de produto de comércio eletrônico consiste em uma LPS altamente distribuída baseada na *World Wide Web* que controla tanto sistemas B2B (*business-to-business*) como B2C (*business-to-consumer*) [Gomaa 2004]. Essa linha de produtos foi desenvolvida segundo um processo de engenharia reversa.

A fim de demonstrar o uso da presente abordagem para o desenvolvimento de um projeto em várias iterações, esta LPS será implementada em duas iterações: uma primeira iteração caracteriza a engenharia de domínio, onde serão implementados os componentes comuns e variantes, e uma segunda iteração para desenvolver as aplicações B2B e B2C, em conjunto.

5.2.1 Primeira Iteração (Engenharia de domínio)

Nesse projeto existem alguns componentes variantes que não aparecem na arquitetura de domínio, já que apenas uma das variações é escolhida. Assim os componentes variantes são

apresentados na Figura 12 e a arquitetura de domínio, contemplando os componentes desse projeto, é apresentada na Figura 13.

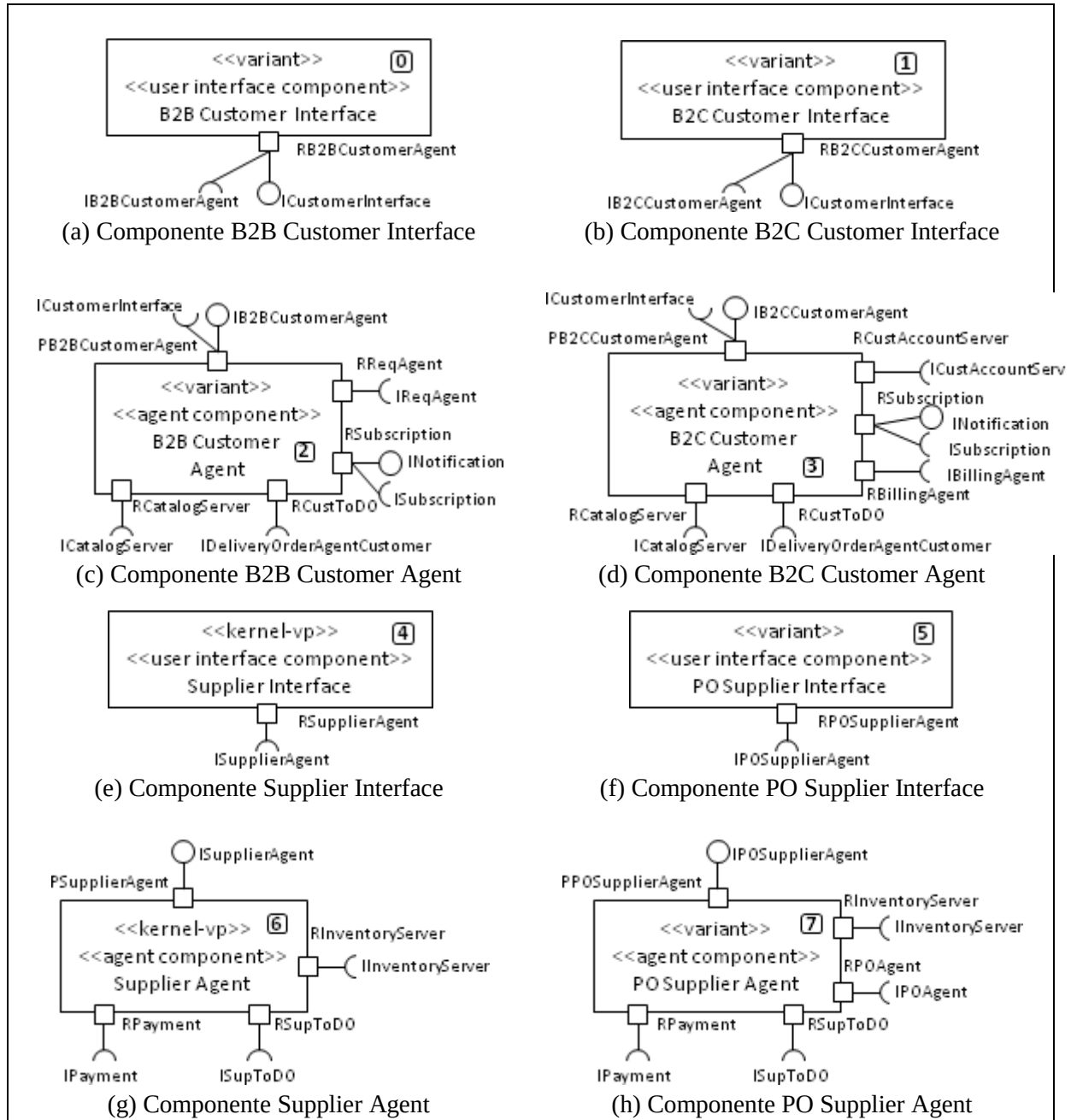


Figura 12. Visão dos componentes variantes

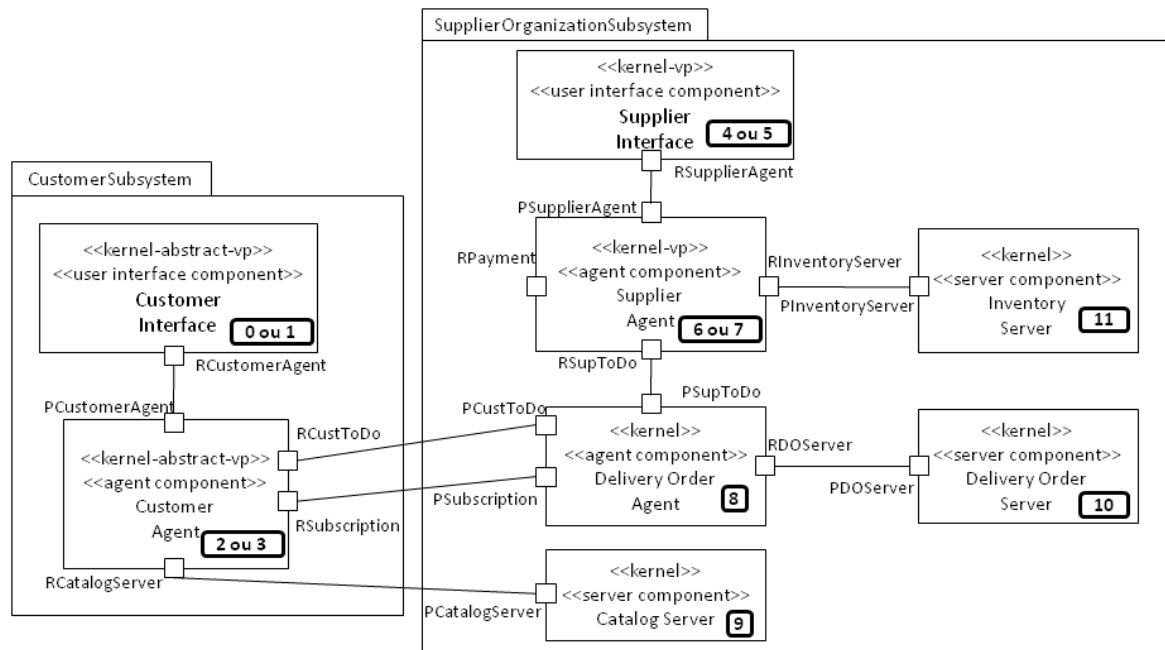


Figura 13. Arquitetura de domínio da LPS E-Commerce

A partir desta arquitetura foram definidos os módulos de *software*, segundo a fase de análise arquitetural do *framework* de alocações [Pereira, 2011]. Foram identificados três módulos distintos:

- Módulo M0 – composto pelos componentes c4, c6, c8, c9, c10, c11;
- Módulo M1 – composto pelos componentes c1, c3, c5, c7;
- Módulo M2 – composto pelos componentes c0, c2.

Desse modo, as tabelas de implementação foram definidas como seguem nas tabelas Tabela 9, Tabela 10 e Tabela 11:

Tabela 9. Tabela de implementação de M0 (E-commerce – domínio).

Baixo	Médio	Alto
Eclipse	Java	Agentes de Software
Svn	JSF	
SQL	Hibernate	
	Scrum	
	Sockets	

Tabela 10. Tabela de implementação de M1 (E-commerce – domínio).

Baixo	Médio	Alto
Eclipse	Java	Agentes de Software
Svn	Scrum	JSF

Tabela 11. Tabela de implementação de M2 (E-commerce – domínio).

Baixo	Médio	Alto
Eclipse	Java	
Svn	JSF	
	Scrum	
	Agentes de Software	

Vale lembrar que, como o projeto E-Commerce é um projeto hipotético desenvolvido com fins de estudo arquitetural apresentado em [Gomaa 2004], o mesmo não define tecnologias para a implementação do projeto. Contudo, a partir da natureza dos componentes e de alguns comentários realizados pelo autor do projeto E-Commerce, foi possível definir o conjunto de tecnologias a ser utilizadas.

Como não houve mudanças na estrutura dos termos *fuzzy* utilizados nem das funções de pertinência, utilizando, portanto, as configurações padrões definidas no Capítulo 4, a etapa de *Descrição dos módulos* é concluída para este projeto com a definição das tabelas Tabela 9, Tabela 10 e Tabela 11.

A partir deste ponto, os estudos para cada conjunto de equipes (A e B) será dividido em duas seções diferentes para um melhor entendimento do leitor.

5.2.1.1 Estudo com Equipes do Conjunto A

Com as tecnologias que devem ser implementadas já definidas, agora é possível definir o conhecimento de cada equipe para estas tecnologias. Para a etapa de *Descrição técnica das equipes*, as equipes foram identificadas através dos questionários e foi calculado para cada equipe o valor *crisp* de conhecimento acerca de cada tecnologia descrita em todas as tabelas de implementação. Para este cálculo foram utilizadas as Equações 1, 2 e 3. O resultado do cálculo do valor *crisp* pode ser observado na Tabela 12. Perceba que o número entre parênteses logo após o nome da equipe representa a quantidade de membros desta

equipe. Na Tabela 13 é apresentado o valor *crisp* calculado representando a tabela de implementação de cada módulo, conforme valores definidos na Figura 6.

Tabela 12. Valores *Crisp* das habilidades técnica das equipes (E-commerce - domínio).

EQUIPES	EQUIPE 1 (9)	EQUIPE 2 (5)	EQUIPE 3 (18)	EQUIPE 4 (8)	EQUIPE 5 (9)	EQUIPE 6 (2)	EQUIPE 7 (5)
JAVA	0,7107	0,6438	0,7064	0,6489	0,6811	0,7378	0,7286
JSF	0,7455	0,7140	0,7542	0,7140	0,6300	0,9368	0,8722
SQL	0,7662	0,7137	0,7996	0,7932	0,7670	0,8722	0,7793
HIBERNATE	0,8726	0,7603	0,7838	0,7958	0,8641	0,9047	0,8255
ECLIPSE	0,8898	0,8747	0,8986	0,8833	0,8925	0,9368	0,9594
AGENTES DE SOFTWARE	0,5833	0,0000	0,7912	0,0000	0,5833	0,6675	0,0000
SVN	0,8523	0,8073	0,8793	0,8363	0,9089	0,9047	0,7722
SCRUM	0,6846	0,4667	0,6016	0,6016	0,5030	0,5030	0,5646
SOCKETS	0,7140	0,6981	0,8219	0,8084	0,8499	0,5833	0,5833
REPUTACAO	0,3426	0,9197	0,5000	0,1787	0,4116	0,5189	0,4715

Tabela 13. Valores *Crisp* da complexidade esperada das tecnologias (E-commerce – domínio).

MODULOS	M0	M1	M2
JAVA	0,5000	0,5000	0,5000
JSF	0,5000	0,8000	0,5000
SQL	0,2000		
HIBERNATE	0,5000		
ECLIPSE	0,2000	0,2000	0,2000
AGENTES DE SOFTWARE	0,8000	0,8000	0,5000
SVN	0,2000	0,2000	0,2000
SCRUM	0,5000	0,5000	0,5000
SOCKETS	0,5000		

Utilizando a **política de qualificação equivalente** (Tabela 3) foi obtido o resultado apresentado na Tabela 14.

Tabela 14. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto A).

RESULTADO (SEM REPUTACAO)					
M0		M1		M2	
EQUIPE 1	0,7046	EQUIPE 3	0,7477	EQUIPE 5	0,7843
EQUIPE 5	0,7001	EQUIPE 6	0,7132	EQUIPE 1	0,7513
EQUIPE 3	0,6973	EQUIPE 1	0,6984	EQUIPE 6	0,7132
EQUIPE 2	0,6782	EQUIPE 5	0,6863	EQUIPE 3	0,7055
EQUIPE 6	0,6716	EQUIPE 7	0,6375	EQUIPE 4	0,6668
EQUIPE 7	0,6303	EQUIPE 4	0,5947	EQUIPE 2	0,6469
EQUIPE 4	0,6192	EQUIPE 2	0,5747	EQUIPE 7	0,6277

Note que na Tabela 14 os resultados não levam ainda em consideração o fator reputação nem o ponto de corte, contudo para um melhor entendimento do leitor as equipes que não atingiram a pontuação necessária para o ponto de corte estão apresentadas com o valor em vermelho.

Utilizando então o cálculo de reputação (Equação 6) e o ponto de corte padrão definido como 0.7 é feito a ordenação dos resultados e obtido o artefato final *mapeamento equipes × módulos* apresentado na Tabela 15.

Tabela 15. Versão final do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto A).

RESULTADO FINAL					
M0		M1		M2	
EQUIPE 2	0,8097	EQUIPE 3	0,7477	EQUIPE 2	0,7891
EQUIPE 3	0,6973	EQUIPE 2	0,7400	EQUIPE 5	0,7444
EQUIPE 6	0,6814	EQUIPE 6	0,7221	EQUIPE 6	0,7221
EQUIPE 5	0,6485	EQUIPE 5	0,6330	EQUIPE 3	0,7055
EQUIPE 7	0,6129	EQUIPE 7	0,6203	EQUIPE 1	0,6588
EQUIPE 1	0,5999	EQUIPE 1	0,5922	EQUIPE 7	0,6102
EQUIPE 4	0,2616	EQUIPE 4	0,2336	EQUIPE 4	0,3218

Analisando apenas o conhecimento das equipes, sem levar em consideração o valor calculado com reputação das mesmas, percebe-se que as equipes 1 e 5 conseguiram as maiores pontuações para o módulo M0, as equipes 3 e 6 para o módulo M1 e todas essas quatro equipes (1, 3, 5 e 6) para o módulo M2 (Tabela 14). Para entender o porquê destas equipes serem selecionadas, será apresentada uma comparação entre a equipe 5 e a equipe 7, equipe melhor qualificada e pior qualificada, respectivamente, para o módulo M2 (Tabela 14). Note que a política selecionada, **política de qualificação equivalente**, tende a selecionar

equipes que estão próximas dos níveis de complexidade apresentados na tabela de implementação. O módulo M2 possui em, sua maioria, tecnologias que exigem complexidade ‘média’, dessa forma aquelas equipes que estão em ambos os extremos (conhecimentos mais alto e mais baixo) são avaliadas de forma semelhante, ou seja, não possuem uma avaliação tão boa. Já equipes cujo valor de conhecimento está mais perto do valor esperado, seja para mais ou para menos, terão uma melhor qualificação. Assim note que para as tecnologias JAVA, JSF, AGENTES DE SOFTWARE e SCRUM é esperado que as equipes possuam um conhecimento ‘médio’ em cada tecnologia, já para as tecnologias ECLIPSE e SVN é esperado que as equipes possuam um conhecimento ‘baixo’ em cada tecnologia. Ao se comparar os valores apresentados para JAVA, JSF, AGENTES DE SOFTWARE e SCRUM (Tabela 12) nota-se que a Equipe 5 apresenta os valores 0.6811, 0.7140, 0.5833 e 0.5030 enquanto a Equipe 7 apresenta os valores 0.7286, 0.8722, 0.0000 e 0.5833, respectivamente. Note que considerando o valor ‘médio’ esperado por cada uma dessas tecnologias como o valor aproximado de 0.5, é possível verificar que para todas essas tecnologias a Equipe 5 chega mais perto do valor 0.5 do que a Equipe 7. Já para as tecnologias ECLIPSE e SVN nota-se que a Equipe 5 apresenta os valores 0.8925 e 0.9089 enquanto a Equipe 7 apresenta os valores 0.9594 e 0.7722, respectivamente. Note que considerando o valor ‘baixo’ esperado por cada uma dessas tecnologias como o valor aproximado de 0.2 é possível verificar que para ambas as tecnologias as duas equipes apresentam um valor bem elevado, ganhando, portanto, pouca ou nenhuma pontuação segundo a política. Assim, pela ótica de cada tecnologia isolada, em média é possível notar que realmente a Equipe 5 atende melhor que a Equipe 7 às necessidades exigidas pela **política de qualificação equivalente** e portanto deve realmente possuir uma melhor pontuação, assim como foi calculado pela abordagem de seleção de equipes proposta.

Agora, analisando o conhecimento das equipes junto com a reputação é possível perceber que equipes que tinham ficado fora do ponto de corte adquiriam pontos suficientes para ficar acima do valor de corte, bem como equipes que estavam acima do ponto de corte perderam pontos que as levaram a ficar abaixo do ponto de corte. Tomando como exemplo a Equipe 2, nota-se que para qualquer dos módulos ela estava abaixo do ponto de corte antes de considerar a reputação (Tabela 14), mas obteve pontos suficientes para ficar acima deste ponto, após o cálculo com a reputação (Tabela 15). Esse comportamento é esperado já que mesmo quando calculado sem reputação, a adequabilidade da equipe fica, para cada módulo, próximo do valor de ponto de corte. Como essa equipe possui o valor de reputação 0.9197, é

esperado que a mesma ganhasse pontos consideráveis na reavaliação da adequabilidade com a reputação, pois a pontuação de reputação desta equipe é alta (máximo sendo 1), o que significa que na prática essa equipe foi muito bem avaliada pelo gerente de projetos em projetos passados. Assim, a equipe foi avaliada positivamente no passado segundo o desenvolvimento realizado na prática, portanto é justo que esta equipe ganhe pontos, pois é notório o bom desempenho da mesma.

Analisando agora a Equipe 4, é possível notar que a mesma, depois da reavaliação com adequabilidade (Tabela 15), passou a ser a equipe menos adequada para cada módulo. Esse comportamento também é esperado visto que em geral a adequabilidade calculada, sem reputação, ficou abaixo do ponto de corte. Como o valor de reputação dessa equipe é 0.1787, próximo ao valor mínimo 0, significa que a equipe foi avaliada negativamente no passado segundo o desenvolvimento realizado na prática, portanto é justo que esta equipe perca pontos, pois é notório o mau desempenho da mesma.

Por fim a Equipe 5, que a priori foi selecionada para os módulos M0 e M2 (Tabela 14), após a reavaliação com reputação (Tabela 15), a mesma não mais atingiu o valor mínimo de ponto de corte para o módulo M0, contudo ficou acima do ponto de corte para o módulo M2. É possível verificar esse comportamento, pois a reputação calculada para a Equipe 5 foi de 0.4116, ou seja, próximo ao valor neutro 0.5. Contudo, como foi abaixo de 0.5, é possível que essa equipe tenha tido avaliações neutras e algumas positivas e negativas com leve predominância a avaliações negativas. Assim, por ser considerada pelo gerente de projetos apenas como um pouco abaixo da média, deve-se reavaliar esta equipe penalizando um pouco apenas o valor de adequabilidade calculado para a mesma. Devido a este fato, como a Equipe 5 obteve o valor de adequabilidade inicial 0,7001 para o módulo M0 (Tabela 14), a mesma já se encontrava no limite do ponto de corte. Como a reputação dela é abaixo do valor neutro, a mesma é levemente penalizada e passa a ficar abaixo do ponto de corte. Por outro lado, para o módulo M2, a Equipe 5 obteve o valor de adequabilidade inicial 0.7843 (Tabela 14), quando o valor foi recalculado, considerando a reputação, sofreu uma leve penalização atingindo o valor 0.7444 (Tabela 15). Portanto, é justo que a mesma ainda continue sendo uma opção válida para esse módulo.

Agora utilizando a **política de equipes mais hábeis** (Tabela 4) se obtém os resultados mostrados nas Tabela 16 (sem cálculo de reputação) e Tabela 17 (com cálculo de reputação).

Tabela 16. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - equipes mais hábeis - Conjunto A).

RESULTADO (SEM REPUTACAO)					
M0		M1		M2	
EQUIPE 3	0,9079	EQUIPE 3	0,8730	EQUIPE 3	0,8730
EQUIPE 6	0,8803	EQUIPE 6	0,8694	EQUIPE 6	0,8694
EQUIPE 5	0,8561	EQUIPE 1	0,8286	EQUIPE 1	0,8286
EQUIPE 1	0,8439	EQUIPE 5	0,8039	EQUIPE 5	0,8039
EQUIPE 4	0,8028	EQUIPE 7	0,7503	EQUIPE 7	0,7503
EQUIPE 7	0,7919	EQUIPE 4	0,7123	EQUIPE 4	0,7123
EQUIPE 2	0,7179	EQUIPE 2	0,6915	EQUIPE 2	0,6915

Tabela 17. Versão final do Mapeamento equipes × módulos (E-commerce – domínio – equipes mais hábeis - Conjunto A).

RESULTADO FINAL					
M0		M1		M2	
EQUIPE 3	0,9079	EQUIPE 6	0,8738	EQUIPE 6	0,8738
EQUIPE 6	0,8844	EQUIPE 3	0,8730	EQUIPE 3	0,8730
EQUIPE 2	0,8351	EQUIPE 2	0,8183	EQUIPE 2	0,8183
EQUIPE 5	0,8280	EQUIPE 5	0,7671	EQUIPE 5	0,7671
EQUIPE 7	0,7808	EQUIPE 1	0,7600	EQUIPE 1	0,7600
EQUIPE 1	0,7806	EQUIPE 7	0,7374	EQUIPE 7	0,7374
EQUIPE 4	0,5408	EQUIPE 4	0,3871	EQUIPE 4	0,3871

Seguindo a política de equipes hábeis era esperado que a Equipe 3 fosse a melhor colocada, já que em média ela é a que possui as melhores pontuações para cada tecnologia apresentada. Como descrito no Capítulo 4, essa política tende a escolher as equipes mais hábeis e ao mesmo tempo distancia as demais fazendo com que sejam eliminadas pelo ponto de corte, permanecendo no final apenas aquelas que realmente possuem níveis de conhecimento mais altos. Note que as equipes 2, 4 e 7 apresentam o valor de conhecimento 0.0000 para a tecnologia AGENTES DE SOFTWARE (Tabela 12), o que as faz baixar o valor médio de conhecimento necessário para implementação do módulo. Devido a este fato, pode ser evidenciado que essas três equipes obtiveram as notas mais baixas de adequabilidade para essa política em específico. Note que, comparando os valores sem cálculo de reputação, obtidos utilizando a **política de qualificação equivalente** (Tabela 3) e a **política de equipes mais hábeis** (Tabela 4), para o módulo M0, a equipe melhor qualificada, para a primeira política, foi Equipe 1 enquanto para a segunda política foi a Equipe 3. Note também que a única diferença entre esses dois estudos foi a política.

Assim, como já explicado para a **política de qualificação equivalente**, a Equipe 1 de fato se tornou a mais adequada devido ao fato dela apresentar os conhecimentos mais próximos aos conhecimentos exigidos para implementação no módulo M0. Já no caso da **política de equipes mais hábeis**, realmente, a Equipe 3 foi a melhor qualificada devido ao fato dela ser a que apresenta em média o maior valor de conhecimento para cada tecnologia apresentada. Analisando então a diferença dos resultados, variando apenas a política de seleção, vê-se que a abordagem reproduz o resultado esperado, calculando o valor de adequabilidade e ordem de seleção das equipes baseada na política específica de seleção configurada pelo gerente de projeto.

Analisando agora a reputação das equipes, pode-se perceber que novamente a Equipe 2 passa de um valor abaixo do valor de ponto de corte para um valor acima, se tornando uma equipe adequada para a implementação dos módulos M1 e M2 (Tabela 17). Ainda assim, note que, mesmo a Equipe 2 possuindo o maior valor de reputação 0.9197 (Tabela 12), isso não significa necessariamente que ela sempre será a opção mais adequada da lista de equipes adequadas, ou seja, o cálculo que leva em consideração a reputação não despreza o valor de adequabilidade calculado, mas agrega valor a esse cálculo.

5.2.1.2 Estudo com Equipes do Conjunto B

O resultado do cálculo do valor *crisp* para o conjunto B de equipes pode ser observado na Tabela 18. Novamente, o número entre parênteses logo após o nome da equipe representa a quantidade de membros desta equipe. O conjunto de tecnologias e módulos continua o mesmo utilizado para o conjunto de equipes A apresentado na Tabela 13.

Tabela 18. Valores *Crisp* das habilidades técnica das equipes (E-commerce – domínio – Conjunto B).

