

APOIO À ALOCAÇÃO DE RECURSOS HUMANOS EM PROJETOS DE
SOFTWARE: UMA ABORDAGEM BASEADA EM SATISFAÇÃO DE
RESTRICÇÕES

Ahilton Silva Barreto

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM
ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Márcio de Oliveira Barros, D. Sc.

Prof^ª. Cláudia Maria Lima Werner, D. Sc.

Prof^ª. Ana Regina Cavalcanti da Rocha, D. Sc.

Prof. Crediné Silva de Menezes, D. Sc.

RIO DE JANEIRO, RJ - BRASIL
AGOSTO DE 2005

BARRETO, AHILTON SILVA

Apoio à Alocação de Recursos Humanos em Projetos de Software: Uma Abordagem Baseada em Satisfação de Restrições [Rio de Janeiro] 2005

XI, 93 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Alocação de Recursos Humanos
2. Gerência de Projetos
3. Apoio à Decisão

I. COPPE/UFRJ II. Título (série)

*À minha querida esposa Andrea.
Aos meus pais, meus irmãos.
Aos demais familiares e amigos.*

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me deu força, capacidade e condições para que conseguisse alcançar mais este objetivo.

À Andrea, minha querida esposa, companheira, colega de mestrado e de trabalho, por todo o apoio nestes anos, pelo amor, convívio, compreensão, ajuda nos momentos difíceis e alegres, sem quem teria sido extremamente mais difícil realizar este trabalho.

Ao Prof. Márcio Barros, com quem muito aprendi, pela ótima orientação, por acreditar em mim e no trabalho, por toda ajuda, todo incentivo, aprendizado e amizade ao longo destes anos.

À Prof^ª. Cláudia Werner, igualmente pela ótima orientação, por todo o acompanhamento, pelos ensinamentos, críticas e sugestões que contribuíram muito para meu desenvolvimento e para o trabalho.

À Prof^ª. Ana Regina, por participar da minha banca, mas muito mais que isso, por todo o apoio dado a mim e a Andrea desde que chegamos ao Rio. Pela oportunidade de trabalhar em tantos projetos, por confiar em mim e em minha capacidade, pelos ensinamentos e amizade.

Aos demais professores do programa, com quem aprendi muito, em especial ao Prof. Guilherme Travassos e à Prof^ª. Inês Dutra que possibilitaram o surgimento de tantas das idéias utilizadas neste trabalho.

Ao Prof. Crediné, por participar da minha banca e por ter sido responsável por me ensinar, na época da UFES, a base de muitas coisas utilizadas neste trabalho.

A meus pais, meus irmãos, meus avós e demais familiares e amigos, por sempre torcerem e acreditarem em mim, pelo amor, pelo carinho, pela preocupação, e por tantas vezes terem compreendido nossa ausência.

A todos os colegas com quem trabalhei durante esses anos, na Embratel, Petrobrás, Relacional e Risk Control. Agradeço a todos pelo convívio e pelo muito que aprendi com todos. Em especial, agradeço a Carmen Maidantchick, Renata, Natanael, Mariano, Gleison, Paula, Andrea e Márcio.

A todos os colegas da COPPE, pela convivência, amizade e auxílio. Foi um prazer estudar e trabalhar com todos. Em especial agradeço às pessoas com quem convivi mais: Andrea, Gleison, Mariano, Paula, Ana Cândida, David, Sômulo, Sávio,

Adriano, Tayana, Arilo, Rafael, Lúcia, Hélio, Fábio, Gladys, Jeann, Reinaldo, Catia, Muradas.

Agradeço também a todos que participaram do estudo experimental realizado e aos colegas que deram sugestões e fizeram críticas (especialmente o pessoal da equipe de reuso). Todos foram muito importantes na realização deste trabalho.

À Taisa, Solange, Mercedes, Claudia Prata e demais funcionários do PESC pela ajuda, sempre que necessária.

À CAPES, pelo apoio financeiro.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

APOIO À ALOCAÇÃO DE RECURSOS HUMANOS EM PROJETOS DE
SOFTWARE: UMA ABORDAGEM BASEADA EM SATISFAÇÃO DE
RESTRICÇÕES

Ahilton Silva Barreto

Agosto / 2005

Orientadores: Márcio de Oliveira Barros

Cláudia Maria de Lima Werner

Programa: Engenharia de Sistemas e Computação

Uma das principais decisões que precisa ser tomada pelo gerente em um projeto de software é como alocar pessoas a cada atividade do projeto. Isso se deve ao fato de que pessoas desempenham um papel fundamental em projetos de software e determinam a qualidade e a produtividade de um projeto. Assim, alocar pessoas a um projeto de forma adequada é de grande importância. No entanto, essa atividade não é simples, uma vez que há muitas possíveis combinações de alocação diferentes e fatores, muitas vezes conflitantes, a considerar (custo, tempo, qualidade, entre outros). Portanto, realizar a tarefa sem apoio pode ser bastante difícil e pode levar a escolha de uma equipe que não seja a melhor equipe para uma dada situação. Este trabalho apresenta uma abordagem de apoio à decisão na alocação de pessoal em projetos de software. São consideradas características necessárias para a execução de cada atividade: as características possuídas por cada profissional e um conjunto de restrições associadas a fatores como custo, tempo, experiência e tamanho da equipe. A partir dessas informações, para auxiliar o gerente em sua decisão, são sugeridas alocações que satisfaçam ao maior número possível de restrições e, eventualmente, priorizem algum dos fatores selecionados. Foi construída uma ferramenta para disponibilizar o apoio à decisão proposto. Foi, ainda, realizado um estudo experimental para avaliar a relevância do apoio à decisão proposto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

STAFFING A SOFTWARE PROJECT SUPPORT: A CONSTRAINT
SATISFACTION BASED APPROACH

Ahilton Silva Barreto

August / 2005

Advisors: Márcio de Oliveira Barros

Cláudia Maria de Lima Werner

Department: System and Computing Engineering

One of the main decisions that has to be made by the manager in a software project is how to staff the project. This is due to the fact that people play a fundamental role in software projects, and they determine the quality and productivity of the project. So, staffing a project properly is very important. However, this is not a simple task, because there are many different possible staff combinations and many factors, usually conflicting, to ponder (such as time, cost, quality, and others). Therefore, to perform the staffing activity without some kind of help can be very difficult and can lead the manager to choose a team which is not the best for a given situation. This work presents a decision support approach to staff a software project. The required characteristics to perform project activities, the characteristics possessed by each person and a set of constraints related to attributes such as cost, time, experience, and team size will be accounted for. Based on this information, teams that satisfy the biggest amount of constraints and that eventually prioritize some selected factors will be suggested. A decision support tool was also developed. An experimental study was executed to evaluate the relevance of the support being proposed.

ÍNDICE

CAPÍTULO 1 - Introdução.....	1
1.1 Motivação.....	1
1.2 Objetivo da Tese.....	3
1.3 Estrutura do Texto.....	4
CAPÍTULO 2 - Gerência de Recursos Humanos em Projetos de Software.....	6
2.1 Introdução.....	6
2.2 Gerência de Projetos e Gerência de Recursos Humanos.....	7
2.3 Fatores Importantes na Gerência de Recursos Humanos.....	8
2.4 Processos de Gerência de Recursos Humanos.....	13
2.5 Abordagens de Apoio à Alocação de Recursos Humanos.....	16
2.6 Conclusão.....	18
CAPÍTULO 3 - Apoio à Decisão Gerencial no Desenvolvimento de Software e Problemas de Satisfação de Restrições.....	19
3.1 Introdução.....	19
3.2 Decisão e Apoio à Decisão.....	20
3.3 Apoio à Decisão no Desenvolvimento de Software.....	26
3.4 Problemas de Satisfação de Restrições.....	31
3.5 Conclusão.....	35
CAPÍTULO 4 - Modelagem e Resolução do Problema de Alocação de Recursos Humanos em Projetos de Software como um Problema de Satisfação de Restrições....	36
4.1 Introdução.....	36
4.2 Definição do Problema de Alocação de Recursos Humanos em Projetos de Software.....	37
4.3 Solução do Problema de Alocação de Recursos Humanos em Projetos de Software.....	44
4.3.1 N Soluções Quaisquer.....	48
4.3.2 Problema Ótimo.....	48
4.3.3 Problema parcial ótimo quanto às restrições.....	49
4.3.4 Problema parcial ótimo quanto às atividades.....	50
4.4 Exemplo de Utilização da Abordagem de Alocação Baseada em Satisfação de Restrições.....	51
4.5 Conclusão.....	54
CAPÍTULO 5 - Um Estudo Experimental para Investigar a Relevância do Uso da Abordagem Proposta.....	56
5.1 Introdução.....	56
5.2 Uma Ferramenta de Apoio à Alocação de Recursos Humanos em Projetos de Software.....	56
5.3 Estudo Experimental.....	62
5.3.1 Definição dos Objetivos.....	62
5.3.2 Planejamento do Estudo.....	63
5.3.3 Definição dos Objetivos.....	66
5.3.4 Análise dos Resultados.....	68
5.4 Conclusão.....	74
CAPÍTULO 6 - Conclusão.....	75
6.1 Considerações Finais.....	75
6.2 Contribuições.....	76
6.3 Limitações e Perspectivas Futuras do Trabalho.....	77
REFERÊNCIAS BIBLIOGRÁFICAS.....	80