EQUIPES	EQUIPE 1 (3)	EQUIPE 2 (3)	EQUIPE 3 (3)	EQUIPE 4 (5)	EQUIPE 5 (5)	EQUIPE 6 (5)	EQUIPE 7 (8)	EQUIPE 8 (8)
JAVA	0,6085	0,5694	0,4890	0,5554	0,4349	0,5941	0,4396	0,5838
JSF	0,5814	0,5767	0,4348	0,4791	0,4360	0,6197	0,5177	0,6255
SQL	0,7008	0,5429	0,5185	0,7631	0,4906	0,5986	0,6751	0,6484
HIBERNATE	0,6286	0,6753	0,6002	0,5629	0,5308	0,6615	0,5360	0,6372
ECLIPSE	0,7796	0,7452	0,6272	0,7152	0,6848	0,7694	0,6337	0,7642
AGENTES DE SOFTWARE	0,4564	0,2250	0,2102	0,3267	0,3267	0,2823	0,3267	0,5246
SVN	0,8291	0,6286	0,6740	0,6744	0,5800	0,7248	0,6576	0,5925
SCRUM	0,4212	0,3746	0,5368	0,4548	0,4212	0,5901	0,6999	0,5026
SOCKETS	0,6659	0,4348	0,4126	0,6834	0,6383	0,4695	0,6269	0,4778
REPUTACAO	0,3426	0,9197	0,5000	0,1787	0,4116	0,5189	0,4715	0,7251

EQUIPES	EQUIPE 9 (8)	EQUIPE 10 (10)	EQUIPE 11 (10)	EQUIPE 12 (10)	EQUIPE 13 (15)	EQUIPE 14 (15)	EQUIPE 15 (15)
JAVA	0,5231	0,3913	0,5016	0,5659	0,4962	0,4783	0,5677
JSF	0,4323	0,5406	0,6066	0,6393	0,5444	0,6456	0,6397
SQL	0,5451	0,5899	0,6238	0,7117	0,5483	0,6515	0,7045
HIBERNATE	0,5622	0,4508	0,5875	0,7445	0,5945	0,6767	0,6567
ECLIPSE	0,6894	0,6713	0,6916	0,7537	0,6447	0,8115	0,7470
AGENTES DE SOFTWARE	0,3633	0,3699	0,4246	0,3969	0,3585	0,4759	0,4904
SVN	0,5996	0,5188	0,6651	0,7074	0,6868	0,7063	0,7162
SCRUM	0,4972	0,4454	0,4241	0,4738	0,4602	0,4736	0,4455
SOCKETS	0,5216	0,4858	0,6357	0,5631	0,4946	0,5417	0,6774
REPUTACAO	0,9567	0,4691	0,0610	0,3892	0,9738	0,0588	0,3556

Para o conjunto B de equipes, para efeitos de comparação de resultados, serão analisados os resultados utilizando as mesmas políticas utilizadas para o conjunto A (**política de qualificação equivalente** e **política de equipes mais hábeis**).

Ao utilizar a **política de qualificação equivalente** (Tabela 3) as equipes são selecionadas, qualificadas e ordenadas como apresentado na Tabela 19. Logo após a qualificação é realizado o cálculo de adequabilidade com reputação e utilizado o ponto de corte 0.7 obtendo os resultados apresentados na Tabela 20.

Tabela 19. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto B).

RESULTADO (SEM REPUTACAO)					
M0		M1		M2	
EQUIPE 8	0,8353	EQUIPE 8	0,7673	EQUIPE 8	0,8653
EQUIPE 15	0,8168	EQUIPE 15	0,7286	EQUIPE 13	0,8271
EQUIPE 11	0,8153	EQUIPE 11	0,7230	EQUIPE 15	0,8267
EQUIPE 13	0,8101	EQUIPE 13	0,7192	EQUIPE 11	0,8257
EQUIPE 7	0,8089	EQUIPE 7	0,7133	EQUIPE 7	0,8211
EQUIPE 9	0,8054	EQUIPE 9	0,7044	EQUIPE 9	0,8160
EQUIPE 14	0,7960	EQUIPE 3	0,7002	EQUIPE 3	0,8114
EQUIPE 6	0,7806	EQUIPE 4	0,6952	EQUIPE 4	0,8034
EQUIPE 3	0,7736	EQUIPE 14	0,6940	EQUIPE 12	0,7998
EQUIPE 4	0,7730	EQUIPE 12	0,6919	EQUIPE 6	0,7973
EQUIPE 12	0,7723	EQUIPE 6	0,6895	EQUIPE 14	0,7926
EQUIPE 5	0,7620	EQUIPE 2	0,6691	EQUIPE 2	0,7769
EQUIPE 2	0,7568	EQUIPE 10	0,6667	EQUIPE 10	0,7745
EQUIPE 1	0,7558	EQUIPE 5	0,6353	EQUIPE 5	0,7464
EQUIPE 10	0,7532	EQUIPE 1	0,6343	EQUIPE 1	0,7339

Tabela 20. Versão final do Mapeamento equipes × módulos (E-commerce – domínio - política de qualificação equivalente - Conjunto B).

RESULTADO FINAL					
M0		M1		M2	
EQUIPE 13	0,8975	EQUIPE 13	0,8443	EQUIPE 13	0,9071
EQUIPE 9	0,8931	EQUIPE 8	0,8330	EQUIPE 8	0,9050
EQUIPE 8	0,8833	EQUIPE 9	0,8327	EQUIPE 9	0,8992
EQUIPE 2	0,8595	EQUIPE 2	0,8037	EQUIPE 2	0,8718
EQUIPE 7	0,7986	EQUIPE 3	0,7002	EQUIPE 7	0,8114
EQUIPE 6	0,7877	EQUIPE 6	0,6989	EQUIPE 3	0,8114
EQUIPE 3	0,7736	EQUIPE 7	0,6989	EQUIPE 6	0,8039
EQUIPE 15	0,7524	EQUIPE 10	0,6491	EQUIPE 15	0,7652
EQUIPE 10	0,7393	EQUIPE 15	0,6407	EQUIPE 10	0,7616
EQUIPE 5	0,7188	EQUIPE 12	0,6231	EQUIPE 12	0,7505
EQUIPE 12	0,7176	EQUIPE 5	0,5764	EQUIPE 5	0,7010
EQUIPE 1	0,6646	EQUIPE 1	0,5146	EQUIPE 1	0,6366
EQUIPE 4	0,4865	EQUIPE 4	0,3615	EQUIPE 4	0,5420
EQUIPE 11	0,1876	EQUIPE 11	0,0700	EQUIPE 11	0,2081
EQUIPE 14	0,1437	EQUIPE 14	0,0448	EQUIPE 14	0,1385

Note que as equipes 1 e 2 obtiveram para o módulo M0, sem considerar o valor de reputação (Tabela 19), adequabilidades tão parecidas que se pode dizer que ambas possuem a mesma adequabilidade (0.7558 e 0.7568 respectivamente). Contudo, quando se analisa os valores de conhecimento de cada tecnologia de ambas as equipes (Tabela 18), verifica-se que

para algumas tecnologias esses valores não são tão parecidos e assim não se justificaria a adequabilidade de ambas terem sido calculadas com valores tão próximos. Todavia, verificando uma a uma, vê-se que para as tecnologias SQL, ECLIPSE e SVN espera-se que as equipes possuam um conhecimento ‘baixo’. Para estas tecnologias a Equipe 1 apresenta os valores 0.7008, 0.7796 e 0.8291 e a Equipe 2 os valores 0.5429, 0.7492 e 0.6286 respectivamente (Tabela 19). Note que para estas tecnologias, e para a política selecionada, a Equipe 2 se sai melhor que a Equipe 1 pois apresenta valores mais próximos do valor ‘baixo’ esperado para implementação dos módulos. Já para as tecnologias JAVA, JSF, HIBERNATE, SCRUM e SOCKETS, espera-se que as equipes possuam um conhecimento ‘médio’. Para estas tecnologias a Equipe 1 apresenta os valores 0.6085, 0.5814, 0.6286, 0.4212 e 0.6659 e a Equipe 2 os valores 0.5694, 0.5767, 0.6753, 0.3746 e 0.4348 (Tabela 19). Para estas tecnologias, a Equipe 1 se sai melhor que a Equipe 2, pois para as tecnologias apresentadas a equipe apresenta em média os valores mais próximos do valor ‘médio’, visto que o conjunto *fuzzy* para o termo ‘médio’ vai de 0.4 até 0.7 e para o termo ‘baixo’ vai de 0.1 a 0.5 (Figura 11). Assim se observa que a Equipe 2 apresenta um valor (para tecnologia SCRUM) que se encontra totalmente dentro do conjunto ‘baixo’, enquanto as demais estão entre ‘baixo’ e ‘médio’ ou totalmente em ‘médio’, fazendo com que a Equipe 1 possua uma pontuação maior que a Equipe 2. Por fim, para a tecnologia AGENTES DE SOFTWARE, é esperado que as equipes possuam um nível de conhecimento ‘alto’. Para esta tecnologia a Equipe 1 apresenta o valor 0.4564 e a Equipe 2 o valor 0.2250 (Tabela 19). Percebe-se então que para esta tecnologia a Equipe 1 apresenta um valor mais adequado que a Equipe 2.

Por essa análise, percebe-se que para algumas tecnologias uma equipe é mais adequada e para outras não, mas em média ambas possuem semelhante proporção de adequabilidade considerando tecnologia a tecnologia, o que justifica o valor de adequabilidade final de ambas ser tão próximo. Já quando se considera a reputação de ambas (Tabela 20), a Equipe 1 passa a ser penalizada enquanto a Equipe 2 passa a se bonificar, já que os valores de reputação são respectivamente 0.3426 e 0.9197. Nesse caso, a reputação serviu nitidamente como fator decisivo, caso fosse considerado apenas essas duas equipes. Como ambas obtiveram adequabilidades semelhantes, a reputação das equipes fez a diferença na adequabilidade final, como de fato deve ser, pois pelo valor de reputação percebe-se que a Equipe 1 obteve em seu histórico de implementações de módulos de *software* um desempenho bem mais baixo que o valor apresentado pela Equipe 2, e não obstante a Equipe

2 apresentou um ótimo histórico de implementações. Por isso é mais que justo que na avaliação final a Equipe 2 seja considerada mais adequada que a Equipe 1.

Agora ao se utilizar a **política de equipes mais hábeis** (Tabela 4) se obtém os resultados mostrados nas Tabela 21 e Tabela 22.

Tabela 21. Primeira versão do Mapeamento equipes × módulos (E-commerce – domínio - equipes mais hábeis - Conjunto B).

RESULTADO (SEM REPUTACAO)					
M0		M1		M2	
EQUIPE 1	0,7154	EQUIPE 8	0,7338	EQUIPE 8	0,7338
EQUIPE 8	0,7146	EQUIPE 1	0,7197	EQUIPE 1	0,7197
EQUIPE 14	0,7090	EQUIPE 14	0,7106	EQUIPE 14	0,7106
EQUIPE 15	0,7001	EQUIPE 15	0,6950	EQUIPE 15	0,6950
EQUIPE 12	0,6896	EQUIPE 6	0,6885	EQUIPE 6	0,6885
EQUIPE 6	0,6814	EQUIPE 12	0,6568	EQUIPE 12	0,6568
EQUIPE 4	0,6623	EQUIPE 11	0,6202	EQUIPE 11	0,6202
EQUIPE 11	0,6487	EQUIPE 13	0,6199	EQUIPE 13	0,6199
EQUIPE 13	0,6459	EQUIPE 4	0,6131	EQUIPE 4	0,6131
EQUIPE 7	0,6436	EQUIPE 7	0,6124	EQUIPE 7	0,6124
EQUIPE 9	0,6396	EQUIPE 2	0,6078	EQUIPE 2	0,6078
EQUIPE 2	0,6157	EQUIPE 9	0,6064	EQUIPE 9	0,6064
EQUIPE 3	0,6034	EQUIPE 3	0,6020	EQUIPE 3	0,6020
EQUIPE 5	0,5832	EQUIPE 10	0,5565	EQUIPE 10	0,5565
EQUIPE 10	0,5803	EQUIPE 5	0,5287	EQUIPE 5	0,5287

Tabela 22. Versão final do Mapeamento equipes × módulos (E-commerce – domínio – equipes mais hábeis - Conjunto B).

RESULTADO FINAL					
M0		M1		M2	
EQUIPE 13	0,7989	EQUIPE 8	0,8078	EQUIPE 8	0,8078
EQUIPE 8	0,7932	EQUIPE 13	0,7823	EQUIPE 13	0,7823
EQUIPE 9	0,7917	EQUIPE 9	0,7699	EQUIPE 9	0,7699
EQUIPE 2	0,7682	EQUIPE 2	0,7628	EQUIPE 2	0,7628
EQUIPE 6	0,6910	EQUIPE 6	0,6979	EQUIPE 6	0,6979
EQUIPE 7	0,6266	EQUIPE 1	0,6188	EQUIPE 1	0,6188
EQUIPE 12	0,6204	EQUIPE 3	0,6020	EQUIPE 3	0,6020
EQUIPE 1	0,6133	EQUIPE 15	0,5995	EQUIPE 15	0,5995
EQUIPE 15	0,6057	EQUIPE 7	0,5945	EQUIPE 7	0,5945
EQUIPE 3	0,6034	EQUIPE 12	0,5827	EQUIPE 12	0,5827
EQUIPE 10	0,5599	EQUIPE 10	0,5355	EQUIPE 10	0,5355
EQUIPE 5	0,5194	EQUIPE 5	0,4611	EQUIPE 5	0,4611
EQUIPE 4	0,3157	EQUIPE 4	0,2544	EQUIPE 4	0,2544
EQUIPE 14	0,0537	EQUIPE 14	0,0547	EQUIPE 14	0,0547
EQUIPE 11	0,0288	EQUIPE 11	0,0199	EQUIPE 11	0,0199

Note que, como apresentado na Tabela 22, a maioria das equipes não atingiu um valor de adequabilidade necessário para passar pelo ponto de corte, diferentemente do resultado apresentado para o conjunto A de equipes, como apresentado na Tabela 16. Isso se deve ao fato de que em média, se comparado os valores *crisp* de conhecimento apresentados pelas equipes para cada tecnologia, as equipes do conjunto A apresentam valores mais altos que os apresentados pelas equipes do conjunto B. Como a **política de equipes mais hábeis** tende a selecionar as equipes mais hábeis, ou seja, as que apresentam valores mais altos de conhecimento, é natural que o conjunto B de equipes possua mais equipes que não atingem o valor de adequabilidade mínimo necessário.

Ao analisar as equipes 1 e 2 novamente (Tabela 21), observa-se um comportamento oposto ao apresentado pelo resultado obtido ao se utilizar a **política de qualificação equivalente**, visto que ao utilizar a **política de equipes mais hábeis** a Equipe 1 apresenta, para qualquer módulo, valor de adequabilidade mais alto que o da Equipe 2. Isso é justificado ao observar que para qualquer tecnologia apresentada, com exceção da tecnologia HIBERNATE, a Equipe 1 possui um conhecimento maior que a Equipe 2 (Tabela 18). Desse modo, fica óbvio que a Equipe 1 possui um conhecimento maior do que a Equipe 2 no conjunto de tecnologias exigidas pelos módulos de *software*, assim tornando-a mais hábil, logo mais adequada pela regra da política.

É interessante notar que para a **política de equipes mais hábeis**, os valores apresentados de adequabilidades de cada equipe são iguais para os módulos M1 e M2, mas diferente para o módulo M0 (Tabela 21). Isso se dá, pois como a regra da política é analisar quem possui mais conhecimento, não importa qual o nível de adequabilidade exigida pelo módulo, verifica-se apenas quanto uma equipe apresenta. Assim, a única informação relevante do módulo é a tecnologia exigida, mas não seu nível. Isso pode ser observado novamente comparando-se os valores de adequabilidade calculados da Equipe 1 e Equipe 2 para os módulos M0, M1 e M2. Note que a Equipe 1 obteve o valor 0.7154 para o módulo M0 e 0.7197 para os módulos M1 e M2. Já a Equipe 2 obteve os valores 0.6157 para o módulo M1 e 0.6078 para os módulos M1 e M2. Verifica-se que a Equipe 1 atingiu um valor ligeiramente maior para M1 e M2 e menor para M0, já a Equipe 2 possui comportamento contrário, apresenta valor ligeiramente maior para M0 e menor para M1 e M2. Esse comportamento é explicado pelo fato do módulo M0 requerer a tecnologia HIBERNATE, a única tecnologia na qual a Equipe 2 é melhor que a Equipe 1 e, por isso, a Equipe 2 apresenta o valor de adequabilidade ligeiramente maior para esse módulo e a Equipe 1 ligeiramente menor. Em suma, para todos os módulos, a Equipe 1 apresenta maior adequabilidade, satisfazendo a regra explicitada na política de seleção.

Por fim, ainda analisando as equipes 1 e 2, verifica-se que, no primeiro resultado atingido de cálculo de adequabilidade (Tabela 21), a Equipe 1 se apresenta acima do ponto de corte enquanto a Equipe 2 não. Contudo, ao se calcular com a reputação (Tabela 22), o resultado se inverte, ou seja, a Equipe 2 passa a atingir um valor de adequabilidade maior que o ponto de corte enquanto a Equipe 1 passa a ficar abaixo. Isso também pode ser justificado devido ao fato da Equipe 2 apresentar um valor muito alto de reputação enquanto a Equipe 1 apresenta um valor muito baixo. Como no estudo de caso o valor de reputação foi criado aleatoriamente, os resultados podem parecer estranhos, pois muitas vezes pode apresentar essa inversão de valores. Contudo, num caso real, espera-se no caso geral que equipes que possuam valores altos de conhecimento não apresentem valores baixos de reputação, assim não apresentando essa mudança de valores. Isso não é regra, mas uma tendência já que quanto maior o conhecimento maior a tendência de se apresentar um melhor desempenho de implementação.

5.2.2 Segunda iteração (Engenharia de aplicação)

Na segunda iteração de desenvolvimento, 14 componentes foram adicionados, os quais em sua maioria são identificados na arquitetura das aplicações B2B e B2C apresentadas pela Figura 14 e pela Figura 15, respectivamente.

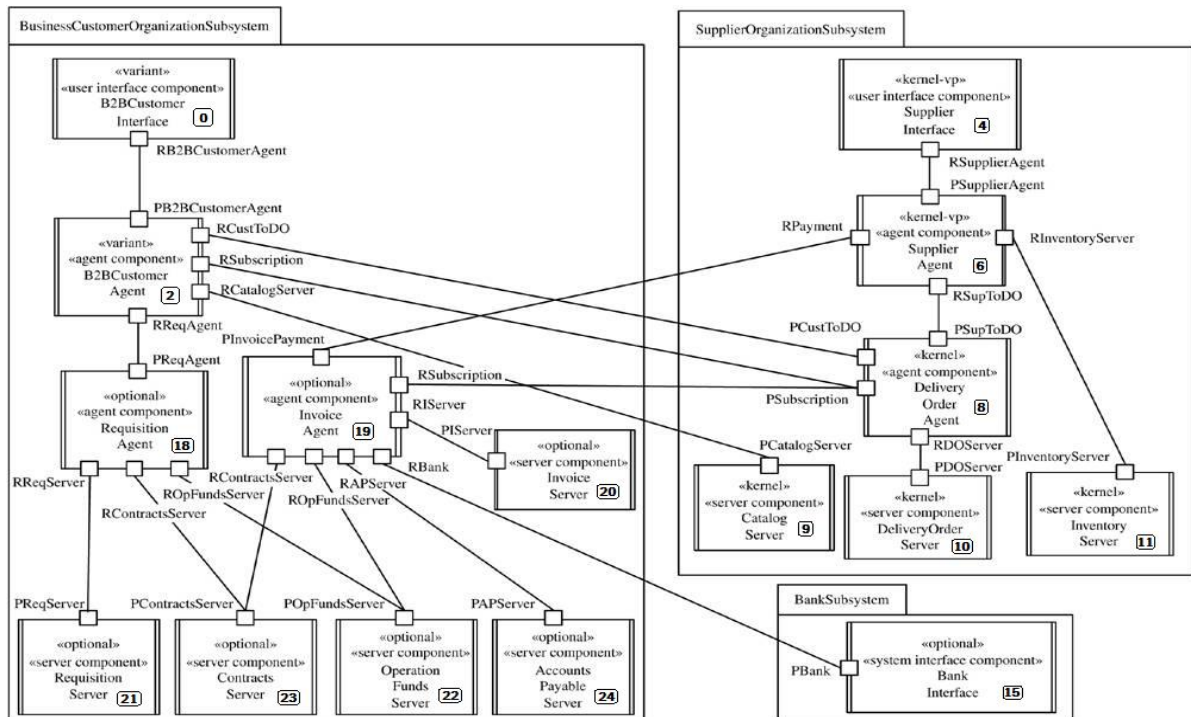


Figura 14. Arquitetura da aplicação B2B

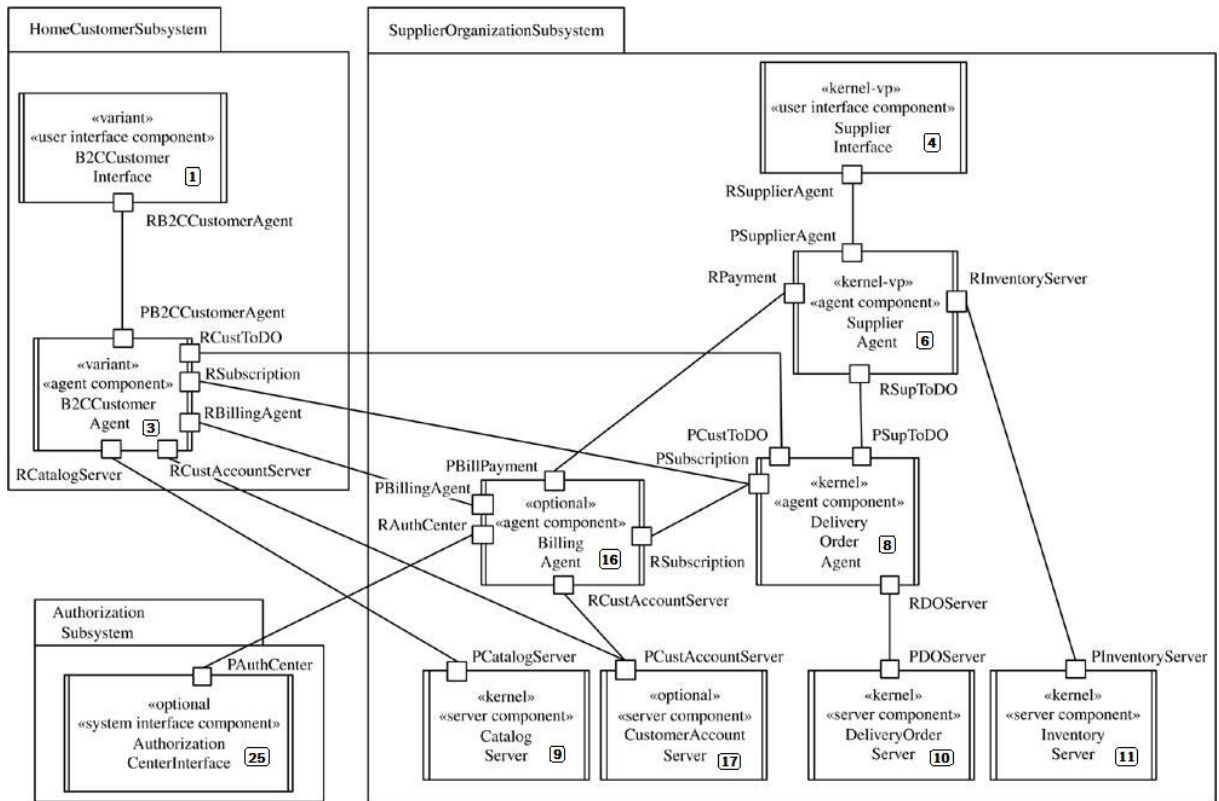


Figura 15. Arquitetura da aplicação B2C.

Os demais componentes alternativos são apresentados na Figura 16.

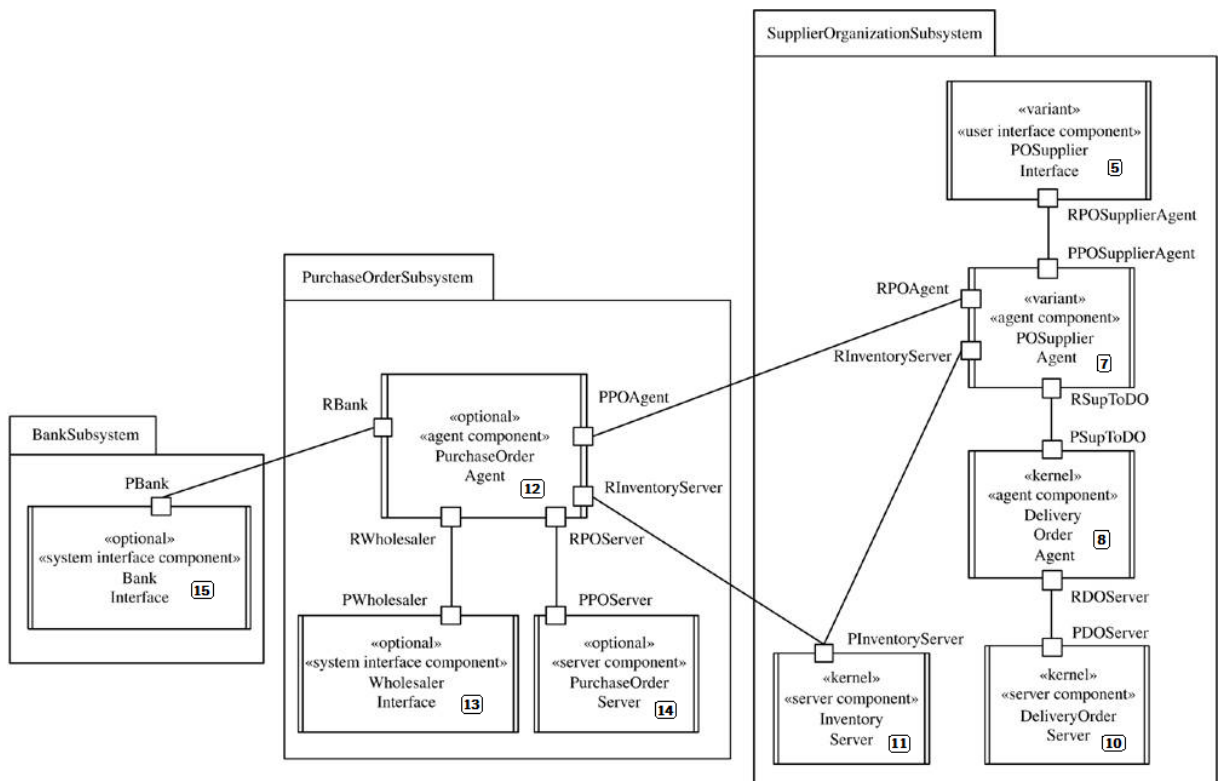


Figura 16. Componentes opcionais da LPS de E-Commerce.

A partir desta arquitetura foram definidos os módulos de *software*, segundo a fase de análise arquitetural do *framework* de alocações [Pereira, 2011]. Foram identificados quatro módulos distintos:

- Módulo M0 – composto pelos componentes [c4, c6, c8, c9, c10, c11];
- Módulo M1 – composto pelos componentes [c1, c3, c5, c7] + c16, c17, c25
- Módulo M2 – composto pelos componentes [c0, c2] + c18, c19, c20, c21, c22, c23, c24;
- Módulo M3 – composto pelos componentes c12, c13, c14, c15;

Os componentes em colchetes são aqueles que já foram implementados na iteração passada (engenharia de domínio). Na segunda iteração do projeto, novos componentes foram adicionados aos módulos existentes M1 e M2. Ao módulo M0 nenhum componente foi associado. Como ele foi definido na fase anterior, será admitido que este componente já está implementado e portanto não será considerado nessa iteração. Por fim, um novo módulo M3, não identificado na engenharia de domínio, foi criado.

Desse modo, as tabelas de implementação foram definidas como seguem na Tabela 23, Tabela 24 e Tabela 25:

Tabela 23. Tabela de implementação de M1 (E-commerce – aplicação).

Baixo	Médio	Alto
Eclipse	Java	Agentes de Software
Svn	Scrum	
	Transações Bancárias	

Tabela 24. Tabela de implementação de M2 (E-commerce – aplicação).

Baixo	Médio	Alto
Eclipse	Java	Agentes de Software
Svn	Scrum	
	Transações Bancárias	

Tabela 25. Tabela de implementação de M3 (E-commerce – aplicação).

Baixo	Médio	Alto
Eclipse	Java	Transações Bancárias
Svn	Scrum	
	Agentes de Software	

Os novos módulos M1 e M2 possuem componentes que desempenham funcionalidades semelhantes. Assim, foi decidido configurar as tabelas de implementação de ambas do mesmo modo. Desse modo tornou-se possível verificar se a abordagem de seleção realmente retornaria sempre as mesmas recomendações para ambos os módulos. Isso de fato foi evidenciado. O módulo M3 também possui componentes cujas funcionalidades não diferem muito das funcionalidades dos componentes dos módulos M1 e M2. Aproveitando-se disso, a tabela de implementação do módulo M3 foi feita de modo a ser semelhante às tabelas de implementação dos módulos M1 e M2, apenas duas tecnologias foram trocadas de lugar. Essa mudança pequena foi feita para averiguar o quanto estas pequenas diferenças alterariam as recomendações.

5.2.2.1 Estudo com Equipes do Conjunto A

Como algumas tecnologias foram retiradas e outras adicionadas em relação à engenharia de domínio, uma nova tabela com as habilidades técnicas *crisp* das equipes é apresentada na Tabela 26. Na Tabela 27 é apresentado o valor *crisp* calculado representando a tabela de implementação de cada módulo

Tabela 26. Valores Crisp das habilidades técnica das equipes (E-commerce - aplicação).

EQUIPES	EQUIPE 1 (9)	EQUIPE 2 (5)	EQUIPE 3 (18)	EQUIPE 4 (8)	EQUIPE 5 (9)	EQUIPE 6 (2)	EQUIPE 7 (5)
JAVA	0,7107	0,6438	0,7064	0,6489	0,6811	0,7378	0,7286
ECLIPSE	0,8898	0,8747	0,8986	0,8833	0,8925	0,9368	0,9594
AGENTES DE SOFTWARE	0,5833	0,0000	0,7912	0,0000	0,5833	0,6675	0,0000
SVN	0,8523	0,8073	0,8793	0,8363	0,9089	0,9047	0,7722
SCRUM	0,6846	0,4667	0,6016	0,6016	0,5030	0,5030	0,5646
TRANSACOES BANCARIAS	0,8393	0,0000	0,6675	0,5833	0,6981	0,5833	0,0000
REPUTACAO	0,3426	0,9197	0,5000	0,1787	0,4116	0,5189	0,4715

Tabela 27. Valores *Crisp* da complexidade esperada das tecnologias(E-commerce - aplicação).

MODULOS	M1	M2	M3
JAVA	0,5000	0,5000	0,5000
ECLIPSE	0,2000	0,2000	0,2000
AGENTES DE SOFTWARE	0,8000	0,8000	0,5000
SVN	0,2000	0,2000	0,2000
SCRUM	0,5000	0,5000	0,5000
TRANSACOES BANCARIAS	0,5000	0,5000	0,8000

Utilizando a **política de equipes mais hábeis** (Tabela 4) foram obtidos os resultados apresentados nas Tabela 28 (sem cálculo de reputação) e Tabela 29 (com cálculo de reputação).

Tabela 28. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de equipes mais hábeis – Conjunto A).

RESULTADO (SEM REPUTACAO)					
M1		M2		M3	
EQUIPE 1	0,8579	EQUIPE 1	0,8579	EQUIPE 1	0,8579
EQUIPE 3	0,8497	EQUIPE 3	0,8497	EQUIPE 3	0,8497
EQUIPE 6	0,8204	EQUIPE 6	0,8204	EQUIPE 6	0,8204
EQUIPE 5	0,8039	EQUIPE 5	0,8039	EQUIPE 5	0,8039
EQUIPE 4	0,7059	EQUIPE 4	0,7059	EQUIPE 4	0,7059
EQUIPE 7	0,6033	EQUIPE 7	0,6033	EQUIPE 7	0,6033
EQUIPE 2	0,5870	EQUIPE 2	0,5870	EQUIPE 2	0,5870

Tabela 29. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – equipes mais hábeis - Conjunto A).

RESULTADO FINAL					
M1		M2		M3	
EQUIPE 3	0,8497	EQUIPE 3	0,8497	EQUIPE 3	0,8497
EQUIPE 6	0,8263	EQUIPE 6	0,8263	EQUIPE 6	0,8263
EQUIPE 1	0,7996	EQUIPE 1	0,7996	EQUIPE 1	0,7996
EQUIPE 5	0,7671	EQUIPE 5	0,7671	EQUIPE 5	0,7671
EQUIPE 2	0,7485	EQUIPE 2	0,7485	EQUIPE 2	0,7485
EQUIPE 7	0,5851	EQUIPE 7	0,5851	EQUIPE 7	0,5851
EQUIPE 4	0,3774	EQUIPE 4	0,3774	EQUIPE 4	0,3774

Como esperado, os resultados obtidos para os módulos M1, M2 e M3 são os mesmos, já que todos possuem tabelas de implementação com as mesmas tecnologias e a política utilizada é de **equipes mais hábeis**. Note que o módulo M1 e M2 são derivados, respectivamente, dos módulos M1 e M2 criados na engenharia de domínio. Devido a este fato,

pode-se indagar que a mesma solução encontrada na engenharia de domínio servirá para o problema da engenharia de aplicação. Isso não é verdade, pois apesar de possuir tecnologias em comum e até por vezes com mesmo nível, a engenharia de aplicação deve ser encarada como um novo problema a ser desenvolvido, ou seja, nem sempre só porque uma equipe implementou certo módulo na engenharia de domínio implementará módulos derivados desse na engenharia de aplicação, cada caso é diferente. Caso a equipe possua as habilidades requeridas, ele se tornará apta, caso contrário não. Isso pode ser evidenciado ao se comparar os módulos M1 e M2 nas soluções encontradas para engenharia de domínio e aplicação. Contudo, percebe-se que as equipes 1, 3 e 6 ficam à frente das demais, pois elas realmente possuem em média os maiores níveis de conhecimento nas tecnologias exigidas pelos módulos citados. Todavia, a ordem de seleção sem reputação é diferente ao se comparar engenharia de domínio e aplicação, justamente pelo que foi comentado.

A política utilizada anteriormente seleciona as equipes que possuem mais conhecimento nas tecnologias apresentadas. Contudo, existe uma política que rege a seleção de forma quase inversa, essa política se chama **política de provisão de treinamento** (Tabela 6). O intuito dessa política é selecionar aquelas equipes que possuem um conhecimento mais baixo que o exigido pelo módulo, porém quanto mais perto do exigido mais pontos a equipe receberá. Desse modo se consegue prover treinamentos para as equipes nas tecnologias evidenciadas na tabela de implementação, pois se o conhecimento da equipe for muito mais baixo que o necessário para o treinamento, é possível que a equipe não aproveite o treinamento, pois lhe falta conhecimentos intermediários. Com esse conhecimento em mente, agora será realizada uma comparação dos resultados encontrados utilizando essa política, e, assim, é possível traçar as diferenças das soluções encontradas entre essa e a **política de equipes mais hábeis**.

As tabelas Tabela 30 e Tabela 31 apresentam, respectivamente, os resultados obtidos pela **política de provisão de treinamento**, sem e com o cálculo de reputação.

Tabela 30. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de provisão de treinamento – Conjunto A).

RESULTADO (SEM REPUTACAO)					
M1		M2		M3	
EQUIPE 5	0,7843	EQUIPE 5	0,7843	EQUIPE 5	0,7843
EQUIPE 3	0,7796	EQUIPE 3	0,7796	EQUIPE 1	0,7768
EQUIPE 6	0,7623	EQUIPE 6	0,7623	EQUIPE 6	0,7623
EQUIPE 1	0,7277	EQUIPE 1	0,7277	EQUIPE 4	0,7353
EQUIPE 4	0,6765	EQUIPE 4	0,6765	EQUIPE 3	0,7347
EQUIPE 7	0,6277	EQUIPE 7	0,6277	EQUIPE 7	0,6277
EQUIPE 2	0,6275	EQUIPE 2	0,6275	EQUIPE 2	0,6275

Tabela 31. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação - política de provisão de treinamento - Conjunto A).

RESULTADO FINAL					
M1		M2		M3	
EQUIPE 3	0,7796	EQUIPE 3	0,7796	EQUIPE 2	0,7762
EQUIPE 2	0,7762	EQUIPE 2	0,7762	EQUIPE 6	0,7698
EQUIPE 6	0,7698	EQUIPE 6	0,7698	EQUIPE 5	0,7444
EQUIPE 5	0,7444	EQUIPE 5	0,7444	EQUIPE 3	0,7347
EQUIPE 1	0,6289	EQUIPE 1	0,6289	EQUIPE 1	0,6917
EQUIPE 7	0,6102	EQUIPE 7	0,6102	EQUIPE 7	0,6102
EQUIPE 4	0,3350	EQUIPE 4	0,3350	EQUIPE 4	0,4230

Note que os valores de adequabilidade, para todas as equipes em todos os módulos, obtidos para a **política de provisão de treinamento**, são mais baixos que os obtidos na **política de equipes mais hábeis** (Tabela 30). Esse é o comportamento esperado já que se as equipes possuem conhecimentos altos tendem a precisar de menos treinamento. Ainda assim, as equipes 1, 3 e 6 continuam sendo boas opções para realizar um treinamento. Já a equipe 5, que não possui os maiores conhecimentos, passa a ser a mais adequada para treinamento. Isso pode ser verificado de acordo com a regra da política. A equipe 5 possui conhecimentos altos, mas mais baixo que os apresentados pelas equipes 1, 3 e 6 e mais altos que as demais equipes, ou seja, ela é a que possui os conhecimentos mais próximos aos conhecimentos necessários para a realização de um treinamento. As equipes 1, 3, 6 também possuem conhecimentos próximos, mas ligeiramente maiores que os conhecimentos exigidos. Já as equipes 2, 4 e 7 apresentam conhecimento 0.0000 para a tecnologia AGENTES DE SOFTWARE e a equipe 7 apresenta ainda conhecimento 0.0000 para o conhecimento de domínio TRANSAÇÕES BANCÁRIAS (Tabela 26). Como o conhecimento mínimo exigido para o treinamento nessa tecnologia e domínio são respectivamente os valores ‘alto’ e ‘médio’, isso faz com que estas

equipes não possuem o conhecimento mínimo para um treinamento, assim elas perdem pontos na adequabilidade para o treinamento e passam a não se configurar como boas opções para treinamento.

5.2.2.2 Estudo com Equipes do Conjunto B

O resultado do cálculo do valor *crisp* para o conjunto B de equipes pode ser observado na Tabela 32. Novamente, o número entre parênteses logo após o nome da equipe representa a quantidade de membros desta equipe. O conjunto de tecnologias e módulos continua o mesmo utilizado para o conjunto de equipes A apresentado na Tabela 27.

Tabela 32. Valores Crisp das habilidades técnica das equipes (E-commerce - aplicação)

EQUIPES	EQUIPE 1 (3)	EQUIPE 2 (3)	EQUIPE 3 (3)	EQUIPE 4 (5)	EQUIPE 5 (5)	EQUIPE 6 (5)	EQUIPE 7 (8)	EQUIPE 8 (8)
JAVA	0,6085	0,5694	0,4890	0,5554	0,4349	0,5941	0,4396	0,5838
ECLIPSE	0,7796	0,7452	0,6272	0,7152	0,6848	0,7694	0,6337	0,7642
AGENTES DE SOFTWARE	0,4564	0,2250	0,2102	0,3267	0,3267	0,2823	0,3267	0,5246
SVN	0,8291	0,6286	0,6740	0,6744	0,5800	0,7248	0,6576	0,5925
SCRUM	0,4212	0,3746	0,5368	0,4548	0,4212	0,5901	0,6999	0,5026
TRANSACOES BANCARIAS	0,4465	0,4205	0,1650	0,4903	0,5006	0,3401	0,4205	0,4261
REPUTACAO	0,3426	0,9197	0,5000	0,1787	0,4116	0,5189	0,4715	0,7251
EQUIPES	EQUIPE 9 (8)	EQUIPE 10 (10)	EQUIPE 11 (10)	EQUIPE 12 (10)	EQUIPE 13 (15)	EQUIPE 14 (15)	EQUIPE 15 (15)	
JAVA	0,5231	0,3913	0,5016	0,5659	0,4962	0,4783	0,5677	
ECLIPSE	0,6894	0,6713	0,6916	0,7537	0,6447	0,8115	0,7470	
AGENTES DE SOFTWARE	0,3633	0,3699	0,4246	0,3969	0,3585	0,4759	0,4904	
SVN	0,5996	0,5188	0,6651	0,7074	0,6868	0,7063	0,7162	
SCRUM	0,4972	0,4454	0,4241	0,4738	0,4602	0,4736	0,4455	
TRANSACOES BANCARIAS	0,4437	0,3772	0,4773	0,4099	0,4508	0,2650	0,4508	
REPUTACAO	0,9567	0,4691	0,0610	0,3892	0,9738	0,0588	0,3556	

Utilizando a **política de equipes mais hábeis** (Tabela 4) foram obtidos os resultados apresentados nas Tabela 33 (sem cálculo de reputação) e Tabela 34 (com cálculo de reputação).

Tabela 33. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de equipes mais hábeis – Conjunto B).

RESULTADO (SEM REPUTACAO)					
M1		M2		M3	
EQUIPE 8	0,6919	EQUIPE 8	0,6919	EQUIPE 8	0,6919
EQUIPE 1	0,6885	EQUIPE 1	0,6885	EQUIPE 1	0,6885
EQUIPE 15	0,6660	EQUIPE 15	0,6660	EQUIPE 15	0,6660
EQUIPE 14	0,6517	EQUIPE 14	0,6517	EQUIPE 14	0,6517
EQUIPE 6	0,6297	EQUIPE 6	0,6297	EQUIPE 6	0,6297
EQUIPE 4	0,6200	EQUIPE 4	0,6200	EQUIPE 4	0,6200
EQUIPE 9	0,6123	EQUIPE 9	0,6123	EQUIPE 9	0,6123
EQUIPE 11	0,6051	EQUIPE 11	0,6051	EQUIPE 11	0,6051
EQUIPE 12	0,6050	EQUIPE 12	0,6050	EQUIPE 12	0,6050
EQUIPE 13	0,5909	EQUIPE 13	0,5909	EQUIPE 13	0,5909
EQUIPE 3	0,5754	EQUIPE 3	0,5754	EQUIPE 3	0,5754
EQUIPE 7	0,5673	EQUIPE 7	0,5673	EQUIPE 7	0,5673
EQUIPE 5	0,5652	EQUIPE 5	0,5652	EQUIPE 5	0,5652
EQUIPE 2	0,5627	EQUIPE 2	0,5627	EQUIPE 2	0,5627
EQUIPE 10	0,4977	EQUIPE 10	0,4977	EQUIPE 10	0,4977

Tabela 34. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – equipes mais hábeis - Conjunto B)

RESULTADO FINAL					
M1		M2		M3	
EQUIPE 8	0,7757	EQUIPE 8	0,7757	EQUIPE 8	0,7757
EQUIPE 9	0,7739	EQUIPE 9	0,7739	EQUIPE 9	0,7739
EQUIPE 13	0,7633	EQUIPE 13	0,7633	EQUIPE 13	0,7633
EQUIPE 2	0,7315	EQUIPE 2	0,7315	EQUIPE 2	0,7315
EQUIPE 6	0,6404	EQUIPE 6	0,6404	EQUIPE 6	0,6404
EQUIPE 1	0,5801	EQUIPE 1	0,5801	EQUIPE 1	0,5801
EQUIPE 3	0,5754	EQUIPE 3	0,5754	EQUIPE 3	0,5754
EQUIPE 15	0,5646	EQUIPE 15	0,5646	EQUIPE 15	0,5646
EQUIPE 7	0,5482	EQUIPE 7	0,5482	EQUIPE 7	0,5482
EQUIPE 12	0,5244	EQUIPE 12	0,5244	EQUIPE 12	0,5244
EQUIPE 5	0,5001	EQUIPE 5	0,5001	EQUIPE 5	0,5001
EQUIPE 10	0,4753	EQUIPE 10	0,4753	EQUIPE 10	0,4753
EQUIPE 4	0,2625	EQUIPE 4	0,2625	EQUIPE 4	0,2625
EQUIPE 14	0,0262	EQUIPE 14	0,0262	EQUIPE 14	0,0262
EQUIPE 11	0,0163	EQUIPE 11	0,0163	EQUIPE 11	0,0163

É interessante notar que, no resultado de adequabilidade apresentado na Tabela 33, nenhuma equipe se apresentou com valor acima do ponto de corte. Novamente, vale lembrar que, para a política utilizada, as equipes mais adequadas são aquelas que possuem os maiores conhecimentos. Note que, em média, as equipes não apresentam valores altos de

conhecimento nas tecnologias exigidas pelos módulos, como pode ser visto na Tabela 32, e, por isso, não se tornam adequadas para implementação dos módulos, segundo a política selecionada. Como o resultado para a **política de equipes mais hábeis** apresentou equipes com valores baixos de adequabilidade, então se espera que utilizando a **política de provisão de treinamento** o resultado seja bem diferente, apresentando as equipes como adequadas para treinamento. De fato, essa previsão é confirmada, como se pode ver na Tabela 35 e Tabela 36. Isso indica como a seleção de equipes depende da política a ser utilizada, ou seja, de acordo com a especificidade de cada projeto é possível que o gerente de projetos configure políticas que melhor se adequem para o tipo de seleção que o mesmo pretende obter.

Tabela 35. Primeira versão do Mapeamento equipes × módulos (E-commerce – aplicação - política de provisão de treinamento – Conjunto B).

RESULTADO (SEM REPUTACAO)					
M1		M2		M3	
EQUIPE 8	0,8653	EQUIPE 8	0,8653	EQUIPE 5	0,9020
EQUIPE 11	0,8639	EQUIPE 11	0,8639	EQUIPE 11	0,8874
EQUIPE 15	0,8561	EQUIPE 15	0,8561	EQUIPE 4	0,8845
EQUIPE 10	0,8529	EQUIPE 10	0,8529	EQUIPE 13	0,8748
EQUIPE 7	0,8529	EQUIPE 7	0,8529	EQUIPE 9	0,8719
EQUIPE 13	0,8529	EQUIPE 13	0,8529	EQUIPE 7	0,8622
EQUIPE 9	0,8529	EQUIPE 9	0,8529	EQUIPE 10	0,8529
EQUIPE 5	0,8529	EQUIPE 5	0,8529	EQUIPE 3	0,8391
EQUIPE 4	0,8424	EQUIPE 4	0,8424	EQUIPE 15	0,8358
EQUIPE 3	0,8393	EQUIPE 3	0,8393	EQUIPE 2	0,8352
EQUIPE 2	0,8259	EQUIPE 2	0,8259	EQUIPE 8	0,8279
EQUIPE 14	0,8231	EQUIPE 14	0,8231	EQUIPE 12	0,8208
EQUIPE 12	0,8162	EQUIPE 12	0,8162	EQUIPE 6	0,7973
EQUIPE 6	0,7973	EQUIPE 6	0,7973	EQUIPE 14	0,7894
EQUIPE 1	0,7733	EQUIPE 1	0,7733	EQUIPE 1	0,7691

Tabela 36. Versão final do Mapeamento equipes × módulos (E-commerce – aplicação – política de provisão de treinamento - Conjunto B)

RESULTADO FINAL					
M1		M2		M3	
EQUIPE 13	0,9216	EQUIPE 13	0,9216	EQUIPE 13	0,9336
EQUIPE 9	0,9202	EQUIPE 9	0,9202	EQUIPE 9	0,9308
EQUIPE 8	0,9050	EQUIPE 8	0,9050	EQUIPE 2	0,9067
EQUIPE 2	0,9013	EQUIPE 2	0,9013	EQUIPE 5	0,8822
EQUIPE 7	0,8448	EQUIPE 7	0,8448	EQUIPE 8	0,8779
EQUIPE 10	0,8441	EQUIPE 10	0,8441	EQUIPE 7	0,8545
EQUIPE 3	0,8393	EQUIPE 3	0,8393	EQUIPE 10	0,8441
EQUIPE 5	0,8243	EQUIPE 5	0,8243	EQUIPE 3	0,8391
EQUIPE 6	0,8039	EQUIPE 6	0,8039	EQUIPE 6	0,8039
EQUIPE 15	0,8037	EQUIPE 15	0,8037	EQUIPE 15	0,7770
EQUIPE 12	0,7703	EQUIPE 12	0,7703	EQUIPE 12	0,7759
EQUIPE 1	0,6872	EQUIPE 1	0,6872	EQUIPE 4	0,7093
EQUIPE 4	0,6188	EQUIPE 4	0,6188	EQUIPE 1	0,6817
EQUIPE 11	0,3015	EQUIPE 11	0,3015	EQUIPE 11	0,3755
EQUIPE 14	0,1911	EQUIPE 14	0,1911	EQUIPE 14	0,1339

5.3 Multi-MOM

O Multi-MOM consiste em uma linha de produtos de *middleware* para computação móvel que está essencialmente focada na funcionalidade de comunicação [Bezerra, 2010]. A Figura 17 descreve a arquitetura de domínio desta LPS.

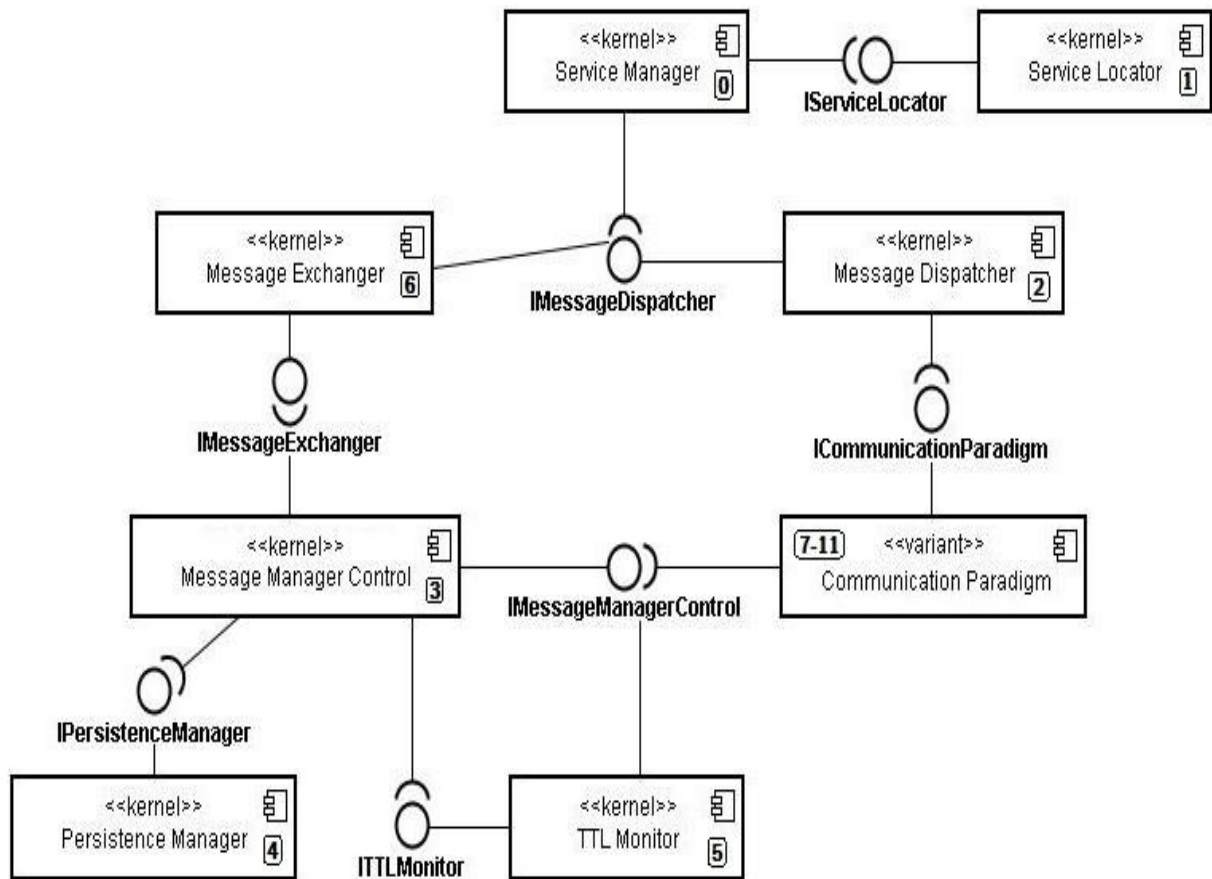


Figura 17. Arquitetura de domínio do Multi-MOM.