ANEXO I - Questionário para Estudo sobre Atividade de Alocação de Recursos Humanos em Projetos de Software.....	89
ANEXO II - Instruções da Execução do Estudo Experimental Distribuídas aos Participantes do Estudo	90
i - Descrição do Problema	90
ii – Lista de Profissionais do Projeto de Exemplo.....	91
iii – Lista de Atividades do Projeto de Exemplo	92

Índice de Figuras

Figura 4.1 – Algoritmo de determinação do domínio inicial	46
Figura 4.2 – Algoritmo FixaAlocação.....	46
Figura 4.3 – Algoritmo Aloca	47
Figura 4.4 – Atividades do exemplo.....	52
Figura 4.5 – Profissionais do exemplo	52
Figura 4.6 – Profissionais qualificados para realizar as atividades	53
Figura 4.7 – Conjuntos de soluções para o problema.....	54
Figura 5.1 – Tela de Cadastro de Atividades.....	58
Figura 5.2 Tela de Seleção e Configuração de Modelos de Produtividade.....	59
Figura 5.3 – Tela de Geração de Equipes (Aspecto Inicial).....	60
Figura 5.4 – Tela de Geração de Equipes, otimizando o fator custo.....	61
Figura 5.5 – Instrumento do Estudo	67
Figura 5.6 – Resultados agrupados por formação acadêmica dos participantes	70
Figura 5.7 – Resultados agrupados por experiência no desenvolvimento de software dos participantes.....	71
Figura 5.8 – Resultados agrupados por experiência no desenvolvimento de software dos participantes.....	72
Figura 5.9 – Resultados do Estudo	73

Índice de Tabelas

Tabela 1.1 – Resultados do Chaos Report (extraído dos relatórios do Standish Group)	1
Tabela 4.1 – Características do exemplo e seus domínios	51
Tabela 4.2 – Variação dos fatores no exemplo.....	53
Tabela 5.1 – Resultado obtido pelo pesquisador	68
Tabela 5.2 – Resultados obtidos pelos participantes	68
Tabela 5.3 – Resultados obtidos pelos participantes agrupados por formação	69
Tabela 5.4 – Resultados obtidos pelos participantes agrupados por experiência na alocação de recursos humanos.....	70
Tabela 5.5 – Resultados obtidos pelos participantes agrupados por tempo de execução do estudo.....	72
Tabela i – Lista de Profissionais.....	91
Tabela ii – Lista de Atividades	92

CAPÍTULO 1

Introdução

1.1 Motivação

Software ainda é produzido com atrasos e consome mais recursos que o planejado. Muitas vezes projetos são cancelados ao longo do desenvolvimento ou se chega a um produto final com qualidade menor que a esperada. Em 1994, em pesquisa realizada pelo Standish Group (STANDISH GROUP, 1994), constatou-se que, dos projetos analisados, apenas 16,2% tiveram sucesso, ou seja, foram concluídos no prazo, no orçamento e com todas as funcionalidades especificadas. 52,7% foram concluídos com atrasos, além do orçamento ou com menos funcionalidades. Por fim, 31,1% foram cancelados ao longo do desenvolvimento. Naquele momento, essas estatísticas eram alarmantes, uma vez que quantias enormes de dinheiro eram gastas nesses projetos e falhar representa um prejuízo muito grande.

A pesquisa do Standish Group é realizada a cada dois anos e é possível perceber melhorias significativas nesses resultados desde a pesquisa inicial em 1994, como mostrado na tabela 1.1. A taxa de sucesso aumentou para mais de um terço (34%) dos projetos. Isso representa mais de 100% de melhoria em relação à taxa de 16% de 1994. O cancelamento de projetos caiu para 15%, que representa menos da metade dos 31% de 1994. Os projetos concluídos fora do planejado ainda são em torno de 51%. Em (STANDISH GROUP, 2003) há uma análise mostrando diversas melhorias ocorridas ao longo desses anos de pesquisa.

Tabela 1.1 – Resultados do Chaos Report (extraído dos relatórios do Standish Group)

	1994	1996	1998	2000	2002
Com Sucesso	16%	27%	26%	28%	34%
Cancelados	31%	40%	28%	23%	15%
Concluídos fora do planejado	53%	33%	46%	49%	51%

Apesar de já ser possível identificar melhorias importantes na taxa de sucesso de projetos de software, ainda há muito que ser feito para aumentar ainda mais o número de projetos bem sucedidos. Muitos problemas ainda impedem a conclusão de um projeto de software no prazo, no orçamento, com as funcionalidades e o nível de

qualidade desejada. Muitos, senão todos, desses problemas foram atribuídos à gerência de projetos de software inadequada (THAYER, 1997). Brooks (1987) já afirmava há mais de quinze anos atrás que novas tecnologias e metodologias não resolveriam os problemas e que não há uma solução milagrosa que solucione tudo de uma só vez. O fato é que o principal fator envolvido no problema são as pessoas que desenvolvem o software (WOHLIN, 2004).

Pessoas são um dos fatores menos formalizados nos processos de desenvolvimento de software atuais. No entanto, sua importância é evidente: as pessoas se comportam de forma não previsível e subjetiva, determinando a produtividade de software, que é fundamentalmente uma atividade intelectual e social (SOMMERVILLE e RODDEN, 1995).

Uma vez que as pessoas desempenham um papel tão fundamental nos projetos de software, alocar pessoas, ou seja, determinar quem irá realizar cada atividade em um projeto de software, é de grande importância.

Na atividade de alocação de recursos humanos algumas decisões precisam ser tomadas pelo gerente de projetos. Entre elas, o gerente deve decidir quem alocar a cada atividade. Para isso, é preciso determinar quais as características (conhecimentos, habilidades, experiências, formação acadêmica e outras) necessárias para a realização de cada atividade, buscar na organização quais são os profissionais que possuem a qualificação necessária e que estejam disponíveis no momento e, finalmente, atribuir os profissionais às atividades a serem desempenhadas (BARRETO *et al.*, 2005b).

Um outro aspecto importante que deve ser considerado na alocação de recursos humanos em um projeto de software é a criação de valor para o projeto e para a organização. O desenvolvimento de software envolve tempo, talento e dinheiro. Em um mercado competitivo, um dos maiores objetivos das organizações é maximizar a criação de valor para um dado investimento. Valor pode ser visto como *time-to-market*, reputação, qualidade do produto, entre outros. Assim sendo, o bom uso de cada recurso disponível é muito importante (BOEHM e SULLIVAN, 2000). Como um dos recursos mais importantes na criação de valor para um projeto é justamente o recurso humano, a alocação deve não apenas considerar os aspectos técnicos (características necessárias para executar as atividades e características possuídas pelos profissionais), mas também deve ser feita de modo a priorizar algum fator que leve a criação de valor para o projeto (BARRETO *et al.*, 2005a). Exemplos desses fatores que podem variar de equipe para

equipe são experiência da equipe, custo com a equipe do projeto, tamanho da equipe, entre outros.

Portanto, a tarefa de alocação não é simples, uma vez que geralmente há uma série de diferentes combinações de alocação possíveis e fatores, muitas vezes conflitantes, a considerar. Por exemplo, considerando uma situação onde haja dez atividades e dez profissionais aptos a realizá-las, o número de diferentes combinações pode chegar a dez bilhões. Portanto, realizar a atividade, considerando todos os fatores envolvidos e todas as combinações sem algum tipo de apoio automatizado, é extremamente difícil, ou até mesmo impossível. Devido a essa complexidade, o gerente nem sempre é capaz de considerar todas as equipes que podem ser montadas e o impacto de cada uma delas sobre algum fator (custo, número de pessoas alocadas, entre outros) do projeto, acabando por escolher uma das equipes baseando-se apenas em sua intuição e em sua experiência. No entanto, ao não considerar todas as potenciais equipes, há o risco de que o gerente não escolha a melhor equipe para uma dada situação. Assim, é interessante que o gerente seja apoiado nessa complexa atividade, de forma a reduzir o esforço necessário para sua realização e possivelmente melhorar a qualidade das equipes escolhidas (BARRETO *et al.*, 2005b).

A disciplina de apoio à decisão na engenharia de software integra inteligência humana e computacional para facilitar a tomada de melhores decisões durante o ciclo de vida do software. Apoio, aqui, significa fornecer acesso à informação, que seria de outra forma inacessível ou difícil de obter, facilitando a geração e avaliação de soluções distintas e priorizando alternativas através do uso de modelos explícitos que forneçam a estrutura para decisões particulares (RUHE, 2003b). Vale salientar que os sistemas de apoio à decisão se propõem a apoiar e não a substituir seus usuários. A decisão final continua sendo feita por humanos (RUS *et al.*, 2002).