A partir desta arquitetura foram definidos os módulos de *software*, segundo a fase de análise arquitetural do *framework* de alocações [Pereira, 2011]. Foram identificados cinco módulos distintos:

- Módulo M0 – composto pelos componentes c2, c3, c7, c8, c9, c10, c11;
- Módulo M1 – composto pelos componentes c0, c1;
- Módulo M2 – composto pelos componentes c4;
- Módulo M3 – composto pelos componentes c6;
- Módulo M4 – composto pelos componentes c5;

A descrição das tabelas de implementação foram feitas pelo arquiteto que projetou o Multi-MOM. Apesar de ser permitidas mudanças quanto ao número de termos ou sobre quais termos utilizar, o arquiteto do Multi-MOM achou os termos padrão ‘Baixo’, ‘Médio’ e ‘Alto’ suficientes para categorizar a complexidade esperada de cada tecnologia. As tabelas de implementação definidas para o Multi-MOM podem ser observadas na Tabela 37, Tabela 38, Tabela 39, Tabela 40 e Tabela 41.

Tabela 37. Tabela de implementação de M0 (Multi-MOM)

Baixo	Médio	Alto
	SQL	Java
		Android
		Programação Reflexiva
		Paradigmas de Rede

Tabela 38. Tabela de implementação de M1 (Multi-MOM)

Baixo	Médio	Alto
SQL	Java	Android
Programação Reflexiva	Paradigmas de Rede	Protocolos de Rede
		Sockets

Tabela 39. Tabela de implementação de M2 (Multi-MOM)

Baixo	Médio	Alto
Paradigmas de Rede	Java	SQL
		Android
		Programação Reflexiva

Tabela 40. Tabela de implementação de M3 (Multi-MOM)

Baixo	Médio	Alto
SQL	Java	Sockets
	Android	
	Programação Reflexiva	
	Paradigmas de Rede	
	Protocolos de Redes	

Tabela 41. Tabela de implementação de M4 (Multi-MOM)

Baixo	Médio	Alto
Paradigmas de Rede	Java	
	SQL	
	Android	
	Programação Reflexiva	

Como não aconteceram mudanças na estrutura dos termos *fuzzy* utilizados nem das funções de pertinência, a etapa de *Descrição dos módulos* foi concluída com a realização das tabelas de implementação feitas pelo arquiteto do Multi-MOM. Como nos estudos de casos anteriores, as próximas seções avaliam a abordagem proposta considerando os dois grupos de equipes distintas.

5.3.1 Estudo com Equipes do Conjunto A

O resultado do cálculo do valor *crisp* de conhecimento das equipes nas tecnologias identificadas pode ser observado na Tabela 42. Na Tabela 43 é apresentado o valor *crisp* calculado representando a tabela de implementação de cada módulo.

Tabela 42. Valores Crisp das habilidades técnica das equipes.

EQUIPES	EQUIPE 1 (9)	EQUIPE 2 (5)	EQUIPE 3 (18)	EQUIPE 4 (8)	EQUIPE 5 (9)	EQUIPE 6 (2)	EQUIPE 7 (5)
ANDROID	0,3750	0,0000	0,3969	0,3750	0,8750	0,0000	0,3750
JAVA	0,7107	0,6438	0,7064	0,6489	0,6811	0,7378	0,7286
PROGRAMACAO REFLEXIVA	0,8415	0,6981	0,8172	0,8393	0,8255	0,7378	0,7631
SQL	0,7662	0,7137	0,7996	0,7932	0,7670	0,8722	0,7793
SOCKETS	0,7140	0,6981	0,8219	0,8084	0,8499	0,5833	0,5833
PARADIGMAS DE REDE	0,9047	0,5833	0,7785	0,7609	0,7455	0,5833	0,6675
PROTOCOLOS DE REDE	0,7700	0,7609	0,8309	0,8636	0,7912	1,0000	0,7842
REPUTACAO	0,3426	0,9197	0,5000	0,1787	0,4116	0,5189	0,4715

Tabela 43. Valores Crisp da complexidade esperada das tecnologias (Multi-MOM).

MODULOS	M0	M1	M2	M3	M4
ANDROID	0,8000	0,8000	0,8000	0,5000	0,5000
JAVA	0,8000	0,5000	0,5000	0,5000	0,5000
PROGRAMACAO REFLEXIVA	0,8000	0,2000	0,8000	0,5000	0,5000
SQL	0,5000	0,2000	0,8000	0,2000	0,5000
SOCKETS		0,8000		0,8000	
PARADIGMAS DE REDE	0,8000	0,5000	0,2000	0,5000	0,2000
PROTOCOLOS DE REDE		0,8000		0,5000	

Utilizando a **política de qualificação equivalente**, são obtidos os resultados apresentados na Tabela 44 (sem cálculo de reputação) e Tabela 45 (com cálculo de reputação).

Tabela 44. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação equivalente – Conjunto A).

RESULTADO (SEM REPUTACAO)									
M0		M1		M2		M3		M4	
EQUIPE 5	0,8645	EQUIPE 5	0,8056	EQUIPE 5	0,8849	EQUIPE 2	0,7878	EQUIPE 2	0,8004
EQUIPE 1	0,7767	EQUIPE 2	0,7202	EQUIPE 4	0,7638	EQUIPE 5	0,7809	EQUIPE 7	0,7740
EQUIPE 3	0,7401	EQUIPE 4	0,6996	EQUIPE 3	0,7524	EQUIPE 4	0,7584	EQUIPE 5	0,7495
EQUIPE 4	0,7296	EQUIPE 7	0,6877	EQUIPE 7	0,7484	EQUIPE 7	0,7562	EQUIPE 4	0,7267
EQUIPE 7	0,6946	EQUIPE 3	0,6852	EQUIPE 1	0,7079	EQUIPE 3	0,7440	EQUIPE 3	0,7059
EQUIPE 2	0,6357	EQUIPE 6	0,6587	EQUIPE 6	0,6992	EQUIPE 1	0,7260	EQUIPE 1	0,7028
EQUIPE 6	0,6277	EQUIPE 1	0,6401	EQUIPE 2	0,6547	EQUIPE 6	0,6933	EQUIPE 6	0,7000

Tabela 45. Versão final do Mapeamento equipes × módulos (Multi-MOM - política de qualificação equivalente - Conjunto A).

RESULTADO FINAL									
M0		M1		M2		M3		M4	
EQUIPE 5	0,8379	EQUIPE 2	0,8366	EQUIPE 5	0,8619	EQUIPE 2	0,8784	EQUIPE 2	0,8860
EQUIPE 2	0,7817	EQUIPE 5	0,7690	EQUIPE 2	0,7943	EQUIPE 3	0,7440	EQUIPE 7	0,7621
EQUIPE 3	0,7401	EQUIPE 3	0,6852	EQUIPE 3	0,7524	EQUIPE 7	0,7436	EQUIPE 6	0,7092
EQUIPE 1	0,6915	EQUIPE 7	0,6723	EQUIPE 7	0,7354	EQUIPE 5	0,7405	EQUIPE 3	0,7059
EQUIPE 7	0,6794	EQUIPE 6	0,6688	EQUIPE 6	0,7083	EQUIPE 6	0,7026	EQUIPE 5	0,7045
EQUIPE 6	0,6385	EQUIPE 1	0,5215	EQUIPE 1	0,6040	EQUIPE 1	0,6267	EQUIPE 1	0,5977
EQUIPE 4	0,4140	EQUIPE 4	0,3680	EQUIPE 4	0,4706	EQUIPE 4	0,4613	EQUIPE 4	0,4093

É interessante notar a diferença de resultados entre módulos. Por exemplo, verificando as adequabilidades calculadas da Equipe 2, percebe-se que ela é selecionada como a mais adequada para implementação do módulo M4, enquanto ela tem sua adequabilidade calculada com o valor mais baixo para a implementação do módulo M2 (Tabela 44). Esse resultado pode ser verificado analisando os seus valores de conhecimento de cada tecnologia e os valores requeridos pelos módulos. Note que, para a implementação do módulo M2, é requerido que uma equipe possua: ‘alto’ conhecimento em ANDROID, PROGRAMAÇÃO REFLEXIVA e SQL; ‘médio’ conhecimento em JAVA; e ‘baixo’ conhecimento em PARADIGMAS DE REDE. Para essas tecnologias, a Equipe 2 apresenta apenas o conhecimento próximo do requerido para a tecnologia JAVA e SOCKETS. Para as demais, ou ela sabe mais do que é requerido ou sabe menos. Uma peculiaridade deve ser observada para tecnologia ANDROID, na qual o conhecimento da Equipe 2 é 0.000. Contudo, conforme indicado na Tabela 43, é necessário que uma equipe possua um conhecimento ‘alto’ para que

possa desenvolver o módulo. Devido a esta tecnologia, a Equipe 2 será penalizada para que outras equipes com conhecimentos mais próximos tenham mais chances de serem escolhidas.

Já o módulo M4 requer: conhecimento ‘médio’ em ANDROID, JAVA, PROGRAMAÇÃO REFLEXIVA e SQL; e conhecimento ‘baixo’ em PARADIGMAS DE REDE. Comparando os valores apresentados de conhecimento nas tecnologias exigidas por esse módulo para todas as equipes, verifica-se que a Equipe 2 é a que possui os conhecimentos mais próximos do conhecimento ‘médio’. Com exceção da tecnologia PARADIGMAS DE REDE, o módulo demanda um conhecimento médio. Assim, nada mais que justo que a Equipe 2 possua a melhor pontuação. Outra relação interessante de se notar nesse caso é o cálculo de adequabilidade com reputação para as equipes 2 e 5 no módulo M2. Repare que a equipe 5 é calculada primeiramente com adequabilidade 0.8849 e a Equipe 2 com o valor 0.6547. Ao verificar os valores de reputação para ambas, percebe-se que a Equipe 5 apresenta o valor 0.4116 e a Equipe 2 tem o valor 0.9197. A Equipe 2 apresenta o maior valor de reputação dentro do grupo de equipes. Já a equipe 5 apresenta um valor ligeiramente mais baixo que o valor neutro. Devido a estes dois fatos, após o novo cálculo de adequabilidade, a Equipe 2 passa de última para segunda da lista de adequabilidades, perdendo apenas para a própria Equipe 5, apesar da reputação da Equipe 5 ser bem mais baixa que a da Equipe 2. Mas de fato, deve-se lembrar que a reputação não é um fator que vem a sobrepor a adequabilidade puramente técnica, mas um fator que vem a agregar ao valor da adequabilidade. Assim, como para a Equipe 5 foi calculado um valor alto de adequabilidade, não seria justo torná-la inadequada apenas por sua reputação ser ligeiramente menor que o ponto neutro de reputação. Note que, de fato, a Equipe 5 é penalizada por conta de sua reputação, mas mesmo assim continua sendo a melhor opção, pois de fato ela se apresentou como a mais adequada.

Como já mencionado, o gerente de projetos pode a qualquer momento criar novas políticas do zero ou derivar novas políticas a partir de outras já existentes. Nesse trabalho foi apresentada uma política que se derivou de outra. Ela é denominada **política de qualificação padrão** (Tabela 5) e foi derivada da **política de qualificação equivalente** (Tabela 3). Assim, a seguir, será realizada uma comparação entre os valores de adequabilidade calculados entre ambas. Os valores de adequabilidade para a **política de qualificação padrão** são demonstrados na Tabela 46 (sem cálculo de reputação) e Tabela 47 (com cálculo de reputação).

Tabela 46. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação padrão – Conjunto A).

RESULTADO (SEM REPUTACAO)									
M0		M1		M2		M3		M4	
EQUIPE 5	0,9059	EQUIPE 5	0,9946	EQUIPE 5	0,9762	EQUIPE 5	1,0000	EQUIPE 5	1,0000
EQUIPE 1	0,8177	EQUIPE 3	0,9076	EQUIPE 3	0,8702	EQUIPE 3	0,9580	EQUIPE 7	0,9412
EQUIPE 3	0,7987	EQUIPE 4	0,9076	EQUIPE 4	0,8646	EQUIPE 4	0,9580	EQUIPE 1	0,9412
EQUIPE 4	0,7846	EQUIPE 7	0,8564	EQUIPE 1	0,8464	EQUIPE 1	0,9215	EQUIPE 3	0,9412
EQUIPE 7	0,7422	EQUIPE 1	0,8554	EQUIPE 7	0,8285	EQUIPE 7	0,9160	EQUIPE 4	0,9412
EQUIPE 6	0,6866	EQUIPE 6	0,8319	EQUIPE 6	0,7845	EQUIPE 6	0,8319	EQUIPE 6	0,8235
EQUIPE 2	0,6471	EQUIPE 2	0,8125	EQUIPE 2	0,7135	EQUIPE 2	0,8319	EQUIPE 2	0,8235

Tabela 47. Versão final do Mapeamento equipes × módulos (Multi-MOM – política de qualificação padrão - Conjunto A).

RESULTADO FINAL									
M0		M1		M2		M3		M4	
EQUIPE 5	0,8869	EQUIPE 5	0,9934	EQUIPE 5	0,9712	EQUIPE 5	1,0000	EQUIPE 5	1,0000
EQUIPE 3	0,7987	EQUIPE 3	0,9076	EQUIPE 3	0,8702	EQUIPE 3	0,9580	EQUIPE 3	0,9412
EQUIPE 2	0,7893	EQUIPE 2	0,8933	EQUIPE 2	0,8323	EQUIPE 7	0,9111	EQUIPE 7	0,9377
EQUIPE 1	0,7455	EQUIPE 7	0,8484	EQUIPE 7	0,8191	EQUIPE 2	0,9048	EQUIPE 1	0,9153
EQUIPE 7	0,7289	EQUIPE 6	0,8375	EQUIPE 6	0,7914	EQUIPE 1	0,8875	EQUIPE 2	0,8998
EQUIPE 6	0,6960	EQUIPE 1	0,7961	EQUIPE 1	0,7839	EQUIPE 4	0,8868	EQUIPE 4	0,8440
EQUIPE 4	0,5072	EQUIPE 4	0,7623	EQUIPE 4	0,6656	EQUIPE 6	0,8375	EQUIPE 6	0,8294

Dentre as políticas de seleção apresentadas até o momento, a política de qualificação padrão apresentou em média os maiores valores de adequabilidade para todas as equipes. Para entender esse comportamento é necessário entender o sentido dessa política. Como já mencionado, ela é derivada da **política de qualificação equivalente**. A mudança realizada na política teve o intuito de tornar não adequadas as equipes que não possuem conhecimento, tornando mais adequadas aquelas que possuem conhecimento necessário ou maior que o requerido pelos módulos. No caso de equipes que possuem algum conhecimento, mas menor do que o requerido, ganham poucos pontos, diminuindo a pontuação quanto mais se afasta do conhecimento requerido pelos módulos. Essa regra tenta se assemelhar às regras utilizadas para a maioria das seleções de RH de empresas, ou seja, quem não possui conhecimento não é escolhido, quem tem o mínimo de conhecimento pode ser selecionado desde que não exista alguém que tenha mais conhecimento, pois esse tende a ser mais visado.

Com o propósito da política entendida, fica fácil entender o comportamento do resultado gerado. Note que a Equipe 5 é a aquela que apresenta maior adequabilidade para qualquer um dos módulos (Tabela 46). Ao verificar o conhecimento que essa equipe apresenta para cada tecnologia, nota-se que essa equipe é a que apresenta os conhecimentos entre os níveis ‘médio’ e ‘alto’ (lembrando que médio se encontra entre 0.4 e 0.8 e alto entre 0.7 e 1).

Agora, observando a tabela de regras da política, percebe-se que quanto maior o conhecimento das equipes, maiores são as chances de obter uma adequabilidade alta. Como, de fato, a Equipe 5 é a que apresenta os maiores conhecimentos, maior a chance dela ser a mais adequada. Não se deve confundir essa política com a **política de equipes mais hábeis**, pois esta verifica as equipes que possuem o maior conhecimento numa certa tecnologia, sem verificar o nível exigido. Já a **política de qualificação padrão** verifica o nível requerido pela tecnologia, e, caso o conhecimento seja igual ou superior, a equipe ganhará pontos.

5.3.2 Estudo com Equipes do Conjunto B

O resultado do cálculo do valor *crisp* de conhecimento das equipes nas tecnologias identificadas pode ser observado na Tabela 48.

Tabela 48. Valores Crisp das habilidades técnica das equipes (Conjunto B).

	EQUIPE 1	EQUIPE 2	EQUIPE 3	EQUIPE 4	EQUIPE 5	EQUIPE 6	EQUIPE 7	EQUIPE 8
EQUIPES	(3)	(3)	(3)	(5)	(5)	(5)	(8)	(8)
ANDROID	0,3555	0,3440	0,3920	0,3251	0,4074	0,3124	0,2362	0,4739
JAVA	0,6085	0,5694	0,4890	0,5554	0,4349	0,5941	0,4396	0,5838
PROGRAMACAO REFLEXIVA	0,6834	0,4526	0,4300	0,5771	0,3693	0,6383	0,6152	0,5745
SQL	0,7008	0,5429	0,5185	0,7631	0,4906	0,5986	0,6751	0,6484
SOCKETS	0,6659	0,4348	0,4126	0,6834	0,6383	0,4695	0,6269	0,4778
PARADIGMAS DE REDE	0,6300	0,2165	0,4212	0,4204	0,3483	0,4807	0,5142	0,4497
PROTOCOLOS DE REDE	0,6300	0,3321	0,4807	0,5462	0,4212	0,5843	0,6249	0,5018
REPUTACAO	0,3426	0,9197	0,5000	0,1787	0,4116	0,5189	0,4715	0,7251
	EQUIPE 9	EQUIPE 10	EQUIPE 11	EQUIPE 12	EQUIPE 13	EQUIPE 14	EQUIPE 15	
EQUIPES	(8)	(10)	(10)	(10)	(15)	(15)	(15)	
ANDROID	0,2861	0,4239	0,4490	0,3652	0,4332	0,4490	0,3973	
JAVA	0,5231	0,3913	0,5016	0,5659	0,4962	0,4783	0,5677	
PROGRAMACAO REFLEXIVA	0,6251	0,3401	0,7776	0,4004	0,4360	0,4508	0,5957	
SQL	0,5451	0,5899	0,6238	0,7117	0,5483	0,6515	0,7045	
SOCKETS	0,5216	0,4858	0,6357	0,5631	0,4946	0,5417	0,6774	
PARADIGMAS DE REDE	0,4856	0,4744	0,5501	0,5716	0,4502	0,5432	0,5958	
PROTOCOLOS DE REDE	0,4558	0,3570	0,6674	0,5578	0,4239	0,5432	0,6088	
REPUTACAO	0,9567	0,4691	0,0610	0,3892	0,9738	0,0588	0,3556	

Utilizando a **política de qualificação equivalente**, são obtidos os resultados apresentados na Tabela 49 (sem cálculo de reputação) e Tabela 50 (com cálculo de reputação).

Tabela 49. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação equivalente – Conjunto B).

RESULTADO (SEM REPUTACAO)									
M0		M1		M2		M3		M4	
EQUIPE 11	0,7703	EQUIPE 12	0,7741	EQUIPE 8	0,7704	EQUIPE 1	0,8734	EQUIPE 8	0,9475
EQUIPE 8	0,7089	EQUIPE 14	0,7705	EQUIPE 11	0,7703	EQUIPE 15	0,8711	EQUIPE 2	0,9096
EQUIPE 1	0,6934	EQUIPE 8	0,7391	EQUIPE 4	0,7698	EQUIPE 8	0,8657	EQUIPE 6	0,8928
EQUIPE 15	0,6901	EQUIPE 1	0,7390	EQUIPE 2	0,7192	EQUIPE 11	0,8587	EQUIPE 9	0,8903
EQUIPE 9	0,6820	EQUIPE 15	0,7366	EQUIPE 6	0,7046	EQUIPE 14	0,8568	EQUIPE 4	0,8856
EQUIPE 6	0,6785	EQUIPE 11	0,7266	EQUIPE 9	0,7021	EQUIPE 7	0,8467	EQUIPE 13	0,8822
EQUIPE 14	0,6767	EQUIPE 13	0,7148	EQUIPE 13	0,6975	EQUIPE 6	0,8465	EQUIPE 1	0,8816
EQUIPE 7	0,6525	EQUIPE 7	0,7122	EQUIPE 15	0,6967	EQUIPE 9	0,8441	EQUIPE 15	0,8783
EQUIPE 13	0,6366	EQUIPE 6	0,7120	EQUIPE 1	0,6946	EQUIPE 12	0,8249	EQUIPE 3	0,8724
EQUIPE 12	0,6140	EQUIPE 9	0,7083	EQUIPE 3	0,6795	EQUIPE 4	0,8089	EQUIPE 11	0,8610
EQUIPE 4	0,6005	EQUIPE 5	0,6807	EQUIPE 14	0,6772	EQUIPE 13	0,7956	EQUIPE 14	0,8584
EQUIPE 2	0,5898	EQUIPE 10	0,6802	EQUIPE 7	0,6560	EQUIPE 3	0,7491	EQUIPE 7	0,8442
EQUIPE 3	0,5834	EQUIPE 4	0,6745	EQUIPE 5	0,6402	EQUIPE 2	0,7355	EQUIPE 5	0,8379
EQUIPE 10	0,5518	EQUIPE 3	0,6714	EQUIPE 12	0,6304	EQUIPE 10	0,7347	EQUIPE 12	0,8139
EQUIPE 5	0,5068	EQUIPE 2	0,6327	EQUIPE 10	0,5972	EQUIPE 5	0,7339	EQUIPE 10	0,7912

Tabela 50. Versão final do Mapeamento equipes × módulos (Multi-MOM - política de qualificação equivalente - Conjunto B).

RESULTADO FINAL									
M0		M1		M2		M3		M4	
EQUIPE 9	0,8187	EQUIPE 13	0,8416	EQUIPE 2	0,8360	EQUIPE 9	0,9152	EQUIPE 8	0,9635
EQUIPE 13	0,7930	EQUIPE 9	0,8351	EQUIPE 8	0,8354	EQUIPE 8	0,9053	EQUIPE 2	0,9498
EQUIPE 8	0,7888	EQUIPE 8	0,8118	EQUIPE 9	0,8312	EQUIPE 13	0,8892	EQUIPE 9	0,9411
EQUIPE 2	0,7505	EQUIPE 2	0,7797	EQUIPE 13	0,8311	EQUIPE 6	0,8516	EQUIPE 13	0,9377
EQUIPE 6	0,6881	EQUIPE 6	0,7209	EQUIPE 6	0,7136	EQUIPE 2	0,8462	EQUIPE 6	0,8965
EQUIPE 7	0,6359	EQUIPE 12	0,7197	EQUIPE 3	0,6795	EQUIPE 7	0,8382	EQUIPE 3	0,8724
EQUIPE 15	0,5936	EQUIPE 7	0,6978	EQUIPE 7	0,6395	EQUIPE 15	0,8236	EQUIPE 7	0,8356
EQUIPE 1	0,5860	EQUIPE 3	0,6714	EQUIPE 15	0,6016	EQUIPE 1	0,8208	EQUIPE 15	0,8333
EQUIPE 3	0,5834	EQUIPE 10	0,6631	EQUIPE 1	0,5875	EQUIPE 12	0,7809	EQUIPE 1	0,8320
EQUIPE 12	0,5344	EQUIPE 15	0,6506	EQUIPE 5	0,5817	EQUIPE 3	0,7491	EQUIPE 5	0,8067
EQUIPE 10	0,5306	EQUIPE 1	0,6431	EQUIPE 10	0,5773	EQUIPE 10	0,7199	EQUIPE 10	0,7791
EQUIPE 5	0,4380	EQUIPE 5	0,6267	EQUIPE 12	0,5528	EQUIPE 5	0,6868	EQUIPE 12	0,7675
EQUIPE 4	0,2400	EQUIPE 4	0,3322	EQUIPE 4	0,4809	EQUIPE 4	0,5525	EQUIPE 4	0,7118
EQUIPE 11	0,1178	EQUIPE 14	0,1089	EQUIPE 11	0,1178	EQUIPE 11	0,2869	EQUIPE 11	0,2933
EQUIPE 14	0,0361	EQUIPE 11	0,0730	EQUIPE 14	0,0363	EQUIPE 14	0,2687	EQUIPE 14	0,2729

Para este conjunto de equipes é interessante notar a diferença dos resultados obtidos entre os módulos M0 e M4, apresentados na Tabela 49. Note que o módulo M0 requer para as tecnologias ANDROID, JAVA, PROGRAMAÇÃO REFLEXIVA e PARADIGMAS DE REDE o valor de conhecimento ‘alto’ e para a tecnologia SQL o conhecimento ‘médio’ (Tabela 43). Já o módulo M4 requer para as tecnologias ANDROID, JAVA, PROGRAMAÇÃO REFLEXIVA e SQL o conhecimento ‘médio’ e para a tecnologia PARADIGMAS DE REDE o conhecimento ‘baixo’. Como esse grupo de equipes apresenta

valores de conhecimento ‘médio’ para a maioria das tecnologias, e como o módulo M4 requer para maioria de suas tecnologias o nível ‘médio’ de conhecimento, é justificável que os valores de adequabilidade para esse módulo sejam mais altos, já que a política requer que o conhecimento das equipes seja o mais próximo ao requerido pelos módulos. Por outro lado, como para as tecnologias requeridas para o módulo M0 exige-se conhecimento ‘alto’, e as equipes não apresentam em média valores altos de conhecimento, é justificado que os níveis de adequabilidades, para esse módulo sejam mais baixos que os apresentados para o módulo M4.

Agora, ao analisar a **política de qualificação padrão** (Tabela 5), verificam-se os resultados obtidos na Tabela 51 (sem cálculo de reputação) e Tabela 52 (com cálculo de reputação).

Tabela 51. Primeira versão do Mapeamento equipes × módulos (Multi-MOM - política de qualificação padrão – Conjunto B).

RESULTADO (SEM REPUTACAO)									
M0		M1		M2		M3		M4	
EQUIPE 11	0,7703	EQUIPE 11	0,8483	EQUIPE 11	0,8291	EQUIPE 11	0,9341	EQUIPE 8	0,9803
EQUIPE 8	0,7089	EQUIPE 14	0,8363	EQUIPE 8	0,8032	EQUIPE 1	0,9160	EQUIPE 11	0,9666
EQUIPE 1	0,6941	EQUIPE 1	0,8235	EQUIPE 4	0,7858	EQUIPE 15	0,9160	EQUIPE 1	0,9412
EQUIPE 15	0,6941	EQUIPE 12	0,8235	EQUIPE 15	0,7555	EQUIPE 8	0,9077	EQUIPE 4	0,9412
EQUIPE 9	0,6820	EQUIPE 15	0,8235	EQUIPE 1	0,7534	EQUIPE 14	0,8988	EQUIPE 9	0,9412
EQUIPE 6	0,6785	EQUIPE 8	0,8231	EQUIPE 9	0,7529	EQUIPE 7	0,8887	EQUIPE 6	0,9412
EQUIPE 14	0,6767	EQUIPE 7	0,7963	EQUIPE 6	0,7529	EQUIPE 6	0,8885	EQUIPE 15	0,9412
EQUIPE 7	0,6525	EQUIPE 6	0,7960	EQUIPE 14	0,7360	EQUIPE 9	0,8861	EQUIPE 14	0,9172
EQUIPE 4	0,6400	EQUIPE 9	0,7923	EQUIPE 13	0,7306	EQUIPE 4	0,8819	EQUIPE 13	0,9152
EQUIPE 13	0,6366	EQUIPE 4	0,7894	EQUIPE 2	0,7192	EQUIPE 12	0,8741	EQUIPE 2	0,9096
EQUIPE 12	0,6239	EQUIPE 13	0,7750	EQUIPE 7	0,7148	EQUIPE 13	0,8377	EQUIPE 7	0,9030
EQUIPE 2	0,5898	EQUIPE 3	0,7291	EQUIPE 3	0,6960	EQUIPE 3	0,7912	EQUIPE 3	0,8890
EQUIPE 3	0,5834	EQUIPE 10	0,7222	EQUIPE 12	0,6893	EQUIPE 2	0,7775	EQUIPE 12	0,8826
EQUIPE 10	0,5518	EQUIPE 5	0,7189	EQUIPE 10	0,6424	EQUIPE 10	0,7767	EQUIPE 5	0,8379
EQUIPE 5	0,5068	EQUIPE 2	0,6992	EQUIPE 5	0,6402	EQUIPE 5	0,7722	EQUIPE 10	0,8363

Tabela 52. Versão final do Mapeamento equipes × módulos (Multi-MOM – política de qualificação padrão - Conjunto B).

RESULTADO FINAL									
M0		M1		M2		M3		M4	
EQUIPE 9	0,8187	EQUIPE 9	0,8855	EQUIPE 9	0,8622	EQUIPE 9	0,9388	EQUIPE 8	0,9864
EQUIPE 13	0,7930	EQUIPE 13	0,8773	EQUIPE 8	0,8597	EQUIPE 8	0,9354	EQUIPE 9	0,9688
EQUIPE 8	0,7888	EQUIPE 8	0,8744	EQUIPE 13	0,8511	EQUIPE 13	0,9131	EQUIPE 13	0,9555
EQUIPE 2	0,7505	EQUIPE 2	0,8232	EQUIPE 2	0,8360	EQUIPE 6	0,8923	EQUIPE 2	0,9498
EQUIPE 6	0,6881	EQUIPE 6	0,8027	EQUIPE 6	0,7608	EQUIPE 15	0,8839	EQUIPE 6	0,9433
EQUIPE 7	0,6359	EQUIPE 7	0,7854	EQUIPE 7	0,7004	EQUIPE 7	0,8824	EQUIPE 15	0,9183
EQUIPE 15	0,5985	EQUIPE 12	0,7792	EQUIPE 3	0,6960	EQUIPE 1	0,8798	EQUIPE 1	0,9153
EQUIPE 1	0,5869	EQUIPE 15	0,7611	EQUIPE 15	0,6742	EQUIPE 2	0,8721	EQUIPE 7	0,8975
EQUIPE 3	0,5834	EQUIPE 1	0,7533	EQUIPE 1	0,6615	EQUIPE 12	0,8413	EQUIPE 3	0,8890
EQUIPE 12	0,5455	EQUIPE 3	0,7291	EQUIPE 10	0,6239	EQUIPE 3	0,7912	EQUIPE 12	0,8517
EQUIPE 10	0,5306	EQUIPE 10	0,7069	EQUIPE 12	0,6200	EQUIPE 10	0,7639	EQUIPE 4	0,8440
EQUIPE 5	0,4380	EQUIPE 5	0,6698	EQUIPE 5	0,5817	EQUIPE 5	0,7304	EQUIPE 10	0,8266
EQUIPE 4	0,2869	EQUIPE 4	0,5161	EQUIPE 4	0,5094	EQUIPE 4	0,7035	EQUIPE 5	0,8067
EQUIPE 11	0,1178	EQUIPE 11	0,2597	EQUIPE 11	0,2153	EQUIPE 11	0,5719	EQUIPE 11	0,7568
EQUIPE 14	0,0361	EQUIPE 14	0,2187	EQUIPE 14	0,0738	EQUIPE 14	0,4037	EQUIPE 14	0,4794

Novamente percebe-se que os valores de adequabilidade para a **política de qualificação padrão** são em média mais altos que os demonstrados para as demais políticas. Ainda deve-se notar que o comportamento do resultado apresentado na Tabela 51 é diferente dos apresentados para a **política de qualificação equivalente**, com exceção do módulo M0. Isso é justificado porque a **política de qualificação padrão** seleciona equipes que possuem no mínimo o nível de conhecimento exigido pelo módulo. Já a **política de qualificação equivalente** seleciona equipes que possuem o conhecimento próximo ao exigido pelo módulo, seja esse conhecimento para mais ou para menos. Como o módulo M0 requer, para a maioria das tecnologias, ‘alto’ nível de conhecimento, e as equipes apresentam em média conhecimento ‘médio’ para as tecnologias exigidas, ambas as políticas acabam por fazer uma seleção semelhante. Já para os demais módulos que apresentam em sua maioria tecnologias que demandam conhecimento de nível ‘médio’, a seleção demonstra resultados diferentes. Note por exemplo os resultados obtidos para o módulo M2 (Tabela 51) utilizando a **política de qualificação equivalente** e a **política de qualificação padrão**. As equipes tendem a ter valores de adequabilidades maiores quando se utiliza a segunda política. Isso se deve ao fato da própria política que tende a atribuir maiores pontuações às equipes, visto que equipes que possuem o conhecimento mais alto que o requerido pelos módulos também se tornam adequadas, diferentemente da **política de qualificação equivalente** que tende a dar menos pontos quanto maior a distância do conhecimento requerido e o apresentado pelas equipes.

5.4 Considerações Finais

Neste capítulo foi realizada uma apresentação dos resultados dos estudos de casos. Dois estudos de casos foram tomados como base: E-Commerce e Multi-MOM. Para cada um dos estudos de casos foram realizadas diferentes avaliações. Para o E-Commerce foi gerado o estudo de duas iterações, um sendo de engenharia de domínio e outra de engenharia de aplicação. Tanto para o E-Commerce quanto para o Multi-MOM foram utilizados dois conjuntos diferentes de equipes e duas políticas de seleção para cada conjunto de equipes. Assim totalizando doze estudos. Cada resultado foi analisado e os aspectos mais interessantes foram discutidos neste capítulo, comprovando a aplicabilidade e funcionalidade a qual se propõem esse trabalho. Não houve diferenças perceptíveis na qualidade dos resultados referentes ao tipo de formação dos grupos de equipes A (randômica) e B (guiada), o que indica que a abordagem deve funcionar com qualquer formação de equipes.

Desse modo, verifica-se a aplicabilidade de políticas diferentes para necessidades diferentes, ou seja, nem sempre o gerente de projetos necessita alocar as equipes mais hábeis para seu projeto, pois, as vezes, nem se torna necessário dependendo do nível de complexidade que é requerido para o desenvolvimento dos módulos. Assim, a abordagem se torna flexível ao ponto de permitir variar as regras de seleção de equipes de acordo com as especificidades requeridas por cada projeto.

Capítulo 6

Trabalhos relacionados

O presente capítulo tem como objetivo apresentar alguns trabalhos relacionados ao tema do trabalho proposto, compará-los e evidenciar seus pontos fracos e fortes. Apesar do presente trabalho ter por objetivo principal selecionar um conjunto de equipes distribuídas tecnicamente qualificadas para desenvolvimento de projetos de *softwares*, foram selecionados também trabalhos de recomendação de especialistas, pois apresentam aspectos relevantes aos abordados nesta proposta.

Foram selecionados trabalhos mais próximos possíveis do tema abordado. Porém, a grande maioria dos trabalhos nesta linha consideram apenas alocação ou seleção de **pessoas** para auxílio imediato ou para formação de uma equipe de implementação. A literatura é escassa quando se trata de alocação de **equipes** terceirizadas ou de subsidiárias de uma empresa, principalmente quando estas estão espalhadas pelo globo. Outro aspecto que os trabalhos relacionados não abordam é o fato do projeto de *software* ao qual se destina o suporte poder ser um projeto de LPS. A grande maioria dos trabalhos presentes na literatura, inclusive os abordados neste capítulo, tomam por base um projeto de *software* qualquer e não de LPS. A seguir são apresentados os trabalhos relacionados.

6.1. ICARE

O ICARE (*Intelligent Context Awareness for Recommending Experts*) [Petry, 2007] é um Sistema de Recomendação de Especialista (SRE) e como tal, é um sistema que retorna referências para atores humanos acerca de especialistas em um determinado domínio requisitado [Reichling, et al., 2005]. Diferentemente da maioria dos SRE que têm o foco apenas no especialista, o ICARE também considera aspectos do usuário e apresenta como principal diferencial o fato de levar em consideração o contexto atual dos envolvidos na recomendação. O contexto é representado por informações que caracterizam a situação atual de uma entidade [Dey, et al., 2001]. Assim o objetivo do ICARE é apoiar a etapa inicial de uma colaboração informal, ou seja, buscar um especialista que possa ajudar um usuário em determinados problemas. Leva em consideração aspectos que ao mesmo tempo identifiquem

especialistas com bom conhecimento técnico acerca do assunto e que maximizem a afinidade entre os especialistas e usuários envolvidos na recomendação.

Para atingir tal propósito, são analisados atributos que identifiquem o contexto atual tanto do usuário quanto do especialista, a fim de gerar recomendações mais adequadas a presente situação de ambos. Entre tais atributos são analisados, por exemplo, a disponibilidade, acessibilidade, nível organizacional, rede social, tarefa atual, reputação, habilidades e grau de especialidade tanto do especialista quanto do usuário requerente.

O processo de recomendação no ICARE se dá em três etapas:

- *Determinar especialidade* – identifica quais as especialidades requisitadas pelo usuário;
- *Recuperar especialistas* – com base nas especialidades definidas pelo usuário e nos atributos contextuais, cria-se uma lista de especialistas encontrados ordenada por maior adequação entre usuário e especialista;
- *Apresentar especialista* – mostra a lista de especialistas encontrados com as informações de contato.

Deve ser notado que, para determinar as especialidades utilizadas no processo de recomendação, o ICARE utiliza uma Ontologia [Antoniou, et al., 2008] que é responsável por relacionar termos e conceitos dos domínios de conhecimento e por prover propriedade semântica aos mesmos. Com base nessa ontologia é possível descobrir, a partir da palavra digitada pelo usuário, os conceitos relacionados. Por exemplo, no caso do usuário procurar pela palavra “Redes”, vários conceitos estão associados, como redes neurais e redes de computadores. Assim são retornados os conceitos mais frequentes, baseando-se nos pesos de cada um deles. A partir da especialidade, buscam-se os especialistas que possuem essa habilidade e que posteriormente são ordenados de acordo com os demais atributos contextuais analisados. Por fim uma lista ordenada de especialistas é gerada e apresentada para o usuário.

6.2. WebAPSEE-Planner

A proposta do WebAPSEE-Planner, segundo Silva [Silva, 2007], é proporcionar um mecanismo de geração de soluções de alocação de pessoas para realização de atividades de um projeto de *software*. As alocações são feitas com base em políticas definidas pelos usuários através de uma linguagem própria definida por Lima Reis [Lima Reis, 2003]. O WebAPSEE-Planner foi projetado para ser um módulo integrado a uma outra ferramenta

chamada WebAPSEE [Lima, et al., 2006]. O WebAPSEE, por sua vez, é um Ambiente de Engenharia de *Software* Centrado em Processo (PSEE) que tem como principal objetivo o auxílio à modelagem e execução de processos de desenvolvimento de *software* [Silva, 2007].

No WebAPSEE-Planner, o gerente de projetos deve criar políticas que definem como ocorrerá as alocações das pessoas às tarefas baseando-se em atributos de ambos. É provida uma gramática com a linguagem que deve ser utilizada para a construção das políticas. As políticas, além de serem regidas por esta gramática, possuem uma série de atributos próprios tais como: condições para que a política seja adotada, critérios de restrição e critérios de ordenação.

As condições de adoção de políticas são criadas para indicar qual política é mais representativa segundo a atual situação da organização, já que estas políticas são instanciadas dinamicamente ao decorrer do processo. Assim, o WebAPSEE pode utilizar sempre a política mais adequada para cada situação. Nas condições de adoção de políticas são levados em consideração os cargos que devem ser preenchidos, os agentes e as habilidades dos mesmos.

Quando uma política é selecionada, são adotados os critérios de restrição definidos pelo usuário. Os critérios de restrição são utilizados para que o processo de seleção dos recursos seja refinado, diminuindo o espaço de busca por agentes compatíveis a serem alocados. Alguns dos critérios de restrição são custo e carga de trabalho, onde para cada atributo é passado um critério de comparação (<, >, =, <=, >=, <>) e um valor a ser comparado. Outro atributo que pode ser comparado é a experiência dos agentes. Todavia, é analisado para esse atributo apenas se o agente possui ou não experiência em determinada tecnologia, sem mensurar esse determinado nível de experiência.

Após os agentes terem sido selecionados, os mesmos são ordenados segundo os critérios de ordenação definidos pelo usuário. Alguns desses critérios de ordenação são, por exemplo, menor custo, maior custo, menor carga de trabalho, maior habilidade e maior afinidade (medida que representa quão bem dois agentes trabalham juntos).

Após esse processo, é gerada uma lista com os agentes mais indicados para serem alocados aos cargos requeridos.

6.3. Apoio à alocação de RH baseado em satisfação de restrições

A proposta de Barreto [Barreto, 2005] é a criação de um sistema de apoio à alocação de pessoas num projeto de *software* utilizando como base para solução problemas de satisfação

de restrição. Um problema de satisfação de restrição é composto por três partes: um conjunto de variáveis, um conjunto de domínios e um conjunto de restrições. Nesse sentido, um problema de satisfação de restrição é solucionado através da atribuição de valores (contidos num domínio) a todas as variáveis, de forma que as restrições definidas sejam atendidas.

Nessa abordagem, são considerados como atributos de um profissional sua habilidade, conhecimento, experiência, formação acadêmica, certificações, etc. Para que um profissional seja passível de alocação, cada um desses atributos deve corresponder ao conjunto de características exigidas por cada atividade. Além disso, é levado em consideração fatores como o custo de cada profissional, períodos de disponibilidade do mesmo, bem como o período em que as atividades devem ser realizadas.

Com base nesse conjunto de atributos, são definidas as seguintes regras para que um profissional possa ser alocado a uma atividade:

- Cada profissional precisa possuir todas as características exigidas pela atividade em intensidade igual ou superior ao exigido;
- O profissional deve estar disponível no período em que a atividade será executada;
- Um profissional está indisponível quando: i) Houver alguma indisponibilidade registrada para o período da atividade (férias, licença, etc.); ii) estiver alocado em outra atividade que sobreponha o período da atividade analisada.

Baseando-se no conjunto de regras supracitadas, as soluções propostas para o problema de alocação contemplam quatro possibilidades diferentes. Cada possibilidade gera uma resposta diferente ao problema de alocação. Contudo, elas são fixas e não podem ser refinadas diretamente pelo gerente de projetos. Todavia, o gerente pode definir quais restrições devem ser satisfeitas ou não segundo o conjunto de restrições disponíveis pela abordagem. As quatro possibilidades de solução são:

- *N soluções qualquer* – retorna uma quantidade N de soluções que satisfaçam as restrições;
 - *Problema ótimo* – solução que satisfaz as restrições e minimiza o custo, dado algum fator (menor equipe, menor tempo, melhor qualificação, etc.);
 - *Problema parcial ótimo quanto às restrições* – soluções parciais que infligam o menor número de restrições;
-

- *Problema parcial ótimo quanto às atividades* – soluções parciais que atribuam profissionais ao maior número de atividades distintas.

Como discutido pelo autor, a abordagem proposta visa à diminuição dos casos possíveis a ser considerados numa alocação de pessoas às atividades de um projeto de *software*, visto que podem existir um grande número de pessoas e atividades envolvidas nesse processo. Desse modo, é visto que se forem definidas restrições, apenas aquelas soluções que atendam às restrições se tornam válidas e, portanto, o número de comparações a se fazer para a escolha da solução mais adequada se torna menor e mais simples.

6.4. Task Allocation Optimizer

Duggan [Duggan, et al., 2004] apresenta uma ferramenta que se utiliza de otimização multi-objetivo para gerar soluções de alocação de profissionais para tarefas de implementação de um *software*. A temática multi-objetivo é utilizada, pois segundo os autores, é muito difícil chegar a apenas uma solução que seja a melhor. Desse modo, segundo os autores, esta se torna a melhor forma de interação homem máquina num sistema de recomendação, já que o sistema pode chegar a soluções que um humano possa não pensar, mas oferece um leque de opções boas para que o gerente, utilizando seus conhecimentos, escolha a melhor para o contexto de seu problema.

A ferramenta considera como um módulo de implementação pacotes de *software*. Os pacotes de *software* são detalhados em termos de sua complexidade segundo a medida de complexidade ciclomática de Thomas J. McCabe [McCabe, 1976], que é a agregação de complexidade das classes que compõem o pacote. Após detalhados os pacotes, o gerente de projetos descreve cada profissional em termos de proficiência em relação a todos os pacotes. A descrição de competência é feita utilizando os termos: *novice*, *average*, *good*, *very good* e *expert*, contudo o usuário da aplicação pode redefinir os termos.

O algoritmo multi-objetivo adotado no trabalho é um Algoritmo Genético Elitista [Sivanandam, et al. 2001]. Soluções são geradas seguindo os operadores de Algoritmos Genéticos de mutações e cruzamentos. Assim que geradas, cada solução é comparada par a par, procurando soluções dominantes e dominadas. Uma solução dominante é dita como uma solução que domina outra solução em todos os aspectos, ou seja, por ser multi-objetivo, uma solução que seja melhor que outra em todos os objetivos. As soluções dominadas são então desprezadas e o conjunto de soluções restantes é chamado de conjunto *Pareto-Optimal*, onde nenhuma solução domina outra.

O conjunto de soluções é então oferecido ao gerente que pode visualizá-lo segundo os vários objetivos, verificando quais soluções são melhores em cada aspecto. Com isso o gerente de projetos pode escolher a solução que melhor se adéque às suas necessidades.

6.5. MRES

O MRES [Callegari, et al., 2009] é uma ferramenta de apoio à decisão para os gerentes de projeto quanto à alocação de um profissional a uma atividade de um projeto de *software*. Essa ferramenta baseia-se na utilização de um algoritmo de Lógica *Fuzzy* para inferir as soluções. Diferentemente das demais abordagens, que consideram várias características tanto do profissional quanto da atividade a ser desempenhada, esta abordagem considera apenas as habilidades em alguma tecnologia esperada para o desenvolvimento de uma atividade e o nível atual desta habilidade atribuído ao recurso humano.

Uma das tarefas do usuário da ferramenta MRES é a definição de quais termos linguísticos utilizar para a caracterização dos níveis de habilidades requeridas pelas atividades, dos níveis apresentados pelos recursos e da adequabilidade da seleção. A ferramenta não oferece suporte para o gerente em como calcular a capacidade técnica de um recurso referente a alguma habilidade. É deixado para que o gerente o faça.

Os termos linguísticos são então dispostos em uma tabela que cruza os níveis de recursos e atividades. Com o preenchimento desta tabela, são definidas as regras que devem ser utilizadas para gerar as soluções de alocação. Assim que todos os recursos forem avaliados para uma atividade, é gerada uma lista ordenada pelo valor de aptidão encontrado na avaliação. A partir desta lista o gerente de projetos pode escolher qual recurso alocar para esta atividade.

Um ponto importante abordado neste trabalho, e pouco citado pelos demais, é a questão de retroalimentação dos valores das habilidades definidos para os recursos. Sabe-se que com o passar do tempo as pessoas podem aprimorar seus conhecimentos com o uso ou apresentarem uma queda do mesmo devido à falta de prática. Essa mudança de estado dos atributos de caracterização é contemplada pela ferramenta, onde o próprio gerente de projetos ajusta os valores de acordo com sua impressão acerca do desempenho dos recursos nas atividades as quais os mesmo foram alocados.

6.6. Multi-Criteria Task Assignment in Workflow Management Systems

O trabalho de Shen [Shen, et al., 2003] consiste em dar suporte ao *workflow* de um projeto de *software* consistindo, portanto, num suporte para um Sistema de Gerenciamento de *Workflow* (WfMS). WfMS geralmente modelam a alocação de tarefas de um projeto para papéis a serem desempenhados e não à profissionais específicos. Com isso cada papel fica atrelado a um conjunto de tarefas tornando mais fácil o processo de manutenção das definições do *workflow*. Contudo diferentes trabalhadores com diferentes habilidades podem compartilhar um mesmo papel. Assim este trabalho tem o intuito de selecionar instâncias (profissionais reais) para realizarem as tarefas associadas aos respectivos papéis no *workflow* do projeto.

A abordagem baseia a seleção dos profissionais em três critérios: nas capacidades técnicas dos profissionais, na relação social entre os profissionais e na relação entre tarefas a serem desempenhadas. Para mensurar as habilidades dos profissionais, a importância da tarefa e o *score* da avaliação são utilizados respectivamente os conjuntos *fuzzy*: *no*, *low*, *normal*, *good* e *expert*; *not important*, *barely important*, *moderately important*, *very important* e *extremely important*; e *worst*, *poor*, *fair*, *good*, e *best*. Estes conjuntos são fixos e não há menção se podem ser modificados pelo usuário da aplicação. Também não é mencionado sobre o conjunto de regras *fuzzy* utilizado para fazer a seleção dos profissionais, assim aparentemente não há uma configuração sobre políticas de seleção que podem ser utilizadas.

A seleção dos profissionais ocorre da seguinte forma. Primeiramente é verificado se existem tarefas conflitantes que podem ser alocadas ao mesmo profissional ou se uma nova tarefa pode ser conflitante com uma já associada a algum profissional. Um exemplo de tarefas conflitantes seriam tarefas como contabilidade e auditoria. Caso haja tarefas conflitantes para algum profissional, este é eliminado da avaliação. Os candidatos restantes então são avaliados quanto suas capacidades técnicas referente às tarefas a serem desempenhadas. Depois é verificado se a carga de trabalho dos mesmos permite que a associação com novas tarefas seja feita. Os profissionais são então avaliados segundo suas relações sociais com o intuito de selecionar aqueles que possuem melhor interação social entre si. Novamente não há menção de como esse valor é capturado ou gerado. Por fim, os candidatos são ordenados em grupos, gerando assim o *score* final de cada possível grupo a ser alocado.

6.7. Considerações Finais

Dos trabalhos apresentados neste capítulo, o ICARE é o único que não possui a finalidade de seleção de um profissional às atividades de *software*. Ele é um sistema de recomendação de

especialistas para auxiliar outra pessoa em alguma atividade. Ainda assim, por ser um SRE, ele contempla uma etapa contida nos demais trabalhos que é a etapa de caracterização dos recursos ou especialistas. São considerados vários atributos que fazem uma caracterização adequada de características que um especialista deve possuir e, portanto, tem relevância direta com o presente trabalho proposto.

O ICARE conta também com uma ontologia de domínio que não está presente em nenhum dos outros trabalhos. Tal ontologia agrega grande valor às etapas de caracterização e avaliação dos especialistas, pois, a partir dela, os conhecimentos e habilidades requeridas passam a apresentar semântica e relacionamentos com outros conceitos que são importantes ao se avaliar um especialista. Assim, o processo de avaliação se torna mais eficaz e mais condizente com a realidade.

Quanto ao processo de avaliação dos recursos, o WebAPSEE-Planner apresenta uma característica bastante relevante, que é a possibilidade de permitir ao gerente de projetos definir políticas que devem ser utilizadas para avaliar quais recursos são adequados para qual atividade. A definição de políticas é interessante, pois possibilita ao gerente de projeto um maior controle no processo de seleção, utilizando sua expertise e agregando conhecimento, inclusive acerca de informações não tratadas pela abordagem. Assim refinando e ajustando as avaliações de seleção de acordo com as necessidades específicas de seu projeto.

A proposta de apoio à alocação com base em satisfação de restrições apresentada por Barreto [Barreto, 2005] é, por sua vez, menos flexível quanto às alterações por parte do gerente de projetos. Contudo é permitido ao gerente escolher quais restrições devem ser ou não satisfeitas para que se possam selecionar os recursos no processo de avaliação. Entretanto, até pela própria natureza de resolução através do problema de satisfação de restrição, esta abordagem se preocupa em diminuir o número de soluções possíveis, tornando o processo de avaliação mais fácil, rápido e eficiente. Além disso, são apresentadas quatro possibilidades de alocações possíveis, onde uma delas retorna uma solução ótima dado um fator a ser selecionado (menor custo, menor equipe, melhor qualificação, melhor tempo, etc.). Apesar de ser chamada “solução ótima”, ela é uma solução onde todas as restrições são satisfeitas, mas o gerente deve escolher qual atributo deve ser “otimizado”, seja o custo, o número de pessoas, o tempo, etc. Assim, ela é uma solução ótima para um atributo, mas não na média de todos os atributos, já que é necessário apenas que os demais valores satisfaçam as restrições. Ainda assim, esse tipo de solução é muito mais eficiente do que uma solução dada apenas pela experiência e intuição do gerente de projetos.