1.2 Objetivo da Tese

O principal objetivo desta tese é apoiar decisões gerenciais no contexto da alocação de recursos humanos, fornecendo ao gerente de projetos sugestões de possíveis equipes que satisfaçam às restrições existentes no problema, ou seja, qualificação dos profissionais contra qualificação exigida pelas atividades, além de datas em que os profissionais se encontram disponíveis contra datas em que as atividades precisam ser realizadas. Após atender às restrições do problema, deve ser possível também sugerir as

equipes que otimizem algum fator, tais como custo, tempo, tamanho de equipe e outros, considerando-se todas as possíveis soluções para o problema. Vale destacar que a determinação da qualificação necessária para a realização de cada atividade e a qualificação possuída por cada profissional será de responsabilidade do gerente de projetos. Disponibilizar conhecimento para essa caracterização não está no escopo deste trabalho.

Para fornecer o apoio à decisão proposto, o problema da alocação deverá ser modelado, as funções de custo associadas (menor custo, menor tempo, e outras) deverão ser definidas e formalizadas e deverá ser desenvolvido um algoritmo para determinar as equipes, levando em consideração as restrições do problema.

Deverá, ainda, ser desenvolvida uma ferramenta de apoio à decisão, que implemente o algoritmo e as funções definidos e seja a materialização do apoio proposto neste trabalho. Essa ferramenta deve permitir o cadastro de todas as informações pertinentes (relativas aos profissionais, às atividades, às características, entre outras) e deve ser facilmente integrável com outras ferramentas que porventura já possuam as informações necessárias.

Por fim, deve ser realizado um estudo experimental para caracterizar a execução da atividade de alocação de recursos humanos em projetos de software, buscando obter indícios de que o apoio à decisão proposto realmente traria benefícios para a execução da atividade.

Vale ressaltar que ao propor uma abordagem de apoio à decisão, que se baseia em um modelo simplificado da realidade (a ser apresentado no capítulo 4), esta abordagem não se propõe a resolver o problema para o gerente de projetos, mas sim auxiliá-lo na resolução, fornecendo informações que seriam difíceis de obter de outra forma, e permitindo ao gerente testar diferentes soluções e medir o impacto de cada uma delas no projeto. A decisão final será tomada pelo gerente, que conhece (conhecimento tácito) uma série de outros fatores políticos e estratégicos que não são explicitados no modelo proposto neste trabalho.

1.3 Estrutura do Texto

Este trabalho é composto por cinco capítulos, além desta introdução.

No capítulo 2, *Gerência de Recursos Humanos em Projetos de Software*, aborda-se a importância das pessoas no desenvolvimento de software, mencionando a gerência de

projetos e a gerência de recursos humanos. São apresentados, ainda, diversos estudos que vêm sendo realizados na área, além de processos de gerência de recursos humanos e abordagens de apoio à alocação encontrados na literatura.

No capítulo 3, *Apoio à Decisão Gerencial no Desenvolvimento de Software e Problemas de Satisfação de Restrições*, aborda-se, primeiramente, a questão do apoio à decisão gerencial no desenvolvimento de software, mencionando decisão e apoio à decisão de forma mais ampla e também especificamente no desenvolvimento de software. São, também, apresentados os problemas de satisfação de restrições, a abordagem de modelagem e solução escolhida para o problema da alocação de recursos humanos neste trabalho.

No capítulo 4, *Modelagem e Resolução do Problema de Alocação de Recursos Humanos em Projetos de Software como um Problema de Satisfação de Restrições*, são apresentadas a modelagem e resolução do problema escolhidas neste trabalho. São apresentados os algoritmos desenvolvidos para resolver o problema, além das funções de custo envolvidas no problema.

No capítulo 5, *Um Estudo Experimental para Investigar a Relevância do Uso da Abordagem Proposta*, é apresentado o estudo experimental realizado para avaliar a relevância do uso do apoio proposto no trabalho. Além disso, é apresentada a ferramenta desenvolvida para implementar a abordagem descrita no capítulo 4.

No capítulo 6, *Conclusão*, são apresentadas as considerações finais, contribuições e limitações do trabalho, além de perspectivas de trabalhos futuros.

No anexo I, *Questionário para Estudo Sobre Atividade de Alocação de Recursos Humanos em Projetos de Software*, é apresentado o questionário de caracterização dos participantes utilizado no estudo experimental realizado.

No anexo II, *Instruções da Execução do Estudo Experimental Distribuídas aos Participantes do Estudo*, são apresentadas: a descrição do problema que os participantes deveriam resolver, a lista dos profissionais e das atividades do projeto exemplo utilizado no estudo.

CAPÍTULO 2

Gerência de Recursos Humanos em Projetos de Software

2.1 Introdução

De acordo com a norma NBR ISO 10006 (2000), são as pessoas que determinam a qualidade e o sucesso de um projeto.

A gerência de alocação de recursos humanos visa a encontrar os profissionais com a competência mais apropriada para corresponder às necessidades de um projeto, buscando antecipar eventuais problemas relativos a competências não disponíveis na organização (SCHNAIDER, 2003).

Sendo assim, e como cada projeto de software envolve questões únicas e a tecnologia utilizada muda continuamente (WEI *et al.*, 2002), definir corretamente as características necessárias ao projeto, assim como selecionar, alocar e reter os recursos humanos adequados torna-se fator crítico de sucesso em um projeto. Uma gerência de projeto eficaz consegue trazer as pessoas certas no tempo certo, de forma a conseguir que o trabalho seja realizado com sucesso (REIFER, 2002), enquanto que uma gerência de recursos humanos mal conduzida pode resultar no não atendimento de planos, estimativas ou prazos pré-estabelecidos, impactando a qualidade dos produtos a serem entregues (MURCH, 2000).

Falhar na especificação do fator humano no processo representa o risco de que desenvolvedores desempenhem atividades que estejam além de suas capacidades. A maioria dos modelos de processo existentes não formaliza as habilidades e capacidades necessárias em cada membro (gerentes, desenvolvedores, e outros), para que a equipe seja eficiente. Isso torna difícil gerenciar as pessoas que desempenham as atividades do processo de desenvolvimento, uma vez que não há procedimentos definidos para atribuir pessoas a papéis de acordo com suas capacidades e as capacidades exigidas pelo papel (ACUÑA e JURISTO, 2003).

O entendimento do processo de desenvolvimento de software sob um enfoque social, no qual a interação entre os membros da equipe é a chave para o sucesso, tem como foco os esforços do gerente de projeto na definição de quais tarefas devem ser executadas, por quem, quando e apoiado por qual ferramenta (WASTELL *et al.*, 1999).

Para obter sucesso nos projetos de desenvolvimento de software, as organizações precisam gerenciar eficientemente os recursos humanos alocados, pois a qualidade dos produtos a serem desenvolvidos é altamente dependente da competência e da disponibilidade destes recursos humanos (SCHNAIDER, 2003).

Este capítulo apresenta um estudo realizado sobre a gerência de recursos humanos em projetos de software. A seguir é discutido o relacionamento entre a gerência de recursos humanos e a gerência de projetos de desenvolvimento de software. Na seção 2.3, são discutidos diversos fatores que influem na gerência de recursos humanos em projetos de software. Os processos de gerência de recursos humanos e suas diversas atividades são discutidos na seção 2.4. Na seção 2.5, serão citadas algumas abordagens de apoio à alocação de recursos humanos encontradas na literatura. Por fim, na seção 2.6 são apresentadas as considerações finais deste capítulo.

2.2 Gerência de Projetos e Gerência de Recursos Humanos

O termo “gerência” é definido de diversas formas na literatura. Em (THAYER, 1997), foi definido como o conjunto de atividades e tarefas realizadas por um ou mais indivíduos com o propósito de planejar, executar e controlar as atividades realizadas por um outro grupo, visando atingir determinados objetivos pré-estabelecidos. Em (KOONTZ *et al.*, 1980), é dito que gerenciar consiste em executar atividades e tarefas que têm como propósito planejar e controlar as atividades de outras pessoas, para atingir objetivos que não podem ser alcançados caso as pessoas atuem por conta própria.

A gerência de projetos pode ser definida como sendo a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos (PMI, 2004). Em uma outra definição, de acordo com a norma NBR ISO 10006 (2000), a gerência de projetos inclui o planejamento, organização, supervisão e controle de todos os aspectos do projeto, em processo contínuo, visando alcançar seus objetivos. Alinhado a esta definição, THAYER (1997) considera que a gerência de projeto pode ser vista como o conjunto de procedimentos, práticas, tecnologias, habilidades e experiências que possibilitam a realização do planejamento, organização, seleção de recursos humanos, supervisão e controle necessários para que se gerencie um projeto com sucesso.

A importância da gerência de projetos de software pode ser ilustrada nas seguintes frases, extraídas de relatórios do DoD (Departamento de Defesa Norte-Americano):

- “O gerente desempenha papel principal no desenvolvimento de software. A diferença entre sucesso e fracasso – entre um projeto no cronograma e dentro do orçamento, ou atrasado e além do orçamento – é normalmente uma função da efetividade do gerente” (DoD, 1983).
- “Hoje os maiores problemas no desenvolvimento de software não são os problemas técnicos, mas os problemas de gerenciamento” (BROOKS, 1987).