As propostas de Dugan [Duggan, et al., 2004] e Barreto [Barreto, 2005] apresentam soluções ótimas dado um atributo em particular, possibilitando portanto que o gerente possua diferentes visões acerca do resultado de seleção, cada um com um foco em um objetivo específico. Apesar da utilização dos termos vagos *novice*, *average*, *good*, *very good* e *expert* serem utilizados para definir a proficiência dos profissionais quanto aos pacotes de *software*, esses termos são discretos não *fuzzy*. Como não representam conjuntos *fuzzy*, a proficiência de um profissional está sempre representada apenas por um único termo, sem graus de pertinência, nem a possibilidade de estar associado a mais de um termo. Desse modo, o problema da subjetividade da representação do conhecimento permanece neste trabalho. Do mesmo modo que os demais trabalhos, a medição do conhecimento dos profissionais é deixada a cargo do gerente de projetos.

A ferramenta MRES permite ao gerente de projetos uma boa flexibilidade quanto ao refinamento da avaliação de soluções de seleção, já que é o próprio gerente quem escolhe os termos linguísticos a serem usados para os recursos, para as atividades e para o nível de aptidão. Também é permitido ao gerente a definição das regras de seleção a serem utilizadas. Desse modo, o gerente possui grande controle de como funcionará o processo de avaliação. Idealmente, poderiam ser sugeridos por padrão termos linguísticos e regras que satisfaçam os casos mais utilizados num projeto de *software* ou até mesmo as políticas mais referenciadas na literatura, mas Callegari [Callegari, et al., 2009] não comenta acerca dessa questão em seu trabalho.

Um ponto fraco do MRES é que, diferente dos demais trabalhos, apenas um tipo de atributo, para o recurso e para a atividade, é analisado na avaliação de solução de recomendação. Esse atributo é referente à habilidade em uma tecnologia que o recurso possui e a habilidade requerida nessa tecnologia pela atividade. Apesar do atributo referente à habilidade em uma tecnologia ser de grande importância, se não o de maior importância técnica, outros atributos poderiam ser considerados na avaliação. Considere um grande número de recursos a serem avaliados. É provável que muitos dos recursos apresentem o mesmo nível de conhecimento em uma tecnologia. Assim, quando avaliados, serão gerados grandes grupos de muitos recursos com o mesmo nível de conhecimento, deixando o gerente de projetos com um leque ainda muito grande de soluções para se escolher a mais satisfatória. Caso fossem considerados outros atributos, ou uma maneira mais refinada de se obter estes atributos, soluções mais refinadas poderiam ser geradas, permitindo uma maior facilidade na seleção dos recursos por parte do gerente.

A proposta de Shen [Shen, et al., 2003] faz a seleção seguindo uma lógica multi-objetiva. Dessa forma, dentre os trabalhos apresentados, este parece ser o que leva em consideração o maior número de fatores numa seleção, tais como: atividades conflitantes, atividades já alocadas, conhecimento técnico e interação social entre os participantes, bem como o número de participantes nos grupos formados pela ferramenta. Além disso, ela utiliza a lógica *fuzzy* como parte do “raciocínio” da seleção e também como forma de representar e mensurar os conhecimentos dos profissionais, portanto levando em consideração a subjetividade dessa atividade. Contudo, os conjuntos *fuzzy* são fixos, bem como as regras *fuzzy* utilizadas na seleção. Isso faz com que a ferramenta não tenha tanta flexibilidade para ser adaptada em projeto de diferentes contextos.

A característica de maior destaque dos trabalhos citados neste capítulo é apresentada por Callegari [Callegari, et al., 2009] e Shen [Shen, et al., 2003] que corresponde à utilização de Lógica *Fuzzy* e seus termos linguísticos associados. As demais abordagens utilizam valores discretos para os atributos de caracterização dos profissionais e das atividades. Como foi dito, termos discretos não conseguem capturar bem a realidade de algo incerto ou nebuloso [Cox, 1995], como o nível de conhecimento de um terceiro acerca de uma determinada tecnologia, por exemplo. Então, apesar das regras de avaliações possuírem boa lógica, se os atributos utilizados para disparar as regras e realizar as avaliações não condizem com a realidade do cenário, as regras podem se tornar comprometidas e não produzem bons resultados. Partindo dessa premissa, fica fácil notar que a utilização de termos difusos é uma ótima opção para tratar do cenário de seleção, que é o foco principal destes trabalhos.

A Tabela 53 sintetiza as principais características dos trabalhos citados, realizando assim uma comparação entre eles e o trabalho proposto. Analisando as características levantadas, percebe-se que todas as abordagens apresentam um conjunto de atributos para caracterização do conhecimento. É esperado que todas as abordagens apresentem essa característica, pois esta se apresenta como o principal fator para a caracterização do conhecimento de uma pessoa. Contudo, apesar de levantar atributos para caracterizar o conhecimento, apenas a abordagem de seleção proposta provê uma forma de calcular o conhecimento a partir desses atributos. Não obstante a esse fato, esta abordagem ainda se apresenta como a única, dentre as mencionadas, capaz de gerar o conhecimento de uma equipe como um todo para que se possa realizar um processo de seleção de equipes e não só de pessoas. Quanto ao fato de proporcionar políticas configuráveis, apenas as abordagens ICARE, Apoio à alocação de RH baseado em satisfação de restrições e *Task Allocation*

Optimizer não o apresentam. Como explicado no Capítulo 4, termos vagos *fuzzy* se apresentam como uma ferramenta capaz de representar satisfatoriamente a característica de conhecimento de uma pessoa. Nesse sentido apenas as abordagens MRES, *Multi-Criteria Task Assignment in Workflow Management Systems* e a abordagem proposta utilizam-se de lógica *fuzzy* para essa caracterização. Por fim, foi levantado no Capítulo 4 a importância de uma retroalimentação na abordagem. Nesse sentido apenas o MRES e abordagem de seleção proposta apresentam tal característica.

Tabela 53. Comparação dos trabalhos relacionados

	Conjunto de atributos para caracterização do conhecimento	Forma de calcular o conhecimento das pessoas	Seleção de equipes	Políticas configuráveis para seleção	Termos vagos <i>fuzzy</i>	<i>Feedback</i> das informações
ICARE	X					
WebAPSEE-Planner	X			X		
Apoio à alocação de RH baseado em satisfação de restrições	X					
Task Allocation Optimizer	X					
MRES	X			X	X	X
Multi-Criteria Task Assignment in Workflow Management Systems	X			X	X	
Abordagem proposta	X	X	X	X	X	X

Capítulo 7

Considerações Finais

Neste trabalho, foi apresentada uma abordagem para a seleção de equipes distribuídas tecnicamente qualificadas para implementação de módulos de *software* num projeto de *software*. Essa abordagem foi estruturada em etapas distintas que visam coletar os dados necessários para descrição das tecnologias a serem utilizadas para construção dos módulos de *software*, coleta de dados das equipes acerca dos conhecimentos necessários para implementação dos módulos de *software*, construção de regras para seleção das equipes mais adequadas e a seleção efetiva destas equipes. Desse modo, para aplicar a abordagem proposta, é preciso fazer contato com vários profissionais desempenhando diferentes papéis. Primeiro, deve-se realizar contato com o arquiteto de *software* de modo a construir a chamada tabela de implementação, que contém os requisitos necessários para o desenvolvimento dos módulos. Em seguida, deve-se entrar em contato com as equipes de desenvolvimento. O intuito desse contato é levantar os conhecimentos das equipes acerca das tecnologias necessárias para o desenvolvimento de cada módulo de *software*. Visando uma melhor adaptação ao contexto do projeto, ao gerente de projetos é permitida a configuração de políticas de seleção que regem os critérios que guiam como a seleção deve ser efetuada. Por fim, quando todas as informações são coletadas, a abordagem faz um cruzamento entre equipes e módulos, e, seguindo a política definida, é capaz de gerar recomendações que indicam quais equipes são mais adequadas tecnicamente para o desenvolvimento de cada módulo.

Um dos principais problemas encontrados nesse contexto é a forma de representação e quantificação do conhecimento das equipes. Como já foi mencionado, esse tipo de informação é altamente subjetiva, o que torna difícil uma quantificação apropriada. Além disso, no contexto proposto, as equipes de desenvolvimento nem sempre estão próximas fisicamente ao centro da organização, já que as mesmas podem estar distribuídas. Assim, coletar informações dessas equipes apropriadamente se torna ainda mais difícil, já que não se pode realizar entrevistas presenciais com profissionais experientes que consigam medir o conhecimento destas equipes. Devido a este fato foi adotado o uso de questionários para coleta de dados das equipes. Contudo, como nessa abordagem as próprias equipes informam

sobre seus conhecimentos, se tornaria complicado pedir para que elas mensurassem seu próprio conhecimento, pois agregaria ainda mais subjetividade já que o que uma equipe considera como bom conhecimento, outra equipe poderia considerar como médio conhecimento. Assim, os questionários foram desenvolvidos para que as perguntas geradas retornassem respostas mais concretas e mensuráveis sobre a experiência das equipes. Ainda assim, para contar com eventuais descompassos entre o conhecimento estimado das equipes e o conhecimento real das mesmas, foi proposto um atributo que considera as impressões do gerente de projetos quanto ao desempenho das equipes na prática, o atributo *reputação*.

A fim de testar a abordagem e demonstrar sua eficiência, foram desenvolvidos dois estudos de caso (E-Commerce e Multi-MOM), os quais foram apresentados no Capítulo 5. Como foram gerados vários resultados, apenas alguns foram apresentados e discutidos. Como resultado, percebe-se que a abordagem consegue seguir aquilo o que foi proposto:

- Uma maneira de calcular o conhecimento de uma pessoa acerca de uma tecnologia;
- Uma maneira de calcular o conhecimento de uma equipe acerca de uma tecnologia de acordo com os conhecimentos dos membros dessa equipe;
- Uma maneira de criar regras personalizadas para a seleção de equipes que atendam às necessidades específicas de projetos distintos;
- Uma forma de selecionar equipes através de políticas configuráveis.

7.1. Contribuições

A contribuição inicial da abordagem é a proposição de uma forma de mensurar as habilidades de equipes distribuídas lidando com toda a subjetividade inserida nessa atividade. Outros trabalhos de seleção de equipes, como os apresentados no Capítulo 6, admitem que esta tarefa é deixada para o gerente resolver e tratam apenas da seleção em si. No entanto, como foi visto no Capítulo 2, a maioria dos processos de gerenciamento de RH afirmam que se deve identificar as competências dos recursos antes de selecioná-los. Assim, em relação a outros trabalhos, a abordagem apresentada nesta dissertação se torna mais completa já que além de tratar da avaliação da seleção, promove uma maneira de identificar as competências das equipes, uma atividade complexa que é geralmente deixada a cargo do gerente de projetos.

Além disso, foi desenvolvido um algoritmo baseado em Lógica *Fuzzy* que consegue lidar com toda imprecisão e subjetividade inserida no contexto do problema, sendo capaz de gerar os resultados satisfatórios de recomendação das equipes. O que deve ser notado quanto a

essa abordagem é a possibilidade de configuração pelo gerente de projetos. Isso torna a abordagem flexível e adaptável para diferentes contextos. Ainda, a configuração é uma opção, não uma obrigação, já que configurações padrão foram definidas e disponibilizadas para uso. As configurações mais relevantes que podem ser realizadas dizem respeito às políticas de seleção. Desse modo, o gerente de projetos pode adotar critérios diferentes de seleção para diferentes projetos ou até mesmo para diferentes iterações de um mesmo projeto. Além de terem sido propostas quatro políticas de seleção, a abordagem ainda permite que estas sejam refinadas ou que novas políticas sejam criadas de acordo com as necessidades específicas de cada projeto de *software*.

Além de recomendar quais são as equipes mais adequadas para desenvolver cada módulo de *software* de acordo com a política de seleção utilizada, a abordagem apresenta uma métrica indicando quão adequada cada equipe é. Isso é interessante, pois consegue agregar um maior valor ao processo de seleção já que com essa medida o gerente de projetos pode realizar comparações entre as equipes selecionadas ou até entre projetos distintos nas quais elas participaram. Finalmente, para viabilizar estas contribuições, a abordagem foi definida de forma a possuir etapas distintas com entradas e saídas bem definidas, visando à separação de objetivos e responsabilidades diferentes. Tal separação permite que sejam inseridas novas etapas na abordagem caso seja necessário.

7.2. Limitações

Além das contribuições, algumas limitações foram observadas quanto à abordagem proposta. Primeiramente, a maneira encontrada para mensurar o conhecimento ficou destinado à representação das habilidades das equipes. Todavia, este mesmo conhecimento deve ser calculado no que diz respeito aos módulos de *softwares*, onde as tecnologias que serão utilizadas também devem ser mensuradas quanto à complexidade esperada de implementação dos módulos. Apesar da relevância, a abordagem apresentada não provê nenhuma forma de calcular tal complexidade, deixando a cargo do arquiteto e/ou gerente de projetos que o façam de acordo com sua própria experiência.

Outra limitação da abordagem é que ela está restrita a selecionar as equipes apenas observando sua capacidade técnica. Num projeto real de *software*, outros fatores também devem ser levados em consideração, como o custo de cada equipe, disponibilidade e a carga de trabalho que a mesma pode suportar. Esses fatores não foram incluídos inicialmente na

proposta do trabalho, pois este trabalho está inserido num *framework* maior de alocação onde esses fatores são considerados em outras fases do mesmo.

Apesar de ser considerado como algo bom, a possibilidade de permitir ao gerente de projetos alterar as configurações presentes na abordagem pode se apresentar como um problema ou dificuldade. Caso o mesmo não tenha conhecimento preciso sobre as consequências das modificações dos valores dos atributos configuráveis, a abordagem pode passar a ser prejudicial, pois irá gerar resultados não satisfatórios ou até mesmo errados. Seguindo nessa mesma linha, outra limitação é que não foi realizado um estudo mais profundo sobre como otimizar os conjuntos *fuzzy* e os termos utilizados durante todo o processo. Foram utilizados quatro termos para representar os conhecimentos da equipe e três termos para representar a complexidade das tecnologias. Além disso, a configuração do formato e pontos de cada conjunto foi feita de maneira a fazer uma distribuição semelhante entre as funções. Portanto, não se garante que esta configuração é uma configuração ótima, mesmo porque isso pode variar de projeto para projeto. Assim, a mudança dos termos fica exclusivamente dependente da experiência do gerente de projetos.

Quanto aos estudos de casos, houve uma limitação para validação da abordagem proposta, já que não foi possível validá-la num ambiente real de desenvolvimento de *software* do início ao fim de um projeto. Para ambos os estudos de casos não existiam equipes distribuídas reais de desenvolvimento. Também, por não ter havido de fato a implementação dos módulos pelas equipes selecionadas, não houve como verificar na prática o desempenho das equipes selecionadas.

Por fim, apesar de ter sido implementado um protótipo para validação da abordagem proposta de seleção de equipes, não foi concebida uma interface gráfica para entrada e saída dos dados sobre equipes, módulos e políticas. Assim, todas as configurações e entradas necessárias para a avaliação são feitas diretamente no código fonte do protótipo.

7.3. Trabalhos Futuros

Considerando as limitações percebidas, primeiramente como trabalho futuro é planejado a construção de uma ferramenta com interface gráfica para que se possa fazer uso prático da abordagem proposta num projeto de *software* real. A ferramenta deve possuir um bom suporte gráfico para as configurações das políticas de seleção e da configuração dos conjuntos *fuzzy*.

Com a ferramenta pronta, é planejado utilizá-la num projeto real mais complexo onde a abordagem poderá ser validada com mais propriedades e pode-se fazer uma avaliação mais completa e robusta das contribuições práticas da abordagem desenvolvida.

Para melhorar a abordagem, deve ser feita uma análise mais aprofundada sobre como calcular e mensurar a complexidade esperada das tecnologias a serem empregadas pelas equipes para desenvolver os módulos, repassando para a ferramenta a maior parte do trabalho que é feita atualmente pelo gerente de projetos e arquitetos de *software*. Nessa mesma direção, também se planeja uma análise sobre a representação do conhecimento das equipes de forma a otimizar os formatos e pontos de cada conjunto *fuzzy* associado.

Verificou-se também que o ponto de corte utilizado como padrão (0.7) deve ser mais bem estudado. Isso foi evidenciado nos estudos de casos, onde se pode perceber uma variação média dos resultados obtidos de acordo com a política selecionada. Dessa forma, pode-se questionar se o valor 0.7 é um bom ponto de corte para todas as políticas de seleção. Assim, deve ser realizado um estudo que averigue a correlação entre a configuração de uma política, o ponto de corte, e a adequabilidade das equipes selecionadas.

Referências

- [ABNT, 2000] NBR ISO 10006 Gestão da Qualidade – *Diretrizes para a Qualidade no Gerenciamento de Projetos*. Associação Brasileira de Normas Técnicas, Rio de Janeiro, RJ. 2000.
- [Acuña 2003] ACUÑA, S. T.; JURISTO, N. *Modelling Human Competencies in the Software Process*. ProSim'03, Portland, Estados Unidos.
- [Antoniou, 2008] ANTONIOU G., HARMELEN F. 2008 *A Semantic Web Primer* - London: The MIT Press. Vol. 2.
- [Barreto, 2005] BARRETO A. S. *Apoio à Alocação de Recursos Humanos em Projetos de Software: Uma Abordagem Baseada em Satisfação de Restrições* Dissertação de Mestrado - UFRJ/COPPE.
- [Basili et. al., 2001] BASILI, V., LINDVALL, M., COSTA, P. Implementing the Experience Factory concepts as a set of Experience Bases. In: *Software Engineering and Knowledge Engineering* - SEKE 2001, Buenos Aires, Argentina, June.
- [Bezerra, 2010] BEZERRA, Y. *Multi-MOM: Um Middleware Multi-Paradigma Extensível e Orientado a Mensagens para Computação Móvel*. Universidade Federal da Paraíba. João Pessoa, 2010.
- [Callegari, et al., 2009] CALLEGARI D. A., FOLIATTI F. L., BASTOS R. M. MRES – Ferramenta para seleção de recursos para tarefas de projetos de software via abordagem difusa e multicritérios. *Simpósio Brasileiro de Engenharia de Software (SBES) - Sessão de Ferramentas*. Fortaleza 2009.
- [Carmel, 1999] CARMEL, E. 1999 *Global software teams: collaborating across borders and time zones* - Upper Saddle River Prentice Hall PTR. Vol. 1.
- [Cataldo, 2008] CATALDO M., HERBSLEB D. Communication Networks in Geographically Distributed Software Development. *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. - San Diego : [s.n.], 2008. - pp. 579-588.
- [Chaves 2008] CHAVES A. P., et al. Um modelo baseado em context-awareness para Disseminação de Informações em um Ambiente de Desenvolvimento Distribuído de Software XXXIV *Conferencia Latinoamericana de informática (CLEI)*. Santa Fe.
- [Clements, et al., 2002] CLEMENTS P., NORTHROP L., *Software Product Lines: Practices and Patterns*. Boston: Addison-Wesley, 2002.
- [Collofello, et al., 1998] COLLOFELLO J. [et al.] A System Dynamics Software Process Simulator for Staffing Policies Decision Support. *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences*. - Kohala Coast: [s.n.], 1998. - pp. 103-111.
-

- [Cox, 1995] COX E. D. 1995. *Fuzzy logic for business and industry* - Rockland. Charles River Media.
- [Dey, et;al 2001] DEY, A. K., ABOWD, G. D. *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*, Human-Computer Interaction
- [Dingsoyr et. al. 2001] DINGSOYR, T.; ROYRVIK, E. Skills Management as Knowledge Technology in a Software Consultancy Company. *In Proceedings of the Third international Workshop on Advances in Learning Software Organizations* (September 12 - 13, 2001). K. Althoff, R. L. Feldmann, and W. Müller, Eds. Lecture Notes In Computer Science, vol. 2176. Springer-Verlag, London, 96-105.
- [Duggan, et al., 2004] DUGGAN J., BYRNE J., LYONS G. *A Task Allocation Optimizer for Software Construction*. IEEE Software. - Los Alamitos: IEEE Computer Society Press, 2004. - 3: Vol. 21.
- [Ebert, et al., 2001] EBERT C., NEVE P. D. *Surviving Global Software Development*. IEEE Software. - Los Alamitos: IEEE Computer Society Press, 2001. - 2 : Vol. 18.
- [Gomaa 2004] GOMAA, H. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Object-Oriented Technology Series, ISBN: 0201775956.
- [Gumm, 2006] GUMM D. C. *Distribution Dimensions in Software Development Projects: A Taxonomy*. IEEE Software. - Los Alamitos, Vol. 23. Issue 5 pp. 45-51.
- [Herbsleb, et al., 2006] HERBSLEB J. D., MOCKUS A., ROBERTS J. A. Collaboration in Software Engineering Projects: A Theory of Coordination. *Proceedings International Conference on Information Systems (ICIS)*. - Milwaukee: [s.n.], 2006.
- [Hellendoorn et. al. 1993] Hellendoorn, H., Thomas, C. *Defuzzification in fuzzy controlees*. J. Intell. Fuzzy. Syst., p109-123.
- [Herbsleb, et al., 2001] HERBSLEB J. D., MOITRA, D. *Global Software Development*. IEEE Software. - 2001. - 2 : Vol. 18. - pp. 16-20.
- [Herbsleb, 2007] HERBSLEB J. D. 2007. *Global Software Engineering: The Future of Socio-technical Coordination*. International Conference on Software Engineering (ICSE).
- [Hunt, M. J, 2006] HUNT M. J., MCGREGOR J. D. *Software product lines: a pedagogical application.*: Consortium for Computing Sciences in Colleges, 2006. - 2 : Vol. 22. - pp. 295-302.
- [Jantzen, 1998] JANTZEN J. *Design Of Fuzzy Controllers*, Tech.reportno98-E864
- [Karolak D. W., 99] KAROLAK D. W. *Global Software Development - Managing Virtual Teams and Enviroments*. IEEE Computer Society, 1999.
- [Kagdi, et al., 2008] KAGDI H., HAMMAD M., MALETIC J. I. Who can help me with this source code change?. *IEEE International Conference on Software Maintenance*. - Beijing: [s.n.], 2008. - pp. 157-166.
-

[Kosko, 1992] KOSKO, B. *Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence*. 1992. Prentice Hall, Englewood Cliffs, NJ.

[Lima, et al., 2006] LIMA A., et al. *Gerência Flexível de Processos de Software com o Ambiente WebAPSEE. 19º Simpósio Brasileiro de Engenharia de Software – Sessão de Ferramentas*.

[Lima Reis, 2003] LIMA REIS C. A. *Uma Abordagem Flexível para Execução de Processos de Software Evolutivos*. Tese de Doutorado - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul,.

[Linden, 2007] LINDEN F. J., SCHMID K., ROMMES E., 2007. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering* - New York. Springer-Verlag, 2007. Vol. 1.

[Linden, 2008] LINDEN R. 2008. *Algoritmos genéticos: Uma importante ferramenta da inteligência computacional* - Rio de Janeiro. Brasport, Vol. 2.

[Marquardt, et al., 2001] MARQUARDT M. J., HORVATH L. *Global Teams: how top multinationals span boundaries and cultures with high-speed teamwork*. Palo Alto: Davies-Black, 2001.

[Moraes et. al. 2010] MORAIS, Y., ELIAS, G. Integrating Communication Paradigms in a Mobile Middleware Product Line, *Proceedings of the 9th International Conference on Networks (ICN)*, Menuires, França.

[McCabe, 1976] McCABE T.J. *A Complexity Measure*. IEEE Trans. Software Eng. 2(4). - pp. 308-320.

[Paulish, 2003] PAULISH, D.J. *Product Line Engineering for Global Development*. International Workshop on Product Line Engineering: The Early Steps.

[Pereira, et al., 2010] PEREIRA, T. A. B., SANTOS, V. S., RIBEIRO, B. L., ELIAS, G. A Recommendation Framework for Allocating Global Software Teams in Software Product Line Projects. *Proceedings of 2nd International Workshop on Recommendation Systems for Software Engineering (RSSE 2010)*.

[Pereira, 2011] PEREIRA, T. *Uma Abordagem para Recomendação de Módulos para Projetos de Desenvolvimento Distribuído de Linhas de Produto de Software*. Universidade Federal da Paraíba (UFPB). João Pessoa. 2011

[Petry, 2007] PETRY H., *ICARE: Um sistema de recomendação de especialistas sensível a contexto*. Tese de Mestrado. UFPE.

[Plekhanova, 1999] PLEKHANOVA, V. Capability and Compatibility Measurement in Software Process Improvement. *Proceedings of the 2nd European Software Measurement Conference – FESMA’99*. Amsterdam, The Netherlands.

[PMI, 2004] PMI PMBOK - *Project Management Body of Knowledge: Project Management Institute* - Vol. 3ª Edição

[Pohl, et al., 2005] POHL, K., BCKLE, G., VAN DER LINDEN, F. J. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag, New York, USA.

[Pressman, 2004] PRESSMAN R. S. 2004. *Software Engineering: A Practitioner's Approach*. New York. McGraw-Hill, Vol. 6.

[Prikladnicki, 2008] PRIKLADNICKI R., AUDY J. L. Gestão Quantitativa de Pessoas em DDS: primeira aplicação de um modelo para o cálculo da distância percebida relativa em equipes distribuídas de desenvolvimento de software. *II Workshop de Desenvolvimento Distribuído de Software (WDDS)*. - Campinas

[Ramasubbu, 2005] RAMASUBBU N., KRISHNAN M. S., KOMPALLI P. *Leveraging Global Resources: A Process Maturity Framework for Managing Distributed Development*. IEEE Software. Vol. 22. Issue 3 - pp. 80-86.

[Reichling, 2005] REICHLING T., SCHUBERT K., WULF V. *Matching Human Actors Based on their Texts: Design and Evaluation of an Instance of the ExpertFinding Framework*. ACM SIGGROUP Conference on Supporting Group Work. - New York pp. 61-70.

[Ribeiro, 2013] RIBEIRO, B. *Uma Abordagem para Alocação de Equipes em Projetos de Desenvolvimento Distribuído de Software com base em Atributos Não-Técnicos*. Universidade Federal da Paraíba (UFPB). João Pessoa. 2013

[Ross, 2004] ROSS T. J. 2004. *Fuzzy Logic With Engineering Application*. University of New Mexico, USA. John & Sons Ltd, Second Edition.