A gerência de recursos humanos de um projeto inclui os processos que organizam e gerenciam a equipe do projeto (PMI, 2004). Está inserida na gerência de projetos, representando um item fundamental, dada a imensa importância das pessoas nos projetos, principalmente nos projetos de desenvolvimento de software. Essa relação entre a gerência de recursos humanos com a gerência de projetos pode ser exemplificada em:

- (THAYER, 1997), que considera que uma das funções da gerência é selecionar pessoas;
- NBR ISO 10006 (2000), que trata sobre diretrizes para a qualidade no gerenciamento de projetos e considera, nos processos de gerenciamento do projeto, os processos relacionados às pessoas;
- (PMI, 2004), que considera o gerenciamento de recursos humanos do projeto como sendo uma das áreas de conhecimento envolvidas no gerenciamento de projetos;
- MPS.BR (SOFTEX, 2005), que considera que um dos resultados esperados do processo de gerência de projetos é que os recursos humanos estejam planejados considerando o perfil e conhecimentos necessários para executar o projeto.

Detalhes sobre processos de gerência de recursos humanos e a indicação de onde nesses processos este trabalho se insere são apresentados na seção 2.4. Na próxima seção, apresentamos diversos fatores pesquisados na literatura que influem de alguma forma na gerência de recursos humanos em projetos de software.

2.3 Fatores Importantes na Gerência de Recursos Humanos

Dada a importância das pessoas no desenvolvimento de software, muitos estudos têm sido feitos buscando entender melhor como as pessoas se comportam, o que

esperam, do que gostam, o que as faz trabalhar mais e melhor (ACUÑA e JURISTO, 2003; SUMNER e NIEDERMAN, 2002; PARÉ *et al.*, 2001; LEE, 2002; AGARWAL e FERRAT, 2000; THATCHER *et al.*, 2002). Nesta seção, são apresentadas algumas pesquisas recentes na área de recursos humanos em projetos de software. Em sua maioria, essas pesquisas procuram identificar fatores relevantes e importantes para a gestão de recursos humanos, além das relações entre esses fatores.

O pessoal destinado ao projeto pode diferir de muitas maneiras, e não basta, por exemplo, dizer que um projeto precisa de um analista, dois projetistas e cinco programadores. Duas pessoas com o mesmo cargo podem diferir em pelo menos um dos seguintes aspectos (PFLEEGER, 2004):

- Capacidade para desempenhar o trabalho;
- Interesse no trabalho;
- Experiência com aplicações semelhantes;
- Experiência com ferramentas ou linguagens semelhantes;
- Experiência com técnicas semelhantes;
- Experiência com ambiente de desenvolvimento semelhante;
- Treinamento;
- Capacidade para se comunicar com outras pessoas;
- Capacidade para compartilhar responsabilidades com outras pessoas;
- Habilidades de gerenciamento.

Cada uma dessas características pode afetar a capacidade de um indivíduo para realizar seu trabalho de forma produtiva. Essas variações ajudam a explicar por que, por exemplo, um programador pode escrever determinada rotina em um dia, enquanto outro necessita de uma semana para realizar a mesma tarefa. As diferenças podem ter um grande impacto, não somente para a estimativa do cronograma, mas também para o sucesso do projeto (PFLEEGER, 2004).

Os funcionários de uma organização adquirem conhecimentos, habilidades e experiências com o passar do tempo. Muitas vezes, a alocação a uma determinada posição é realizada baseando-se nessas características: cada posição requer um conjunto mínimo de habilidades, conhecimentos e experiências para que suas atribuições possam ser cumpridas (SOUZA, 2003).

Para identificar fatores que influenciam a produtividade e a motivação das pessoas, e que complementam o conhecimento no que diz respeito à alocação de recursos humanos, foram realizados diversos estudos experimentais nos últimos anos.

Em (SUMNER e NIEDERMAN, 2002), foi feito um estudo experimental que buscava relações entre a diferença de gênero dos profissionais de informática e a satisfação no trabalho, o desempenho na execução das tarefas e nas intenções de pedir demissão da empresa. Como resultado, observou-se que o gênero não tem relação com nenhum desses fatores. Esse resultado indica apenas que homens e mulheres se comportam de forma equivalente quanto aos fatores analisados.

Em (PARÉ *et al.*, 2001), foi feito um estudo que considerou diversos fatores, como comprometimento organizacional (afetivo, de continuidade e normativo), comportamento cidadão organizacional (quando a pessoa executa tarefas que estão além de suas obrigações voluntariamente, para contribuir com a empresa) e várias práticas de recursos humanos (reconhecimento, práticas de desenvolvimento de competências, justiça, compartilhamento de informações, etc) e procurou relacioná-los com as intenções de pedir demissão dos empregados. Os resultados mostram quais fatores fazem diminuir as intenções de pedir demissão. Em geral, as práticas de recursos humanos aumentam o comprometimento e o comportamento cidadão, e estes, por sua vez, diminuem as intenções de pedir demissão da empresa. Esses resultados são difíceis de serem considerados na gerência de recursos humanos, uma vez que são difíceis de medir nas pessoas. Mas sabendo disso, o gerente pode tentar adotar essas práticas, para que o comprometimento e o comportamento cidadão aumentem, diminuindo assim a probabilidade de que essas pessoas venham a deixar a empresa.

Em (LEE, 2002), é apresentada uma pesquisa que procura identificar a importância do apoio social. Apoio social é definido como sendo a disponibilidade de relacionamentos de ajuda e a qualidade desses relacionamentos. O apoio social leva uma pessoa a acreditar que é estimada pelos outros, que os outros se importam com ela e a valorizam. Essa pesquisa concluiu que, em geral, o apoio social dos supervisores e dos colegas influencia negativamente as intenções de pedir demissão dos empregados e positivamente a satisfação no trabalho. A pesquisa conclui, ainda, que a satisfação no trabalho diminui as intenções de pedir demissão. Esse resultado também é difícil de ser usado no contexto da gerência de recursos humanos. No entanto, é possível que o gerente, tendo consciência disso, tente estimular esse apoio social nas equipes.

Em (THATCHER *et al.*, 2002), é pesquisada a relação da motivação intrínseca com as atitudes e intenções dos profissionais em relação a seu trabalho. A motivação intrínseca se refere ao grau de prazer que um empregado experimenta ao fazer seu trabalho efetivamente. Assim, a hipótese era que alguns motivadores influenciavam na motivação intrínseca: autonomia, variedade de tarefas, significância das tarefas, identificação com as tarefas, retorno (*feedback*), satisfação com o supervisor e satisfação com o salário. Além disso, que a motivação intrínseca se relacionava positivamente com a satisfação no trabalho e com o comprometimento afetivo, e que esses dois se relacionavam negativamente com as intenções de pedir demissão. Como resultado, percebeu-se que, na verdade, a motivação intrínseca não exibiu relação com a significância das tarefas, com a identificação com as tarefas, com o retorno e com a satisfação de salário. Entretanto, demonstrou um relacionamento positivo com a autonomia, variedade de tarefas e satisfação com a supervisão. Além disso, a relação positiva entre a motivação intrínseca e a satisfação no trabalho foi confirmada e o fator mediador no comprometimento afetivo e nas intenções de sair também.

Em (AGARWAL e FERRAT, 2000), é feito um estudo para investigar o comportamento de demissão voluntária dos profissionais envolvidos no desenvolvimento de software. A primeira hipótese é que a duração preferida de emprego de profissionais de tecnologia de informação, ou seja, o tempo de permanência na empresa que o profissional considera ideal, é relacionado negativamente com as intenções de sair da empresa. Outra hipótese é que o estágio da carreira de um profissional de tecnologia de informação modera o relacionamento entre a duração preferida de emprego e as intenções de pedir demissão da empresa. Por exemplo, indivíduos em estágios de carreira mais altos tendem a desenvolver um comprometimento organizacional maior e têm uma tendência maior a permanecer no emprego até se aposentarem. A terceira hipótese é que as âncoras de carreira de um profissional de tecnologia de informação moderam o relacionamento entre a duração preferida de emprego e as intenções de sair da empresa. Âncora de carreira diz respeito à percepção pessoal de talentos e habilidades, valores básicos e, mais importante, o conjunto de motivações e necessidades que se tem em relação à carreira. Assim, cada âncora representa um conjunto básico de necessidades que o indivíduo espera atender. A única que foi confirmada pelo estudo foi a primeira hipótese, apesar de parecer coerente que os dois outros fatores também tenham influência. Assim, mais um fator foi definido como importante na questão da saída de profissionais das empresas. Apesar de

a conclusão a que se chegou parecer bastante óbvia, nos casos em que uma empresa procura profissionais para um relacionamento de longo prazo, esse fator também deve ser lembrado.

Reconhecendo a importância da força de trabalho e a importância das práticas relativas ao pessoal da empresa, foi desenvolvido o *People Capability Maturity Model (People CMM)* (SEI, 2001). O *People CMM* é um guia para implementação de práticas que continuamente melhoram as capacidades da força de trabalho de uma organização. Uma vez que a organização não pode implementar todas as melhores práticas de força de trabalho de uma só vez, o *People CMM* as introduz em estágios. Cada estágio produz uma transformação única na cultura da organização, fazendo com que a organização tenha práticas mais efetivas para atrair, desenvolver, organizar, motivar e reter sua força de trabalho.

Em (ACUÑA e JURISTO, 2003), propõe-se um modelo de processo de software orientado a capacidades. Este modelo inclui a formalização das capacidades ou competências comportamentais de pessoas que realizam as atividades do processo de desenvolvimento. A hipótese é que a adição de dois novos elementos (as capacidades das pessoas e as capacidades necessárias para cada papel) melhora o processo de desenvolvimento de software. Assim, foram definidos os relacionamentos: capacidade-pessoa e capacidade-papel. Define-se capacidade como sendo uma habilidade ou atributo de comportamento pessoal que pode ser considerado como uma característica de comportamento. Neste estudo, foi proposta uma forma de avaliar quais capacidades as pessoas possuíam e, além disso, para cada atividade no processo de desenvolvimento de software, quais capacidades eram necessárias para sua execução.

A norma NBR ISO 10006 (2000) recomenda que a competência necessária para a realização de cada atividade de um projeto seja definida em termos de educação, conhecimento e experiência para todos os recursos humanos que serão atuantes no projeto. A norma também cita a necessidade de serem considerados, na identificação das competências, aspectos relacionados aos interesses pessoais, relacionamentos interpessoais e pontos fortes e fracos, pois o conhecimento das características e experiências pessoais de cada profissional pode ajudar na identificação da melhor distribuição de responsabilidades entre os membros da equipe do projeto.

Em (SOUZA, 2003), é desenvolvida uma abordagem de representação da distribuição do conhecimento, habilidades e experiências através da estrutura

organizacional. Esses fatores são utilizados na atribuição de posições na organização e papéis no projeto e podem ser definidos da seguinte forma:

- **Conhecimento:** competência adquirida que representa a apropriação de um objeto pelo pensamento através de definição, percepção clara, apreensão completa, análise, etc. Por exemplo, podem ser considerados conhecimentos sobre a especialidade de cirurgia cardíaca ou sobre a manipulação de instrumentos aeronáuticos;
- **Habilidade:** competência que representa a aptidão nata ou adquirida não associada a uma atividade ou área de competência específica. Por exemplo, capacidade de liderança, disciplina ou tolerância a situações de estresse;
- **Experiência:** competência adquirida através da execução de atividades. Por exemplo, experiência em gerência de projetos, experiência na realização de análise de requisitos e outros;

Em (SCHNAIDER, 2003), aborda-se a questão do planejamento da alocação de recursos humanos, considerando esses mesmos fatores e ainda a formação acadêmica dos profissionais, que pode ser vista como uma informação relativa ao nível de escolaridade, área acadêmica e instituição de ensino de um profissional. Por exemplo, mestrado em Engenharia de Sistemas e Computação na COPPE/UFRJ.

Diante do que foi apresentado nesta seção, percebe-se que são muitos os fatores que influenciam as equipes de software e, portanto, a alocação de recursos humanos. Assim, no modelo de alocação de recursos humanos proposto neste trabalho, as características necessárias para execução de cada atividade e as características possuídas pelos profissionais serão consideradas.

Na próxima seção, aborda-se a questão dos processos que envolvem a gerência de recursos humanos e onde este trabalho se insere.

2.4 Processos de Gerência de Recursos Humanos

Vários processos propostos na literatura contemplam, de alguma forma, a questão da gerência de recursos humanos. A norma NBR ISO 10006 (2000), que trata de diretrizes para a qualidade no gerenciamento de projetos, define os processos relacionados às pessoas. Esses processos visam criar um ambiente no qual a equipe possa contribuir efetiva e eficientemente para o projeto. Esses processos são os seguintes:

- Definição da estrutura organizacional do projeto: definir uma estrutura organizacional adequada às necessidades do projeto, incluindo a identificação de funções e definição de autoridade e responsabilidades;
- Alocação da equipe: selecionar e nomear pessoal suficiente, com competência apropriada para corresponder às necessidades do projeto;
- Desenvolvimento da equipe: desenvolver habilidades individuais e coletivas para melhorar o desempenho do projeto.

No *Project Management Body of Knowledge* (PMI, 2004) define-se o gerenciamento dos recursos humanos como sendo uma das áreas de conhecimento envolvidas na gerência de projetos. O gerenciamento dos recursos humanos do projeto inclui os processos que organizam e gerenciam a equipe do projeto. Esses processos são os seguintes:

- Planejamento de recursos humanos: identificação e documentação de funções, responsabilidades e relações hierárquicas do projeto, além da criação do plano de gerenciamento de pessoal;
- Contratar ou mobilizar a equipe do projeto: obtenção dos recursos humanos necessários para terminar o projeto;
- Desenvolver a equipe do projeto: melhoria de competências e interação de membros da equipe para aprimorar o desempenho do projeto;
- Gerenciar a equipe do projeto: acompanhamento do desempenho de membros da equipe, fornecimento de *feedback*, resolução de problemas e coordenação de mudanças para melhorar o desempenho do projeto.

DINGSOYR e ROYRVIK (2001) permitem concluir, a partir da discussão sobre “gerência de competências” em uma organização, que um processo de gerência de recursos humanos consiste de três etapas principais:

- Busca das competências necessárias: inclui a definição das competências em si, a busca dos profissionais da organização que as possuem e a seleção dos profissionais desejados;
- Alocação de recursos humanos: inclui a alocação e desalocação de profissionais;
- Desenvolvimento de competências: inclui a avaliação dos recursos humanos para posterior capacitação dos mesmos.

A norma ISO/IEC 12207 (2001), processos de ciclo de vida de software, trouxe em sua primeira emenda um novo processo organizacional, o processo de gerência de recursos humanos. O objetivo do processo é fornecer ao projeto e à organização indivíduos que possuam as habilidades e os conhecimentos para desempenharem seus papéis efetivamente e que trabalhem em conjunto como um grupo coeso. Como saída desse processo, espera-se que:

- Indivíduos com as habilidades e competências sejam identificados e recrutados;
- Interação efetiva entre os indivíduos e grupos seja apoiada;
- A força de trabalho tenha as habilidades para compartilhar informação e coordenar suas atividades eficientemente;
- Critérios objetivos sejam definidos para monitorar o desempenho individual e do grupo, de forma a fornecer um retorno do desempenho e melhorá-lo.

Em (SCHNAIDER, 2003), foi definido um processo de gerência de recursos humanos em projetos de desenvolvimento de software. O processo é composto pelos seguintes sub-processos: *Planejamento da Alocação de Recursos Humanos*, *Monitoração da Alocação de Recursos Humanos* e *Avaliação de Recursos Humanos*. Todos esses processos, de uma forma geral, estão sob a responsabilidade do gerente do projeto de desenvolvimento de software em questão, sempre apoiado pelo gerente de informática da organização. No contexto deste trabalho, merece destaque o processo de planejamento da alocação de recursos humanos.

O processo *Planejamento da Alocação de Recursos Humanos* tem por objetivo planejar a alocação dos profissionais necessários à execução do projeto de desenvolvimento de software em questão. Esse planejamento baseia-se na identificação dos perfis de competência necessários à realização de cada atividade do projeto, definidos pelo responsável, baseando-se em sua habilidade em identificar e graduar as competências necessárias a partir das informações levantadas na proposta técnica do projeto. Entende-se por perfil de competência um determinado conjunto de conhecimentos, habilidades, formação acadêmica e experiência em atividades econômicas. São atividades desse processo (SCHNAIDER, 2003):

- Definir Perfis de Competência: o objetivo desta atividade é definir os perfis de competência necessários a execução de cada atividade do projeto, visando,

posteriormente, a seleção e a alocação dos profissionais mais adequados para o projeto;

- **Selecionar Equipe:** o objetivo desta atividade é selecionar os profissionais a serem alocados ao projeto. Profissionais são selecionados para alocação a cada perfil de competência especificado para o projeto;
- **Contratar Profissionais:** o objetivo desta atividade é solicitar à área de recursos humanos a contratação de um profissional para ser alocado a um determinado projeto de desenvolvimento de software. Esta atividade é executada somente quando não é encontrado na organização nenhum recurso humano que possua um determinado perfil de competência especificado para um projeto – seja porque a organização carece de profissionais com este perfil ou porque todos que o possuem encontram-se alocados a outros projetos no período desejado;
- **Capacitar Profissionais:** o objetivo desta atividade é solicitar à área de Recursos Humanos a execução de um plano de capacitação para um profissional a ser alocado a um projeto de desenvolvimento de software. Esta atividade é executada somente quando conhecimentos específicos (ou uma determinada graduação destes) são detectados como necessários a um profissional para que este venha a possuir um perfil de competência desejado. Representa uma opção à contratação de um novo profissional, mas é aplicável apenas quando o responsável entende que o profissional em questão tem potencial de desenvolver em tempo hábil os níveis de competência requeridos.