[Ruhe, 2003] RUHE G. *Software Engineering Decision Support*. ICORE Research Report. - 2003. - Vol. 2. - pp. 153-162.

[Rus, et al., 2002] RUS I., BIFFL S., HALLING M., Systematically Combining Process Simulation and Empirical Data in Support of Decision Analysis in Software Development. *Proceeding of the SEKE'02*,. - Ischia : [s.n.], 2002. - pp. 827-833.

[Ruskova, 2002] RUSKOVA N. A. Decision support system for human resources appraisal and selection. *First International IEEE Symposium "Intelligent Systems"*. - 2002. - Vol. 1. - pp. 354-357.

[Santos, et. al., 2010] SANTOS, V. S. ; BURITY, T. ; RIBEIRO, B. L. ; ELIAS, G. . Um Framework de Recomendação para Alocação de Equipes de Desenvolvimento em Projetos Distribuídos de Linhas de Produto de Software. *In: IV Workshop de Desenvolvimento Distribuído de Software, 2010, Salvador. Congresso Brasileiro de Software: Teoria e Prática (CBSOFT)*.

[Santos, et. al., 2011] SANTOS, V.; ELIAS, G. Uma Abordagem Baseada em Lógica Nebulosa para Seleção de Equipes Distribuídas Tecnicamente Qualificadas. *V Workshop de Desenvolvimento Distribuído de Software (WDDS)*. SãoPaulo: [s.n.]. 2011

[Schnaider, 2003] SCHNAIDER, L.R.C. *Planejamento da alocação de recursos humanos em ambientes de desenvolvimento de software orientados à organização*. Dissertação de Mestrado – Programa de Pós-Graduação em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

[Shanteau et. al, 2002] SHANTEAU J., et al., Performance-based assessment of expertise: How to decide if someone is an expert or not. *European Journal of Operational Research*. Elsevier, Vol. 136.- pp. 253-263.

[Shen, et. al. 2003] M. Shen, G.-H, Tzeng, and D.-R. Liu, “Multi-Criteria Task Assignment in Workflow Management Systems” *36th Annual Hawaii International Conference on System Sciences*, pp 1-9

[Silva, 2007] SILVA M. A. *WebAPSEE-Planner: Auxílio à Alocação de Pessoas em Projetos de Software Através de Políticas*. Tese de Mestrado. UNIVERSIDADE FEDERAL DO PARÁ

[Sivanandam et. al. 2001] SIVANANDAM S. N., DEEPA S. N. *Introduction to Genetic Algorithms*. Springer

[Standish Group, 2009] STANDISH GROUP. The Chaos Report 2009. Online: http://www1.standishgroup.com/newsroom/chaos_2009.php, verificado em Março de 2010.

[Sangwan R. et al., 2006] SANGWAN R. et al *Global Software Development Handbook* -: Taylor & Francis Group, 2006.

[Weiss et. al. 2006] WEISS D. J., SHANTEAU J., HARRIES P., *People who judge people*, Journal of Behavioral Decision Making, 19, pp. 441-454.

[Wastell et. al. 1999] WASTELL, D., ARBAOUI, S., LONCHAMP J., MONTANGERO, C. The Human Dimension of the Software Process. *Software Process: Principles, Methodology, Technology*, Springer.

[Wohlin, 2004] WOHLIN C. *Are Individual Differences in Software Development Performance Possible to Capture Using a Quantitative Survey?* Empirical Software Engineering.- Vol. 9. pp. 211-228.

[Zadeh L. A., 1965] ZADEH L. A., “Fuzzy Sets”, *Information and Control*, Vol. 8, Issue 3, 1965, pp 338-353

APÊNDICE A

-

QUESTIONÁRIO DE TECNOLOGIAS UTILIZADO PARA FORMAÇÃO DAS EQUIPES

Conhecimento: JAVA

Há quantos anos você desenvolve projetos em JAVA? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5--7
- ☐ 7--9
- ☐ 9 ou mais

Quantos projetos você desenvolveu utilizando a linguagem JAVA? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20--25
- ☐ 25--30
- ☐ 30 ou mais

Quais dessas certificações você possui em JAVA?

- ☐ SCJA
 - ☐ SCJP
 - ☐ SCJD
-

- ☐ SCWCD
- ☐ SCBCD
- ☐ SCDJWC
- ☐ SCEA
- ☐ SMAD

Conhecimento: Eclipse

Há quantos anos você trabalha com o Eclipse? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu utilizando Eclipse? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15 ou mais

Conhecimento: JSF (Java Server Faces)

Há quantos anos você desenvolve projetos com JSF? *

- ☐ Nenhum
 - ☐ 1--3
 - ☐ 3--5
 - ☐ 5 ou mais
-

Quantos projetos você desenvolveu utilizando JSF? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Conhecimento: Sockets

Há quantos anos você trabalha com Sockets? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu utilizando Sockets? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Conhecimento: Hibernate

Há quantos anos você utiliza o Hibernate? *

- ☐ Nenhum
 - ☐ 1--3
-

- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu utilizando Hibernate? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Conhecimento: Programação Reflexiva

Há quantos anos você utiliza programação reflexiva? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu utilizando programação reflexiva? *

- ☐ Nenhum
 - ☐ 1--5
 - ☐ 5--10
 - ☐ 10--15
 - ☐ 15--20
 - ☐ 20 ou mais
-

Conhecimento: SVN

Há quantos anos você utiliza SVN? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu utilizando SVN? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Conhecimento: SQL

Há quantos anos você utiliza SQL? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5--7
- ☐ 7--9
- ☐ 9 ou mais

Quantos projetos você desenvolveu utilizando SQL? *

- ☐ Nenhum
 - ☐ 1--5
-

- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Quais certificações ou especializações você possui relacionadas à SQL (Certificações MySQL, Oracle, SQL Server, etc..) ? Separe por vírgula (,)



Conhecimento: Android

Há quantos anos você desenvolve para plataforma Andorid? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu utilizando Android? *

- ☐ Nenhum
 - ☐ 1--5
 - ☐ 5--10
 - ☐ 10--15
 - ☐ 15--20
 - ☐ 20 ou mais
-

Quais certificações ou especializações você possui relacionadas à plataforma Andorid ? Separe por vírgula (,)



Conhecimento: Agentes de Software

Há quantos anos você utiliza Agentes de Software? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5--7
- ☐ 7 ou mais

Quantos projetos você desenvolveu utilizando Agentes de Software? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Conhecimento: Redes de computadores (protocolos)

Conhecimento e familiaridade na utilização de recursos como i) protocolos de descoberta dinâmica de serviços (UPnP, SLP, etc) ii) protocolos de transporte de rede (TCP, UDP)

Há quantos anos você desenvolve projetos que necessitam de protocolos de redes? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5--7
- ☐ 7 ou mais

Quantos projetos você desenvolveu que necessitam de protocolos de redes? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Quais certificações ou especializações você possui relacionadas ao conhecimento acerca de protocolos de redes? Separe por vírgula (,)



Conhecimento: Redes de computadores (paradigmas)

Conhecimento e familiaridade na utilização de recursos como i) redes sem fio (bluetooth, wifi) ii) paradigmas de comunicação (filas, espaço de tuplas, publisher/subscriber, etc)

Há quantos anos você desenvolve projetos que necessitam de algum paradigma de comunicação de redes? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5--7
- ☐ 7 ou mais

Quantos projetos você desenvolveu que necessitam de algum paradigma de comunicação de redes? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Quais certificações ou especializações você possui relacionadas ao conhecimento acerca de paradigmas de redes? Separe por vírgula (,)



Conhecimento: SCRUM

Há quantos anos você utiliza SCRUM? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5--7
- ☐ 7 ou mais

Quantos projetos você desenvolveu utilizando SCRUM? *

- ☐ Nenhum
- ☐ 1--5
- ☐ 5--10
- ☐ 10--15
- ☐ 15--20
- ☐ 20 ou mais

Quais dessas certificações SCRUM você possui?

- ☐ Certified Scrum Product Owner
- ☐ Certified Scrum Master
- ☐ Certified Scrum Practitioner
- ☐ Certified Scrum Coach
- ☐ Certified Scrum Trainer

Conhecimento: Transações Bancárias WEB (E-Commerce)

Conhecimento e familiaridade na implementação de projeto que se utilizam de Transações Bancárias na WEB (E-Commerce)

Há quantos anos você desenvolve software que se utilizam de transações bancárias na web? *

- ☐ Nenhum
- ☐ 1--3
- ☐ 3--5
- ☐ 5 ou mais

Quantos projetos você desenvolveu que utilizam transações bancárias na web? *

- ☐ Nenhum
 - ☐ 1--5
 - ☐ 5--10
 - ☐ 10--15
 - ☐ 15--20
 - ☐ 20 ou mais
-

APÊNDICE B

-

RESPOSTAS DO QUESTIONÁRIO PARA O CONJUNTO A DE PARTICIPANTES

	JAVA	Eclipse	JSF	Sockets	Hibernate
Equipe 1					
Indivíduo 1	0,2333	0,5833	0,0000	0,0000	0,5833
Indivíduo 2	0,8250	1,0000	0,7500	0,0000	1,0000
Indivíduo 3	0,8250	1,0000	0,0000	0,3333	0,0000
Indivíduo 4	0,5667	0,7500	0,5833	0,5833	0,5833
Indivíduo 5	0,8000	1,0000	0,0000	0,8333	1,0000
Indivíduo 6	0,6667	0,8333	0,5833	0,0000	1,0000
Indivíduo 7	0,2333	0,5833	0,0000	0,0000	0,0000
Indivíduo 8	0,4833	0,7500	0,5833	0,0000	0,5833
Indivíduo 9	0,2333	0,0000	0,0000	0,5833	0,0000
Equipe 2					
Indivíduo 10	0,2333	0,5833	0,5833	0,0000	0,5833
Indivíduo 11	0,5667	1,0000	0,5833	0,5833	0,5833
Indivíduo 12	0,5667	0,5833	0,0000	0,5833	0,5833
Indivíduo 13	0,6667	1,0000	0,5833	0,0000	0,5833
Indivíduo 14	0,3667	0,8333	0,5833	0,5833	0,8333
Equipe 3					
Indivíduo 15	0,5917	0,8333	0,5833	0,0000	0,5833
Indivíduo 16	0,8000	1,0000	0,0000	0,8333	0,8333
Indivíduo 17	0,3667	0,8333	0,0000	0,0000	0,5833
Indivíduo 18	0,6917	1,0000	0,5833	0,5833	1,0000
Indivíduo 19	0,2333	0,0000	0,5833	0,0000	0,5833
Indivíduo 20	0,2333	0,5833	0,5833	0,0000	0,5833
Indivíduo 21	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 22	0,9500	1,0000	0,7500	0,7500	1,0000
Indivíduo 23	0,5917	1,0000	0,5833	0,5833	0,5833
Indivíduo 24	0,3667	0,8333	0,0000	0,5833	0,5833
Indivíduo 25	0,3917	0,5833	0,5833	0,0000	0,5833
Indivíduo 26	0,0000	0,7500	0,0000	1,0000	0,0000
Indivíduo 27	0,4333	0,7500	0,0000	0,2500	0,2500
Indivíduo 28	0,3667	0,5833	0,5833	0,5833	0,5833
Indivíduo 29	0,2333	0,5833	0,5833	0,0000	0,5833
Indivíduo 30	0,7000	0,7500	0,0000	0,0000	0,5833

Indivíduo 31	0,8000	1,0000	0,0000	0,8333	0,0000
Indivíduo 32	0,5667	1,0000	0,0000	1,0000	0,5833
Equipe 4					
Indivíduo 33	0,2333	0,5833	0,0000	0,5833	0,0000
Indivíduo 34	0,2333	0,5833	0,5833	1,0000	0,5833
Indivíduo 35	0,2333	1,0000	0,5833	0,0000	0,5833
Indivíduo 36	0,6667	0,5833	0,0000	0,5833	0,0000
Indivíduo 37	0,7500	1,0000	0,0000	0,8333	0,8333
Indivíduo 38	0,5000	0,8333	0,5833	0,5833	0,8333
Indivíduo 39	0,3667	0,8333	0,5833	0,5833	0,5833
Indivíduo 40	0,7250	1,0000	0,0000	0,0000	0,0000
Equipe 5					
Indivíduo 41	0,2333	0,5833	0,0000	0,0000	0,5833
Indivíduo 42	0,6667	1,0000	0,5833	0,0000	1,0000
Indivíduo 43	0,8000	1,0000	0,3333	0,8333	1,0000
Indivíduo 44	0,6667	1,0000	0,0000	0,8333	0,0000
Indivíduo 45	0,5667	1,0000	0,0000	0,0000	0,0000
Indivíduo 46	0,8750	1,0000	0,0000	0,8333	0,5833
Indivíduo 47	0,2333	0,5833	0,5833	0,0000	0,0000
Indivíduo 48	0,2333	0,5833	0,0000	0,5833	0,0000
Indivíduo 49	0,2333	0,5833	0,0000	0,0000	0,0000
Equipe 6					
Indivíduo 50	0,6167	0,8333	0,8333	0,0000	1,0000
Indivíduo 51	0,7167	1,0000	1,0000	0,5833	0,7500
Equipe 7					
Indivíduo 52	0,7250	1,0000	0,8333	0,0000	0,8333
Indivíduo 53	0,8500	1,0000	0,0000	0,5833	0,5833
Indivíduo 54	0,2333	0,8333	0,0000	0,0000	0,0000
Indivíduo 55	0,3917	0,8333	0,0000	0,0000	0,0000
Indivíduo 56	0,7500	1,0000	0,8333	0,0000	0,8333

	Programação Reflexiva	SVN	SQL	Android	Agentes de Software
Equipe 1					
Indivíduo 1	0,0000	0,5833	0,3167	0,0000	0,0000
Indivíduo 2	0,5833	1,0000	1,0000	0,0000	0,0000
Indivíduo 3	1,0000	0,5833	0,4833	0,0000	0,0000
Indivíduo 4	0,5833	0,5833	0,5167	0,0000	0,0000
Indivíduo 5	1,0000	0,8333	1,0000	0,0000	0,0000
Indivíduo 6	0,5833	1,0000	0,8333	0,0000	0,0000
Indivíduo 7	0,0000	0,5833	0,0000	0,0000	0,0000

Indivíduo 8	0,0000	1,0000	0,5167	0,3750	0,5833
Indivíduo 9	0,0000	0,5833	0,3167	0,0000	0,0000
Equipe 2					
Indivíduo 10	0,0000	0,7500	0,5167	0,0000	0,0000
Indivíduo 11	0,5833	0,5833	0,6833	0,0000	0,0000
Indivíduo 12	0,5833	1,0000	0,6833	0,0000	0,0000
Indivíduo 13	0,5833	0,5833	0,4833	0,0000	0,0000
Indivíduo 14	0,0000	0,5833	0,4833	0,0000	0,0000
Equipe 3					
Indivíduo 15	0,5833	0,5833	0,6833	0,0000	0,5833
Indivíduo 16	0,0000	1,0000	0,4833	0,3750	0,0000
Indivíduo 17	0,0000	0,5833	0,3167	0,0000	0,0000
Indivíduo 18	0,5833	1,0000	1,0000	0,0000	0,5833
Indivíduo 19	0,0000	0,0000	0,4833	0,0000	0,0000
Indivíduo 20	0,0000	1,0000	0,6833	0,0000	0,0000
Indivíduo 21	0,0000	0,5833	1,0000	0,0000	0,0000
Indivíduo 22	1,0000	1,0000	1,0000	0,1250	1,0000
Indivíduo 23	0,5833	1,0000	0,3167	0,0000	0,5833
Indivíduo 24	0,5833	1,0000	0,3167	0,0000	0,0000
Indivíduo 25	0,0000	0,5833	0,4833	0,0000	0,0000
Indivíduo 26	1,0000	0,7500	1,0000	0,0000	0,0000
Indivíduo 27	0,0000	0,2500	0,5167	0,2500	0,0000
Indivíduo 28	0,0000	1,0000	0,8500	0,0000	0,0000
Indivíduo 29	0,5833	0,5833	1,0000	0,0000	0,0000
Indivíduo 30	0,0000	0,5833	0,4833	0,0000	0,0000
Indivíduo 31	0,0000	0,8333	0,8500	0,0000	0,0000
Indivíduo 32	0,0000	1,0000	0,3167	0,0000	0,0000
Equipe 4					
Indivíduo 33	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 34	1,0000	1,0000	0,8500	0,0000	0,0000
Indivíduo 35	0,0000	0,5833	0,3167	0,0000	0,0000
Indivíduo 36	0,0000	0,5833	0,3167	0,0000	0,0000
Indivíduo 37	0,5833	0,8333	0,8500	0,3750	0,0000
Indivíduo 38	0,0000	0,5833	0,4833	0,0000	0,0000
Indivíduo 39	0,0000	0,8333	0,8500	0,0000	0,0000
Indivíduo 40	0,0000	0,0000	1,0000	0,0000	0,0000
Equipe 5					
Indivíduo 41	0,0000	0,0000	0,3167	0,0000	0,0000
Indivíduo 42	0,0000	0,8333	0,8333	0,0000	0,0000
Indivíduo 43	0,8333	1,0000	1,0000	0,0000	0,0000
Indivíduo 44	0,5833	0,7500	0,4833	0,0000	0,5833

Indivíduo 45	0,0000	1,0000	0,3167	0,0000	0,0000
Indivíduo 46	0,8333	1,0000	1,0000	0,8750	0,0000
Indivíduo 47	0,0000	0,5833	0,3167	0,0000	0,0000
Indivíduo 48	0,0000	1,0000	1,0000	0,0000	0,0000
Indivíduo 49	0,0000	0,5833	0,3167	0,0000	0,0000
Equipe 6					
Indivíduo 50	0,5833	1,0000	0,8333	0,0000	0,5833
Indivíduo 51	0,7500	0,7500	0,8333	0,0000	0,5833
Equipe 7					
Indivíduo 52	0,5833	1,0000	0,6500	0,0000	0,0000
Indivíduo 53	0,8333	0,5833	1,0000	0,3750	0,0000
Indivíduo 54	0,0000	0,5833	0,6833	0,0000	0,0000
Indivíduo 55	0,5833	0,2500	0,3167	0,0000	0,0000
Indivíduo 56	0,0000	0,8333	0,6500	0,0000	0,0000

	Protocolos de Rede	Paradigmas de Comunicação	SCRUM	Transações Bancárias
Equipe 1				
Indivíduo 1	0,5833	0,0000	0,4667	0,0000
Indivíduo 2	1,0000	0,0000	0,8000	0,0000
Indivíduo 3	0,0000	0,0000	0,4667	0,0000
Indivíduo 4	0,0000	0,0000	0,0000	0,0000
Indivíduo 5	0,8333	1,0000	0,0000	0,0000
Indivíduo 6	0,5833	0,0000	0,4667	0,0000
Indivíduo 7	0,2500	0,0000	0,4667	0,0000
Indivíduo 8	0,0000	0,0000	0,4667	0,5833
Indivíduo 9	0,5833	0,7500	0,0000	1,0000
Equipe 2				
Indivíduo 10	0,0000	0,0000	0,4667	0,0000
Indivíduo 11	0,5833	0,5833	0,0000	0,0000
Indivíduo 12	0,5833	0,0000	0,0000	0,0000
Indivíduo 13	0,5833	0,0000	0,0000	0,0000
Indivíduo 14	0,8333	0,0000	0,0000	0,0000
Equipe 3				
Indivíduo 15	0,5833	0,5833	0,4667	0,0000
Indivíduo 16	0,0000	0,5833	0,4667	0,0000
Indivíduo 17	0,0000	0,5833	0,4667	0,0000
Indivíduo 18	0,8333	0,5833	0,4667	0,0000
Indivíduo 19	0,8333	0,5833	0,0000	0,0000
Indivíduo 20	0,5833	1,0000	0,2000	0,5833
Indivíduo 21	0,0000	0,0000	0,4667	0,0000

Indivíduo 22	1,0000	0,2500	0,4667	0,5833
Indivíduo 23	0,0000	0,2500	0,2000	0,0000
Indivíduo 24	0,5833	0,5833	0,0000	0,0000
Indivíduo 25	0,0000	0,5833	0,0000	0,0000
Indivíduo 26	0,0000	1,0000	0,2000	0,0000
Indivíduo 27	0,0000	0,0000	0,0000	0,0000
Indivíduo 28	0,5833	0,5833	0,4667	0,0000
Indivíduo 29	0,5833	0,0000	0,0000	0,0000
Indivíduo 30	0,5833	0,0000	0,0000	0,0000
Indivíduo 31	0,8333	0,0000	0,0000	0,0000
Indivíduo 32	0,8333	1,0000	0,4667	0,0000
Equipe 4				
Indivíduo 33	0,0000	0,0000	0,0000	0,0000
Indivíduo 34	0,5833	0,5833	0,6667	0,5833
Indivíduo 35	0,0000	0,0000	0,4667	0,0000
Indivíduo 36	0,5833	0,0000	0,0000	0,0000
Indivíduo 37	0,8333	0,5833	0,0000	0,0000
Indivíduo 38	0,8333	0,5833	0,2667	0,0000
Indivíduo 39	0,8333	0,8333	0,0000	0,0000
Indivíduo 40	1,0000	0,0000	0,0000	0,0000
Equipe 5				
Indivíduo 41	0,0000	0,0000	0,0000	0,0000
Indivíduo 42	0,0000	0,0000	0,0000	0,0000
Indivíduo 43	0,0000	0,8333	0,2000	0,5833
Indivíduo 44	0,5833	0,0000	0,0000	0,0000
Indivíduo 45	0,5833	0,2500	0,0000	0,0000
Indivíduo 46	1,0000	0,8333	0,6000	0,0000
Indivíduo 47	0,5833	0,0000	0,0000	0,5833
Indivíduo 48	0,0000	0,0000	0,0000	0,5833
Indivíduo 49	0,0000	0,5833	0,0000	0,0000
Equipe 6				
Indivíduo 50	0,0000	0,0000	0,2000	0,5833
Indivíduo 51	1,0000	0,5833	0,6000	0,0000
Equipe 7				
Indivíduo 52	0,0000	0,5833	0,4667	0,0000
Indivíduo 53	1,0000	0,5833	0,4667	0,0000
Indivíduo 54	0,5833	0,0000	0,0000	0,0000
Indivíduo 55	0,5000	0,0000	0,0000	0,0000
Indivíduo 56	0,0000	0,0000	0,0000	0,0000

APÊNDICE C

-

CONECHIMENTO CALCULADO POR TECNOLOGIA PARA O CONJUNTO B DE PARTICIPANTES

	JAVA	Eclipse	JSF	Sockets	Hibernate
Equipe 1					
Indivíduo 120	0,8000	1,0000	0,8000	1,0000	0,6000
Indivíduo 56	0,4880	0,8000	0,2650	0,5300	0,5300
Indivíduo 94	0,1360	0,2650	0,2650	0,1000	0,3650
Equipe2					
Indivíduo 4	0,4800	0,4300	0,0000	0,2650	0,2650
Indivíduo 20	0,2720	0,8000	0,4300	0,4300	0,7000
Indivíduo 87	0,5370	0,7000	0,5300	0,1650	0,7000
Equipe3					
Indivíduo 117	0,4080	0,5300	0,4300	0,2650	0,7000
Indivíduo 100	0,3220	0,5300	0,2650	0,2650	0,2650
Indivíduo 82	0,2960	0,4300	0,1650	0,2650	0,4300
Equipe 4					
Indivíduo 36	0,2720	0,6300	0,0000	0,4300	0,2650
Indivíduo 88	0,5130	0,8000	0,3650	0,0000	0,3650
Indivíduo 59	0,1360	0,2650	0,3650	0,2650	0,2650
Indivíduo 16	0,6400	1,0000	0,2650	1,0000	0,7000
Indivíduo 27	0,0000	0,1650	0,0000	0,0000	0,0000
Equipe 5					
Indivíduo 111	0,4080	0,8000	0,2650	0,2650	0,2650
Indivíduo 26	0,1360	0,4300	0,0000	0,0000	0,0000
Indivíduo 103	0,4320	0,7000	0,2650	1,0000	0,0000
Indivíduo 68	0,1360	0,3650	0,2650	0,0000	0,6300
Indivíduo 35	0,1360	0,3650	0,2650	0,2650	0,2650
Equipe 6					
Indivíduo 105	0,4080	0,8000	0,5300	0,0000	0,2650
Indivíduo 39	0,2480	0,4650	0,2650	0,1000	0,1000
Indivíduo 123	0,8750	1,0000	0,8000	0,6000	1,0000
Indivíduo 18	0,2160	0,2650	0,0000	0,2650	0,0000
Indivíduo 95	0,3520	0,7000	0,2650	0,0000	0,7000
Equipe 7					
Indivíduo 101	0,4560	0,6000	0,6000	0,5000	0,6000
Indivíduo 54	0,0560	0,2650	0,0000	0,4300	0,2650

Indivíduo 115	0,3120	0,6000	0,2650	0,8000	0,6000
Indivíduo 49	0,4320	0,7000	0,2650	0,0000	0,0000
Indivíduo 8	0,1360	0,2650	0,0000	0,1650	0,1650
Indivíduo 70	0,0560	0,4300	0,0000	0,0000	0,0000
Indivíduo 57	0,2720	0,4300	0,2650	0,0000	0,1650
Indivíduo 83	0,1360	0,2650	0,0000	0,0000	0,2650
Equipe 8					
Indivíduo 112	0,7400	0,7000	0,5300	0,1650	0,7000
Indivíduo 71	0,1360	0,2650	0,0000	0,0000	0,2650
Indivíduo 97	0,5370	0,7000	0,5300	0,1650	0,7000
Indivíduo 51	0,2720	0,5300	0,2650	0,2650	0,2650
Indivíduo 80	0,1360	0,2650	0,2650	0,0000	0,2650
Indivíduo 73	0,3520	0,7000	0,4300	0,2650	0,2650
Indivíduo 53	0,4680	1,0000	0,5300	0,0000	0,5300
Indivíduo 106	0,4320	0,8000	0,5300	0,6000	0,6000
Equipe 9					
Indivíduo 6	0,1360	0,2650	0,0000	0,0000	0,0000
Indivíduo 77	0,4880	0,8000	0,0000	0,2650	0,5300
Indivíduo 43	0,5680	0,8000	0,2650	0,0000	0,5300
Indivíduo 67	0,0560	0,1000	0,1000	0,0000	0,2650
Indivíduo 2	0,4080	0,8000	0,2650	0,2650	0,2650
Indivíduo 30	0,0560	0,1000	0,1000	0,0000	0,1000
Indivíduo 19	0,5680	0,8000	0,4300	0,6000	0,7000
Indivíduo 61	0,2480	0,4650	0,2650	0,0000	0,1650
Equipe 10					
Indivíduo 45	0,3080	0,6300	0,2650	0,0000	0,2650
Indivíduo 110	0,0560	0,2650	0,8000	0,0000	0,2650
Indivíduo 31	0,1360	0,4650	0,0000	0,0000	0,2650
Indivíduo 64	0,2720	0,5300	0,0000	0,0000	0,0000
Indivíduo 91	0,1360	0,2650	0,2650	0,2650	0,2650
Indivíduo 24	0,3080	0,5650	0,0000	0,2650	0,0000
Indivíduo 58	0,2720	0,7000	0,0000	0,0000	0,0000
Indivíduo 3	0,0560	0,2650	0,0000	0,1000	0,0000
Indivíduo 7	0,1360	0,7300	0,0000	0,0000	0,0000
Indivíduo 38	0,1360	0,4300	0,1650	0,6300	0,2650
Equipe 11					
Indivíduo 69	0,4080	0,2650	0,2650	0,2650	0,0000
Indivíduo 40	0,1920	0,3650	0,2650	0,0000	0,2650
Indivíduo 121	0,4080	0,8000	0,5300	0,6000	0,6000
Indivíduo 5	0,1920	0,2650	0,2650	0,0000	0,2650
Indivíduo 102	0,8000	1,0000	0,8000	1,0000	0,6000

Indivíduo 44	0,3080	0,6650	0,0000	0,3650	0,0000
Indivíduo 115	0,0560	0,4300	0,8000	0,0000	0,6000
Indivíduo 25	0,1360	0,4300	0,2650	0,0000	0,1650
Indivíduo 78	0,1360	0,3650	0,1000	0,1000	0,2650
Indivíduo 89	0,2160	0,5300	0,0000	0,4300	0,0000
Equipe 12					
Indivíduo 37	0,1360	0,2650	0,3650	0,0000	0,5300
Indivíduo 109	0,4000	0,2600	0,2600	0,2600	0,0000
Indivíduo 104	0,2720	0,7000	0,0000	0,0000	0,0000
Indivíduo 32	0,8750	1,0000	0,8000	0,6000	1,0000
Indivíduo 1	0,0560	0,9000	0,0000	0,0000	0,0000
Indivíduo 15	0,2970	0,5300	0,0000	0,2650	0,0000
Indivíduo 10	0,3080	0,6650	0,0000	0,3650	0,0000
Indivíduo 98	0,8000	0,7000	0,5300	0,4300	0,7000
Indivíduo 74	0,2720	0,5300	0,0000	0,0000	0,2650
Indivíduo 55	0,1360	0,4300	0,0000	0,0000	0,0000
Equipe 13					
Indivíduo 92	0,5600	0,7300	0,2650	0,0000	0,6300
Indivíduo 28	0,1920	0,1650	0,2650	0,1650	0,2650
Indivíduo 122	0,3520	0,7000	0,4300	0,2650	0,2650
Indivíduo 62	0,0560	0,4300	0,2650	0,2650	0,0000
Indivíduo 119	0,4560	0,6000	0,6000	0,5000	0,6000
Indivíduo 13	0,4320	0,7000	0,2650	0,0000	0,4300
Indivíduo 17	0,0560	0,0000	0,0000	0,4300	0,0000
Indivíduo 46	0,1360	0,2650	0,0000	0,2650	0,2650
Indivíduo 118	0,2480	0,4650	0,2650	0,1000	0,1000
Indivíduo 75	0,1360	0,2650	0,0000	0,0000	0,0000
Indivíduo 34	0,0000	0,2650	0,0000	0,0000	0,0000
Indivíduo 86	0,2720	0,3650	0,0000	0,2650	0,3650
Indivíduo 23	0,2160	0,4300	0,2650	0,0000	0,2650
Indivíduo 33	0,4000	0,2600	0,2600	0,2600	0,0000
Indivíduo 72	0,2720	0,5300	0,4300	0,2650	0,7000
Equipe 14					
Indivíduo 90	0,2720	0,8000	0,2650	0,2650	0,3650
Indivíduo 29	0,4560	0,6000	0,6000	0,5000	0,6000
Indivíduo 96	0,1920	1,0000	0,0000	0,1000	0,0000
Indivíduo 21	0,2720	0,5300	0,0000	0,0000	0,0000
Indivíduo 50	0,0560	0,8000	0,0000	0,5300	0,2650
Indivíduo 11	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 14	0,4080	0,8000	0,5300	0,0000	0,2650
Indivíduo 113	0,3760	0,6000	0,2650	0,2650	0,6000

Indivíduo 85	0,4320	0,8000	0,0000	0,6000	0,6000
Indivíduo 41	0,1360	0,7300	0,0000	0,0000	0,0000
Indivíduo 52	0,0560	0,6000	0,0000	0,0000	0,0000
Indivíduo 99	0,0560	0,4300	0,2650	0,2650	0,0000
Indivíduo 107	0,5370	0,7000	0,5300	0,1650	0,7000
Indivíduo 66	0,0560	0,4300	0,8000	0,0000	0,6000
Indivíduo 60	0,0000	0,0000	0,0000	0,0000	0,0000
Equipe 15					
Indivíduo 22	0,1360	0,2650	0,0000	0,0000	0,0000
Indivíduo 116	0,3080	0,6650	0,0000	0,3650	0,0000
Indivíduo 93	0,0800	0,4300	0,1650	0,0000	0,1650
Indivíduo 9	0,3760	0,6000	0,2650	0,2650	0,6000
Indivíduo 12	0,0560	0,2650	0,0000	0,8000	0,0000
Indivíduo 114	0,5680	0,8000	0,4300	0,6000	0,7000
Indivíduo 108	0,8750	1,0000	0,8000	0,6000	1,0000
Indivíduo 84	0,7400	1,0000	0,0000	0,6000	0,0000
Indivíduo 48	0,3520	0,7000	0,5300	0,2650	0,5300
Indivíduo 42	0,1920	0,3650	0,2650	0,0000	0,2650
Indivíduo 63	0,1360	0,4300	0,2650	0,0000	0,0000
Indivíduo 47	0,0560	0,2650	0,8000	0,0000	0,2650
Indivíduo 65	0,1920	0,3650	0,2650	0,1650	0,2650
Indivíduo 76	0,8000	1,0000	0,8000	1,0000	0,6000
Indivíduo 81	0,3220	0,5300	0,2650	0,2650	0,2650

	Programação Reflexiva	SVN	SQL	Android	Agentes de Software
Equipe 1					
Indivíduo 120	1,0000	1,0000	0,8000	0,2120	0,2250
Indivíduo 56	0,4300	1,0000	0,7200	0,2120	0,4750
Indivíduo 94	0,2650	0,2650	0,2400	0,2120	0,2250
Equipe2					
Indivíduo 4	0,2650	0,2650	0,1600	0,3440	0,0000
Indivíduo 20	0,0000	0,8000	0,4800	0,0000	0,2250
Indivíduo 87	0,4300	0,4300	0,5600	0,0000	0,0000
Equipe3					
Indivíduo 117	0,4300	0,8000	0,4800	0,3920	0,1250
Indivíduo 100	0,0000	0,4300	0,3200	0,0000	0,1250
Indivíduo 82	0,0000	0,4300	0,3200	0,0000	0,0000
Equipe 4					
Indivíduo 36	0,2650	0,2650	0,5600	0,2120	0,0000
Indivíduo 88	0,2650	0,2650	0,6400	0,1320	0,2250

Indivíduo 59	0,4300	0,6000	0,0000	0,2120	0,2250
Indivíduo 16	0,7000	1,0000	0,8000	0,0000	0,0000
Indivíduo 27	0,0000	0,0000	0,0000	0,0000	0,0000
Equipe 5					
Indivíduo 111	0,0000	0,6300	0,4800	0,0000	0,0000
Indivíduo 26	0,0000	0,2650	0,2400	0,0000	0,0000
Indivíduo 103	0,2650	0,0000	0,0000	0,3920	0,2250
Indivíduo 68	0,2650	0,4300	0,2400	0,0000	0,0000
Indivíduo 35	0,0000	0,0000	0,3200	0,2120	0,2250
Equipe 6					
Indivíduo 105	1,0000	0,5300	0,4000	0,0000	0,0000
Indivíduo 39	0,0000	0,5300	0,1600	0,2120	0,1000
Indivíduo 123	0,2650	0,9000	0,8000	0,0000	0,2250
Indivíduo 18	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 95	0,2650	0,4300	0,4000	0,2120	0,1250
Equipe 7					
Indivíduo 101	0,0000	0,6000	0,4800	0,0000	0,0000
Indivíduo 54	0,2650	0,0000	0,7200	0,0000	0,0000
Indivíduo 115	0,9000	1,0000	0,8000	0,2120	0,2250
Indivíduo 49	0,2650	0,0000	0,0000	0,0000	0,0000
Indivíduo 8	0,0000	0,3650	0,5600	0,0000	0,0000
Indivíduo 70	0,0000	0,2650	0,1600	0,0000	0,0000
Indivíduo 57	0,5300	0,2650	0,3200	0,0800	0,0000
Indivíduo 83	0,2650	0,4300	0,4800	0,0000	0,2250
Equipe 8					
Indivíduo 112	0,4300	0,4300	0,5600	0,2120	0,3800
Indivíduo 71	0,0000	0,2650	0,4800	0,2120	0,0000
Indivíduo 97	0,4300	0,4300	0,5600	0,2120	0,3800
Indivíduo 51	0,1650	0,0000	0,0000	0,0000	0,0000
Indivíduo 80	0,0000	0,0000	0,1600	0,3920	0,0000
Indivíduo 73	0,0000	0,2650	0,2400	0,0000	0,0000
Indivíduo 53	0,5300	0,0000	0,7200	0,1650	0,0000
Indivíduo 106	0,4300	0,7000	0,5600	0,4750	0,3800
Equipe 9					
Indivíduo 6	0,0000	0,2650	0,1600	0,0000	0,1250
Indivíduo 77	0,5300	0,6000	0,3200	0,0000	0,2250
Indivíduo 43	0,5300	0,0000	0,4800	0,2120	0,1000
Indivíduo 67	0,0000	0,1000	0,1600	0,0000	0,0000
Indivíduo 2	0,5300	0,6300	0,4800	0,0000	0,0000
Indivíduo 30	0,0000	0,1000	0,0800	0,0000	0,1250
Indivíduo 19	0,4300	0,9000	0,5600	0,0000	0,3500

Indivíduo 61	0,2650	0,2650	0,4800	0,1650	0,0000
Equipe 10					
Indivíduo 45	0,1650	0,2650	0,3200	0,0000	0,1000
Indivíduo 110	0,2650	0,5300	0,6400	0,2920	0,3500
Indivíduo 31	0,0000	0,0000	0,2400	0,0000	0,0000
Indivíduo 64	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 91	0,0000	0,3650	0,2800	0,0000	0,2250
Indivíduo 24	0,0000	0,2650	0,2400	0,0000	0,0000
Indivíduo 58	0,1650	0,0000	0,5600	0,1650	0,0000
Indivíduo 3	0,0000	0,2650	0,2400	0,0000	0,0000
Indivíduo 7	0,0000	0,2650	0,5600	0,2120	0,0000
Indivíduo 38	0,0000	0,2650	0,4000	0,3440	0,0000
Equipe 11					
Indivíduo 69	0,0000	0,0000	0,4000	0,0000	0,0000
Indivíduo 40	0,0000	0,3650	0,3200	0,0000	0,0000
Indivíduo 121	0,4300	0,7000	0,5600	0,4750	0,3800
Indivíduo 5	0,0000	0,2650	0,1600	0,0000	0,0000
Indivíduo 102	1,0000	1,0000	0,8000	0,0000	0,2250
Indivíduo 44	0,0000	0,4300	0,6400	0,2120	0,0000
Indivíduo 115	0,0000	0,6000	0,5600	0,0000	0,0000
Indivíduo 25	0,0000	0,2650	0,2400	0,0000	0,0000
Indivíduo 78	0,0000	0,2650	0,3200	0,2120	0,0000
Indivíduo 89	0,0000	0,4300	0,2400	0,2120	0,2250
Equipe 12					
Indivíduo 37	0,1650	0,2650	0,1600	0,2120	0,1000
Indivíduo 109	0,0000	0,0000	0,6400	0,0000	0,0000
Indivíduo 104	0,1650	0,0000	0,5600	0,1650	0,0000
Indivíduo 32	0,0000	0,9000	0,8000	0,0000	0,0000
Indivíduo 1	0,0000	1,0000	0,8000	0,1320	0,3000
Indivíduo 15	0,0000	0,4300	0,3200	0,0000	0,0000
Indivíduo 10	0,0000	0,2650	0,6400	0,2120	0,0000
Indivíduo 98	0,4300	0,4300	0,5600	0,2120	0,3500
Indivíduo 74	0,0000	0,5300	0,4000	0,0000	0,0000
Indivíduo 55	0,0000	0,0000	0,0000	0,0000	0,0000
Equipe 13					
Indivíduo 92	0,2650	0,8300	0,5600	0,2120	0,0000
Indivíduo 28	0,0000	0,2650	0,2400	0,2120	0,0000
Indivíduo 122	0,0000	0,2650	0,2400	0,0000	0,0000
Indivíduo 62	0,1000	0,5300	0,3200	0,0000	0,2250
Indivíduo 119	0,0000	0,6000	0,4800	0,0000	0,0000
Indivíduo 13	0,2650	0,7000	0,3200	0,0000	0,0000

Indivíduo 17	0,0000	0,0000	0,4000	0,0000	0,0000
Indivíduo 46	0,0000	0,0000	0,1600	0,0000	0,0000
Indivíduo 118	0,0000	0,5300	0,1600	0,2120	0,1000
Indivíduo 75	0,0000	0,2650	0,2400	0,0000	0,2250
Indivíduo 34	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 86	0,0000	0,0000	0,1600	0,3920	0,0000
Indivíduo 23	0,0000	0,2650	0,1600	0,2120	0,2250
Indivíduo 33	0,0000	0,0000	0,6400	0,0000	0,0000
Indivíduo 72	0,4300	0,8000	0,4800	0,0000	0,0000
Equipe 14					
Indivíduo 90	0,0000	0,6300	0,6400	0,0000	0,0000
Indivíduo 29	0,0000	0,6000	0,4800	0,0000	0,0000
Indivíduo 96	0,0000	1,0000	0,2400	0,0000	0,0000
Indivíduo 21	0,2650	0,2650	0,2400	0,0000	0,0000
Indivíduo 50	0,2650	0,0000	0,3200	0,2120	0,0000
Indivíduo 11	0,0000	0,0000	0,3200	0,0000	0,0000
Indivíduo 14	0,0000	0,5300	0,4000	0,0000	0,0000
Indivíduo 113	0,2650	0,2650	0,8000	0,0000	0,0000
Indivíduo 85	0,0000	0,7000	0,0000	0,4750	0,3800
Indivíduo 41	0,0000	0,2650	0,5600	0,2120	0,0000
Indivíduo 52	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 99	0,1000	0,5300	0,3200	0,0000	0,2250
Indivíduo 107	0,4300	0,4300	0,5600	0,2120	0,3800
Indivíduo 66	0,0000	0,6000	0,5600	0,0000	0,0000
Indivíduo 60	0,0000	0,0000	0,0000	0,0000	0,0000
Equipe 15					
Indivíduo 22	0,0000	0,0000	0,0000	0,0000	0,0000
Indivíduo 116	0,0000	0,4300	0,6400	0,2120	0,0000
Indivíduo 93	0,0000	0,2650	0,5600	0,1650	0,0000
Indivíduo 9	0,2650	0,2650	0,8000	0,0000	0,0000
Indivíduo 12	0,2650	0,5300	0,1600	0,0800	0,0000
Indivíduo 114	0,4300	0,9000	0,5600	0,0000	0,3500
Indivíduo 108	0,0000	0,9000	0,8000	0,0000	0,0000
Indivíduo 84	0,0000	0,7000	0,5600	0,0000	0,4750
Indivíduo 48	0,5300	0,0000	0,3200	0,2120	0,0000
Indivíduo 42	0,1650	0,2650	0,5600	0,2120	0,0000
Indivíduo 63	0,2650	0,2650	0,3200	0,1650	0,0000
Indivíduo 47	0,2650	0,5300	0,6400	0,2920	0,3500
Indivíduo 65	0,0000	0,0000	0,2400	0,2120	0,0000
Indivíduo 76	1,0000	1,0000	0,8000	0,0000	0,2250
Indivíduo 81	0,0000	0,4300	0,3200	0,0000	0,1250

	Protocolos de Rede	Paradigmas de Comunicação	SCRUM	Transações Bancárias
Equipe 1				
Indivíduo 120	0,8000	0,8000	0,2800	0,3650
Indivíduo 56	0,6200	0,6200	0,4600	0,2650
Indivíduo 94	0,0800	0,0800	0,0800	0,2650
Equipe2				
Indivíduo 4	0,0800	0,0800	0,3600	0,3650
Indivíduo 20	0,3800	0,0000	0,0000	0,0000
Indivíduo 87	0,0000	0,1800	0,1800	0,2650
Equipe3				
Indivíduo 117	0,5400	0,3600	0,7200	0,1650
Indivíduo 100	0,1800	0,1800	0,1800	0,0000
Indivíduo 82	0,2800	0,2800	0,2800	0,0000
Equipe 4				
Indivíduo 36	0,1800	0,1800	0,1800	0,1650
Indivíduo 88	0,1800	0,2800	0,4600	0,0000
Indivíduo 59	0,3600	0,3600	0,2800	0,1650
Indivíduo 16	0,8000	0,1800	0,0000	0,7000
Indivíduo 27	0,0000	0,0000	0,0000	0,0000
Equipe 5				
Indivíduo 111	0,1800	0,1800	0,1800	0,0000
Indivíduo 26	0,0000	0,0000	0,0000	0,0000
Indivíduo 103	0,2800	0,2800	0,4600	0,2650
Indivíduo 68	0,3600	0,1000	0,0000	0,0000
Indivíduo 35	0,0000	0,1800	0,1800	0,5300
Equipe 6				
Indivíduo 105	0,3600	0,2600	0,2800	0,1650
Indivíduo 39	0,1800	0,1800	0,7200	0,2650
Indivíduo 123	0,8000	0,5600	0,0000	0,0000
Indivíduo 18	0,0000	0,0000	0,0000	0,0000
Indivíduo 95	0,0000	0,0000	0,3600	0,1650
Equipe 7				
Indivíduo 101	0,4800	0,4800	0,2800	0,0000
Indivíduo 54	0,0800	0,0800	0,0000	0,2650
Indivíduo 115	0,8000	0,6400	0,6400	0,0000
Indivíduo 49	0,0000	0,0000	0,0000	0,0000
Indivíduo 8	0,3400	0,0000	0,6000	0,3650
Indivíduo 70	0,8000	0,0000	0,0000	0,0000
Indivíduo 57	0,0000	0,0000	0,0000	0,0000

Indivíduo 83	0,1800	0,1800	0,7400	0,0000
Equipe 8				
Indivíduo 112	0,2800	0,2800	0,4600	0,2650
Indivíduo 71	0,6200	0,0000	0,0000	0,0000
Indivíduo 97	0,2800	0,2800	0,4600	0,2650
Indivíduo 51	0,1000	0,1000	0,1800	0,0000
Indivíduo 80	0,0000	0,0000	0,1800	0,1650
Indivíduo 73	0,1800	0,2800	0,1800	0,2650
Indivíduo 53	0,0000	0,0000	0,1800	0,1650
Indivíduo 106	0,3800	0,3800	0,4600	0,2650
Equipe 9				
Indivíduo 6	0,0000	0,0000	0,1800	0,1650
Indivíduo 77	0,2800	0,3800	0,0000	0,2650
Indivíduo 43	0,0000	0,0000	0,2800	0,2650
Indivíduo 67	0,3600	0,2800	0,0000	0,0000
Indivíduo 2	0,1800	0,0000	0,6400	0,2650
Indivíduo 30	0,1800	0,1800	0,0000	0,0000
Indivíduo 19	0,3800	0,3800	0,1800	0,2650
Indivíduo 61	0,1800	0,2800	0,2800	0,2650
Equipe 10				
Indivíduo 45	0,0000	0,0000	0,1800	0,2650
Indivíduo 110	0,1800	0,3600	0,3600	0,2650
Indivíduo 31	0,2800	0,0000	0,0000	0,0000
Indivíduo 64	0,0000	0,0000	0,0000	0,0000
Indivíduo 91	0,0000	0,0000	0,2600	0,0000
Indivíduo 24	0,0000	0,0000	0,0000	0,0000
Indivíduo 58	0,0000	0,3800	0,0000	0,1650
Indivíduo 3	0,0000	0,0000	0,2600	0,0000
Indivíduo 7	0,0000	0,0000	0,2600	0,0000
Indivíduo 38	0,1800	0,0000	0,1800	0,0000
Equipe 11				
Indivíduo 69	0,8000	0,2800	0,0000	0,0000
Indivíduo 40	0,0000	0,0000	0,1000	0,5300
Indivíduo 121	0,3800	0,3800	0,4600	0,2650
Indivíduo 5	0,0000	0,0000	0,0000	0,2650
Indivíduo 102	0,8000	0,8000	0,2800	0,0000
Indivíduo 44	0,3600	0,3400	0,1800	0,1650
Indivíduo 115	0,3800	0,3800	0,1800	0,0000
Indivíduo 25	0,0000	0,1000	0,1800	0,0000
Indivíduo 78	0,0000	0,0000	0,1800	0,0000
Indivíduo 89	0,2800	0,1800	0,0000	0,0000

Equipe 12				
Indivíduo 37	0,5400	0,3600	0,3600	0,2650
Indivíduo 109	0,1800	0,2800	0,0000	0,0000
Indivíduo 104	0,0000	0,3800	0,0000	0,1650
Indivíduo 32	0,8000	0,5600	0,0000	0,0000
Indivíduo 1	0,1800	0,0000	0,4400	0,2650
Indivíduo 15	0,0800	0,0000	0,0000	0,0000
Indivíduo 10	0,3600	0,3400	0,1800	0,0000
Indivíduo 98	0,3800	0,3800	0,3600	0,2650
Indivíduo 74	0,0000	0,0000	0,1000	0,0000
Indivíduo 55	0,0000	0,0000	0,0000	0,0000
Equipe 13				
Indivíduo 92	0,1800	0,1800	0,0000	0,2650
Indivíduo 28	0,0000	0,1000	0,1800	0,2650
Indivíduo 122	0,1800	0,2800	0,1800	0,2650
Indivíduo 62	0,1000	0,1000	0,1800	0,0000
Indivíduo 119	0,4800	0,4800	0,2800	0,0000
Indivíduo 13	0,0000	0,0000	0,1800	0,0000
Indivíduo 17	0,0800	0,0000	0,0000	0,2650
Indivíduo 46	0,3600	0,3600	0,0000	0,0000
Indivíduo 118	0,1800	0,1800	0,7200	0,2650
Indivíduo 75	0,0000	0,0000	0,0800	0,0000
Indivíduo 34	0,0000	0,0000	0,0000	0,0000
Indivíduo 86	0,1800	0,1800	0,0000	0,0000
Indivíduo 23	0,0000	0,0000	0,0000	0,0000
Indivíduo 33	0,1800	0,2800	0,0000	0,0000
Indivíduo 72	0,0000	0,0000	0,0000	0,0000
Equipe 14				
Indivíduo 90	0,1800	0,2800	0,1800	0,0000
Indivíduo 29	0,4800	0,4800	0,2800	0,0000
Indivíduo 96	0,2800	0,1800	0,2600	0,0000
Indivíduo 21	0,0000	0,0000	0,0000	0,0000
Indivíduo 50	0,0000	0,0000	0,3600	0,0000
Indivíduo 11	0,3600	0,3600	0,1800	0,0000
Indivíduo 14	0,0000	0,0000	0,2800	0,0000
Indivíduo 113	0,5600	0,5600	0,2000	0,0000
Indivíduo 85	0,3800	0,3800	0,2800	0,0000
Indivíduo 41	0,0000	0,0000	0,1800	0,0000
Indivíduo 52	0,0000	0,0000	0,0000	0,0000
Indivíduo 99	0,1000	0,1000	0,1800	0,0000
Indivíduo 107	0,2800	0,2800	0,4600	0,2650

Indivíduo 66	0,3800	0,3800	0,1800	0,0000
Indivíduo 60	0,0000	0,0000	0,0000	0,0000
Equipe 15				
Indivíduo 22	0,0000	0,0000	0,0000	0,0000
Indivíduo 116	0,3600	0,3400	0,1800	0,1650
Indivíduo 93	0,0000	0,1800	0,0000	0,0000
Indivíduo 9	0,5600	0,5600	0,2000	0,0000
Indivíduo 12	0,1800	0,0000	0,1800	0,0000
Indivíduo 114	0,3800	0,3800	0,1800	0,2650
Indivíduo 108	0,8000	0,5600	0,0000	0,0000
Indivíduo 84	0,4800	0,0000	0,0000	0,0000
Indivíduo 48	0,0000	0,0000	0,3600	0,2650
Indivíduo 42	0,5200	0,4400	0,0000	0,3650
Indivíduo 63	0,1800	0,0000	0,0000	0,0000
Indivíduo 47	0,1800	0,3600	0,3600	0,2650
Indivíduo 65	0,1800	0,1000	0,1800	0,0000
Indivíduo 76	0,8000	0,8000	0,2800	0,0000
Indivíduo 81	0,1800	0,1800	0,1800	0,0000