Neste trabalho, consideramos apenas a execução das fases de determinação da qualificação necessária para a execução de cada atividade e da seleção de equipe, que são mencionadas, de alguma forma, em todos os processos citados nesta seção.

Na próxima seção, citamos alguns trabalhos na área de apoio à alocação de recursos humanos encontrados na literatura.

2.5 Abordagens de Apoio à Alocação de Recursos Humanos

Muitas abordagens têm sido propostas na literatura para apoiar a alocação de recursos humanos. Dado que neste trabalho o objetivo é justamente apresentar uma abordagem de apoio à alocação de recursos humanos, nesta seção serão citados alguns

desses trabalhos e descritas as abordagens escolhidas neles para apoiar a realização da atividade.

Em (DUGGAN *et al.*, 2004), é descrito um otimizador de alocação para tarefas de construção de software. O otimizador foi construído com base em algoritmos evolucionários de múltiplos objetivos e em algoritmos genéticos (COELLO *et al.*, 2002) e considera as características do projeto e a experiência dos desenvolvedores em cada área (redes, bancos de dados, interfaces com o usuário, entre outros) envolvida nas atividades que precisam ser realizadas. Com base nessas informações e em modelos desenvolvidos pelos autores, o otimizador é capaz de retornar possíveis equipes que representem aproximações dos valores ótimos, considerando-se os múltiplos objetivos envolvidos no problema (como tempo e número de defeitos, por exemplo).

Em (SCHNAIDER, 2003), é apresentada a ferramenta RHPlan, que apóia os processos de planejamento, monitoração e avaliação da alocação de recursos humanos definidas no processo de gerência da alocação de recursos humanos de (SCHNAIDER, 2003). RHPlan está inserida no contexto dos Ambientes de Desenvolvimento de Software Orientados à Organização (VILLELA, 2004). A ferramenta possibilita a utilização do conhecimento organizacional de competências de recursos humanos, armazenado no repositório da organização.

Em (COLLOFELLO *et al.*, 1998), é apresentado um simulador de processo de desenvolvimento de software, que pode ser ajustado para uma organização específica e que permite avaliar os efeitos de decisões gerenciais quanto à alocação de recursos humanos no orçamento, cronograma e qualidade do projeto em diferentes cenários alternativos (*what-if*). O modelo de simulação desenvolvido, baseado em dinâmica de sistemas (ABDEL-HAMID e MADNICK, 1991), considera fatores como nível de experiência, produtividade, pressão do cronograma, número de defeitos gerados, entre outros.

Em (HUANG *et al.*, 2004), é descrito um sistema baseado em teoria *fuzzy* e em redes neurais para a seleção de recursos humanos. Em (RUSKOVA *et al.*, 2002), se apresenta um sistema para a seleção e avaliação de recursos humanos, também baseado em lógica *fuzzy*.

2.6 Conclusão

As pessoas têm papel fundamental no desenvolvimento de software. Portanto, uma boa gerência de recursos humanos em projetos de software é de extrema importância. Neste capítulo foi apresentada uma revisão da literatura sobre a gerência de recursos humanos, que buscou ressaltar a importância dos recursos humanos em projetos de software e a importância de uma boa gerência de recursos humanos.

Foi apresentada uma relação entre a gerência de projetos e a gerência de recursos humanos, buscando contextualizar a gerência de recursos humanos na gerência de projetos. Além disso, foram citados alguns fatores que influenciam na gerência de recursos humanos.

Foram, também, apresentados alguns processos de gerência de recursos humanos descritos na literatura, e quais fases desses processos são considerados neste trabalho.

No próximo capítulo, discute-se a questão das decisões gerenciais, principalmente do apoio a essas decisões no contexto de projetos de software.

CAPÍTULO 3

Apoio à Decisão Gerencial no Desenvolvimento de Software e Problemas de Satisfação de Restrições

3.1 Introdução

Nos primeiros anos após a sua criação, os computadores eram usados para coletar e armazenar dados de forma a permitir a execução de forma mais eficiente de atividades clericais. Mais tarde, as pessoas se tornaram capazes de manipular e gerenciar os dados para operar mais eficientemente seus negócios, localizando grupos alvos para vendas, usando os dados para determinar procedimentos de redução de custos e outros métodos operacionais. Depois disso, apesar desses métodos operacionais continuarem sendo importantes, os sistemas de apoio à decisão surgiram. Esses sistemas apresentavam modelos estruturados para decisões. Os gerentes poderiam usar esses sistemas quando dados específicos a respeito do futuro, por exemplo, não podiam ser determinados. Processos estocásticos foram adicionados para determinar a probabilidade dos eventos e o modelo resultante ajudou ao tomador de decisão a formar suas opiniões (WYSOCKI e YOUNG, 1990).

A tomada de decisão é uma disciplina bem estabelecida que se originou a partir de outras áreas do conhecimento e possui grande interação com essas áreas, tais como economia, pesquisa operacional, teoria dos jogos, teoria de probabilidade, teoria de controle, psicologia e ciência cognitiva. A ênfase do apoio à decisão é fornecer o máximo possível de informação para se tomar a decisão de fato. O apoio à decisão foi projetado, desenvolvido e aplicado com sucesso em muitas áreas, como logística, manufatura, saúde ou agricultura (RUHE, 2003a).

O desenvolvimento de software é um processo onde cada pessoa envolvida constantemente toma decisões, tanto técnicas quanto gerenciais. Um gerente de projetos tem um grande conjunto de fatores a considerar, várias atividades para planejar e controlar e diferentes decisões a tomar. Por exemplo: Como alocar tempo para as diferentes fases do projeto? Como alocar o pessoal? Como aumentar a qualidade do produto? Quanto de teste deve ser feito? Novas técnicas devem ser inseridas? O que funcionaria melhor para o projeto? (RUS *et al.*, 2002).

Para responder a essas perguntas um gerente poderia recorrer a diversas fontes. Há livros, artigos, publicações na Internet e grupos de discussão disponíveis, mas levaria muito tempo para investigar esses recursos, e tempo é um recurso particularmente escasso para gerentes de projeto. Assim, seria interessante que existissem formas de ajudá-los e guiá-los em suas decisões (RUS *et al.*, 2002).

Muitos métodos e técnicas podem ser usados para auxiliar na modelagem de problemas de decisão. No contexto deste trabalho, destacamos a modelagem dos problemas de decisão como problemas de satisfação de restrições.

Um problema de satisfação de restrições é composto de três partes principais: um conjunto de variáveis com valores desconhecidos, um conjunto de domínios (valores que as variáveis podem assumir) e um conjunto de restrições. Solucionar o problema significa atribuir valores a todas as variáveis de forma a atender as restrições existentes. As aplicações deste tipo de problema incluem escalonamento, computação gráfica, sistemas de bancos de dados, biologia molecular, aplicações comerciais, engenharia elétrica, projeto de circuitos, alocação de recursos, entre outros.

Este capítulo apresenta um estudo realizado sobre sistemas de apoio à decisão, focalizando o contexto do desenvolvimento de software. Além disso, o capítulo apresenta um estudo sobre os problemas de satisfação de restrições, uma vez que estes podem ser aplicados no contexto do apoio à decisão. Na seção 3.2, é discutido o apoio à decisão de forma geral. Na seção 3.3, é discutido o apoio à decisão especificamente no contexto de projetos de desenvolvimento de software. Os problemas de satisfação de restrições são abordados na seção 3.4. Finalmente, na seção 3.5 são apresentadas as considerações finais deste capítulo.

3.2 Decisão e Apoio à Decisão

Segundo SPRADLIN (1997), uma decisão é uma alocação de recursos. O tomador de decisão é aquele que tem a autoridade sobre os recursos sendo alocados. Presumidamente, ele faz a decisão considerando algum objetivo, que é o que ele espera atingir com a alocação de recursos. O tomador de decisão toma decisões consistentes com seus valores, que são fatores que são importantes para ele, especialmente aqueles que são relevantes no contexto da decisão. Um valor comum é o econômico, de acordo com o qual o tomador de decisão tenta aumentar seu ganho. Outros valores podem ser pessoais, como felicidade ou segurança, ou sociais, como justiça (SPRADLIN, 1997).

Um tomador de decisão pode empregar a análise de decisão, que é uma forma estruturada de pensar sobre como as ações tomadas a partir da decisão corrente levariam a um resultado. Ao fazer isso, distinguem-se quatro fatores na situação: a decisão a ser tomada, eventos desconhecidos que podem afetar o resultado, a chance dos eventos acontecerem e o próprio resultado. A análise de decisão então constrói modelos, representações lógicas e matemáticas dos relacionamentos entre esses fatores no contexto da situação de decisão. Os modelos, então, permitem que o tomador de decisão estime possíveis implicações de cada curso de ação que ele pode vir a tomar, de forma que possa melhor entender o relacionamento entre suas ações e seus objetivos (SPRADLIN, 1997).

A humanidade tem tomado decisões fazendo uso do bom senso, de heurísticas ou de forma intuitiva. Somente no último século, a teoria da decisão surgiu como um elemento formal de auxílio à decisão (SAMSON, 1988). A análise de decisão ajuda a dar apoio aos processos decisórios e às habilidades intuitivas e cognitivas do tomador de decisão. Os modelos aplicados são baseados em abstrações da realidade que focalizam cada estágio de solução de um problema, desde a identificação e formulação, até a solução do mesmo (GARBER, 2002).

De acordo com (BELL *et al.*, 1988), podem-se distinguir três diferentes perspectivas no estudo da tomada de decisão. Na perspectiva normativa, o foco está na escolha racional. Modelos normativos são construídos a partir de suposições básicas (ou axiomas) que as pessoas consideram fornecer apoio lógico para suas decisões. No domínio do apoio à decisão sob risco ou incerteza, o modelo de utilidade esperada de VON NEUMANN e MORGENSTERN (1944) e o modelo de utilidade esperada subjetiva de SAGAGE (1954) são os principais modelos normativos para escolhas racionais. Além desses modelos, a teoria probabilística e a estatística Bayesiana também fornecem base normativa (SMITH e WINTERFELDT, 2004).

A perspectiva descritiva foca em como pessoas realmente pensam e se comportam. Estudos descritivos podem desenvolver modelos matemáticos de comportamento, mas esses modelos são avaliados pela extensão pela qual suas predições correspondem às decisões que as pessoas tomam de fato. Um dos modelos descritivos de tomada de decisão sob incerteza que mais se destaca é o modelo da teoria da perspectiva de KAHNEMAN e TVERSKY (1979), refinado mais tarde em (KAHNEMAN *et al.*, 1982). Esse modelo captura muitas das formas pelas quais as pessoas se desviam do ideal normativo do modelo de utilidade esperada (SMITH e WINTERFELDT, 2004).

A perspectiva prescritiva foca em ajudar as pessoas a tomarem melhores decisões, através do uso de modelos normativos, mas com a noção das limitações e realidades descritivas do julgamento humano. Por exemplo, pode-se construir um modelo matemático para ajudar uma empresa a decidir se deve realizar um determinado projeto. Esse poderia não incluir todas as incertezas e fatores realmente envolvidos no problema. Provavelmente, o modelo incluiria aproximações da realidade para que se tornasse mais simples de formular, avaliar e resolver. A pesquisa descritiva de tomada de decisão ajudaria os analistas a entenderem, por exemplo, quais parâmetros de entrada do modelo podem ser verificados de forma confiável e como essas entradas podem estar influenciadas por algum outro fator. Os modelos prescritivos são avaliados pragmaticamente, por exemplo: os tomadores de decisão os acham úteis? Ou, o que é mais difícil de determinar, eles ajudam as pessoas a tomarem decisões melhores? (SMITH e WINTERFELDT, 2004)

A análise de decisão é fundamentalmente uma disciplina prescritiva, baseada em fundamentos normativos e descritivos.

Pode-se dizer que a análise de decisão tem suas origens há alguns séculos atrás. Um dos trabalhos iniciais que merece destaque é o de Bernoulli, em 1738, com o paradoxo de São Petersburgo. Esse paradoxo consiste de um jogo hipotético entre duas pessoas que ilustrava a idéia de que o valor de um item não deve se basear em seu preço, mas na utilidade que produz. Muitos outros trabalhos podem ser citados para mostrar a evolução da análise de decisão ao longo do tempo (BAYES, 1763; RAMSEY, 1931; DeFINETTI, 1937; VON NEUMANN E MORGENSTERN, 1944; SAVAGE, 1954; PRATT *et al.*, 1964; RAIFFA, 1968), mas uma revisão histórica sobre o tema não está no escopo deste trabalho. Uma boa fonte de informações sobre a história da análise de decisão é (SMITH e WINTERFELDT, 2004).

A teoria da decisão pode ser considerada uma filosofia e, ao mesmo tempo, um processo formal de análise que, geralmente, envolve os seguintes estágios (SAMSON, 1988):

- Identificação do problema e, em decorrência deste, a escolha da estrutura do modelo de decisão;
- Avaliação das probabilidades e das magnitudes envolvidas;
- Uso de um critério de decisão para modelar o processo de seleção de alternativas;

- Análise de sensibilidade, para determinar a consistência das soluções;
- Implementação da estratégia preferida ou escolhida hierarquicamente.

Segundo CLEMEN e REILLY (2001), um processo de decisão é formado pelas seguintes atividades:

- Identificar o problema;
- Identificar objetivos e alternativas;
- Decompor o problema em modelos de estrutura, de incertezas e preferências;
- Escolher a melhor alternativa;
- Analisar a sensibilidade, para determinar a consistência das soluções;
- Decidir se é necessária mais análise;
- Implementar a alternativa escolhida.

Ainda segundo CLEMEN e REILLY (2001), há pelo menos quatro dificuldades para a tomada de uma decisão:

- Complexidade do problema: os problemas a serem resolvidos normalmente são muito complexos, com muitas possibilidades que devem ser consideradas;
- Incerteza quanto à situação e a eventos futuros: geralmente é preciso considerar uma série de eventos que, caso aconteçam, podem influenciar o problema;
- Múltiplos objetivos a serem atingidos: muitas vezes uma decisão é tomada na tentativa de atingir mais de um objetivo, que podem inclusive ser conflitantes, e pode ser difícil conseguir um equilíbrio entre os objetivos;
- Diferentes perspectivas: o problema pode ser encarado de forma diferente dependendo de quem o está resolvendo.

Devido à importância das decisões e de sua complexidade, surgiram os sistemas de apoio à decisão, que são sistemas de informação projetados para apoiar interativamente todas as fases do processo de decisão do usuário. Há várias noções sobre todos os aspectos dessa definição. Os sistemas podem ser para indivíduos ou grupos de usuários. O apoio pode ser direto ou indireto, ou seja, pode ir desde um apoio que considere todas as possibilidades e sugira as melhores opções ao tomador de decisão até a um apoio que apenas forneça informações ao tomador de decisão, para que este as interprete e decida.

Devido às várias perspectivas e dimensões envolvidas no apoio à decisão, a área se desenvolveu em uma variedade de direções diferentes. Essas direções ofereceram diferentes focos e contribuições (MORA *et al.*, 2003).

No início dos anos 70, Little descreveu um cálculo de decisões como um conjunto de procedimentos baseados em um modelo para ajudar gerentes em suas decisões. Ele queria utilizar os melhores modelos da administração através de implementações computacionais efetivas. Little foi capaz de listar os requisitos para um sistema de apoio à decisão bem sucedido: ser simples, robusto, fácil de controlar, adaptativo, completo em questões importantes e fácil de se comunicar com o usuário (NIKAKEN, 2003).

Scott-Morton desenvolveu, em 1971, um sistema de apoio à decisão pioneiro para marketing e planejamento de produção. Juntamente com Gorry, ele deu a primeira definição para um sistema de apoio à decisão. O framework para sistemas de apoio à decisão definido por eles mapeia o potencial do apoio do computador em atividades gerenciais em duas dimensões: o quanto a tarefa é estruturada e o nível de atividade gerencial. Gorry e Scott-Morton viram que, baseado em seu framework, tarefas de decisão poderiam ser divididas de várias formas entre um tomador de decisão humano e um sistema computacional. Em uma situação estruturada, todas as fases da tomada de decisão têm uma organização definida, como um processo, sendo potencialmente automatizáveis e, portanto, os sistemas resultantes são sistemas de tomada de decisão. Em um caso semi-estruturado ou não estruturado, uma ou mais fases do processo de decisão são não-estruturadas, ou seja, podem apresentar algum comportamento inesperado e necessitar de intervenção humana. Em situações parcialmente estruturadas ou não estruturadas, há a necessidade de apoio de modo a aumentar a capacidade do tomador de decisão de melhorar a qualidade do processo decisório. Alguns pesquisadores acham que os sistemas de apoio à decisão são úteis apenas nas partes estruturadas dos problemas de decisão e que as pessoas devem resolver as partes não estruturadas. A linha entre situações estruturadas e não estruturadas se move ao longo do tempo, quando os problemas são melhor entendidos e esse conhecimento os dá estrutura (NIKAKEN, 2003).

Do início dos anos 70 até este momento, a pesquisa dos sistemas de apoio à decisão cresceu significativamente e muitas definições foram apresentadas. A maioria das definições considerou a questão de quão estruturada é a tarefa e o problema de distinguir sistemas de apoio à decisão de outros sistemas de informação gerenciais ou modelos de pesquisa operacional. SPRAGUE e CARLSON (1982) trouxeram à

discussão o contexto organizacional de um sistema de apoio à decisão, fornecendo uma visão prática de como organizações poderiam construir e utilizar um sistema de apoio à decisão (NIKAKEN, 2003).

Em (MORA, 2003), é feita uma ampla análise dos sistemas de apoio à decisão considerando cinco aspectos: fundamentos e arquiteturas de sistemas de apoio à decisão, aplicações de sistemas de apoio à decisão, tecnologia de informação avançada para sistemas de apoio à decisão, avaliação e gerenciamento de sistemas de apoio à decisão e desafios e o futuro dos sistemas de apoio à decisão.

Há vários exemplos de uso de sistemas de apoio à decisão nas mais diferentes áreas de conhecimento:

- Em (GARBER, 2002), é apresentada uma abordagem de apoio à decisão relativa a estruturas flutuantes para a exploração de campos de petróleo no mar;
- Em (ADENSO-DIAZ *et al.*, 2003), é apresentado um sistema de apoio à decisão para re-escalonamento de serviços ferroviários sob a ocorrência de eventos não planejados;
- Em (RINALDI e BAIN, 2003), é apresentado um sistema de apoio à decisão para ajudar a tratar problemas de transporte urbano;
- Em (NIKAKEN, 2003), é apresentada uma ontologia de um sistema de apoio à decisão na informática médica;
- Em (GIBSON, 2003), é apresentada uma abordagem para apoio ao gerenciamento do conhecimento necessário para tomada de decisão na indústria farmacêutica.

Em (FORGIONNE *et al.*, 2003), foi feita uma pesquisa com especialistas, procurando identificar o que os sistemas de apoio à decisão alcançaram até este momento e quais são os desafios e oportunidades para o futuro. De acordo com os especialistas, os sistemas de apoio à decisão serão mais integrados tecnologicamente, irão oferecer um apoio mais amplo e profundo e irão fornecer um conjunto maior de aplicações. No processo, novas informações e tecnologias computacionais serão necessárias, a tarefa do tomador de decisão irá mudar e novas estruturas organizacionais surgirão para atender às mudanças. Em particular, haverá uma evolução na direção de paradigmas de tomada de decisão baseados em grupos. Embora a evolução possa necessitar de investimentos significativos, os benefícios organizacionais de

implantações bem sucedidas de sistemas de apoio à decisão deverão ser significativos e substanciais.

Na próxima seção, aborda-se a questão do apoio à decisão especificamente no contexto do desenvolvimento de software.

3.3 Apoio à Decisão no Desenvolvimento de Software

A disciplina de apoio à decisão na engenharia de software integra inteligência humana e computacional para facilitar a tomada de melhores decisões durante o ciclo de vida do software (RUHE, 2003b).

A tomada de decisão é uma atividade importante e complexa na engenharia de software. Normalmente, é um processo não sistemático, uma vez que tipicamente se baseia em experiência pessoal sem o uso de modelos explícitos (RUS *et al.*, 2002).

O apoio à decisão na engenharia de software é de grande interesse tanto para a academia quanto para a indústria. Decisões precisam ser tomadas durante todas as iterações de um ciclo de vida de software. Atualmente, muitas dessas decisões cruciais são tomadas de forma *ad hoc*, baseadas em impressões e sem ligação com os melhores conhecimentos, modelos e experiências. Quanto mais cedo no ciclo de vida decisões ruins são tomadas, maior será seu impacto na qualidade do software. As principais características da tomada de decisão durante as fases iniciais do ciclo de vida do software são (RUHE, 2003b):

- A quantidade e qualidade das informações disponíveis são, normalmente, baixas;
- Os processos e os parâmetros para as decisões estão mudam dinamicamente;
- O número de interesses conflitantes de diferentes pessoas envolvidas com diferentes objetivos e restrições deve ser balanceado e otimizado.

O apoio à decisão inteligente é útil, principalmente, em situações caracterizadas pelos seguintes fatores: complexidade, incerteza, presença de várias pessoas interessadas, grande quantidade de dados e problemas com parâmetros ou informações relacionadas que mudam rapidamente. Apoio, aqui, significa fornecer acesso à informação, que seria de outra forma inacessível ou difícil de obter, facilitando a geração e avaliação de soluções distintas e priorizando alternativas através do uso de modelos explícitos que forneçam a estrutura para decisões particulares (RUHE, 2003b).

CLEMEN e REILLY (2001) identificam algumas razões pelas quais as decisões são difíceis, conforme já descrito anteriormente na seção 3.2. Essas razões se aplicam ao desenvolvimento de software (RUS *et al.*, 2002):

- O desenvolvimento de software é um processo muito complexo, com muitas atividades, artefatos, atores e interações;
- Uma vez que é muito dependente das pessoas, o processo e seus produtos envolvem muita incerteza (pessoas saem, clientes mudam os requisitos, fornecedores não cumprem promessas, etc);
- Software deve ser construído mais rápido, mais barato e com maior qualidade;
- As pessoas tomam decisões de acordo com sua experiência, intuições, crenças pessoais e nas informações que têm disponíveis (que, normalmente, são imperfeitas, insuficientes ou imprecisas).

Os elementos básicos de um problema de decisão são: decisões a serem tomadas, eventos incertos e a valoração dos resultados. Para um gerente de projetos de software, o alvo é, muitas vezes, o fim do projeto; idealmente, o resultado do desenvolvimento de software seria software de alta qualidade, entregue rapidamente e dentro do orçamento. Realisticamente, a satisfação do cliente, *time to market* e os custos devem ser priorizados, de acordo com a estratégia da empresa e com os objetivos do negócio (RUS *et al.*, 2002).

No CMMI (SEI, 2002), *Capability Maturity Model Integration*, é definida a área de processo Análise de Decisão e Resolução, no nível três de maturidade. O objetivo é analisar possíveis decisões usando um processo de avaliação formal, que analisa alternativas identificadas contra critérios estabelecidos. Um processo de avaliação formal envolve as seguintes ações:

- Estabelecer os critérios para a avaliação de alternativas;
- Selecionar métodos para avaliar as alternativas;
- Identificar as soluções alternativas;
- Avaliar as soluções alternativas usando os critérios e métodos estabelecidos;
- Selecionar as soluções recomendadas dentre as alternativas, baseando-se nos critérios de avaliação.

Em (RUHE, 2003a), foram definidos os requisitos considerados pelo autor como ideais para sistemas de apoio à decisão que combinam os recursos intelectuais de indivíduos e organizações com as capacidades do computador para melhorar a efetividade, eficiência e transparência da tomada de decisão. São eles:

- Registro, modelagem e gerência da experiência do corpo de conhecimento existente na área do problema (na organização em questão);
- Integração com os sistemas de informação organizacionais existentes;
- Orientação de processo no apoio a decisão: por exemplo, considerar o processo como as decisões são tomadas e qual seria o impacto no desenvolvimento e nos processos de negócio;
- Modelagem de processo e simulação para planejar, descrever, monitorar, controlar e simular os processos e rastrear mudanças em seus parâmetros e dependências;
- Recuperação e entendimento evolutivo dos comprometimentos;
- Apresentação e explicação do conhecimento gerado e das alternativas de solução de várias formas customizáveis para aumentar a transparência;
- Análise e decisão, consistindo de um conjunto de métodos e técnicas para avaliar e priorizar as alternativas de solução geradas e encontrar *trade-offs* entre os objetivos conflitantes e interesses dos envolvidos;
- Apoio à recuperação e descoberta de conhecimento;
- Apoio à comunicação eletrônica, agendamentos, compartilhamento de documentos e acesso à opinião de especialistas.

Várias abordagens foram propostas para apoiar a tomada de decisão em projetos de software, tais como: executar estudos experimentais, coletar dados e construir modelos a partir dos resultados da análise dos dados, construir bases de conhecimento e experiências e incrementá-las com tecnologias apropriadas para a aquisição, armazenamento, recuperação e disseminação do conhecimento. Todos esses sistemas se propõem a apoiar e não a substituir seus usuários. A decisão final continua sendo feita por humanos (RUS *et al.*, 2002).

Podem ser citados diversos exemplos de técnicas ou ferramentas de apoio à decisão, no contexto do processo de desenvolvimento de software:

- Em (GERSON, 1992), aborda-se o uso de sistemas de apoio à decisão e seu uso em tomadas de decisão estratégicas;
- Em (COLLOFELLO *et al.*, 1998), é proposto um simulador de processo baseado em dinâmica de sistemas para apoio à decisão de políticas de alocação de pessoal em projetos de software;
- Em (BARROS, 2001), é proposto o gerenciamento de projetos baseado em cenários, usando modelagem dinâmica e simulação com o intuito de apoiar as decisões gerenciais;
- Em (CENTENO E CARRILLO, 2001), são abordados os desafios de se introduzir a simulação como uma ferramenta de apoio à decisão;
- Em (KARLSTRÖM e RUNESON, 2002), é proposta uma abordagem para apoio à decisão no que diz respeito à introdução de *Extreme Programming* e à seleção de suas práticas;
- Em (RAFFO *et al.*, 2002), é apresentado um *framework* de apoio à decisão que integra métricas com modelos de simulação do processo de desenvolvimento para apoiar na função de controle do gerenciamento do projeto de software;
- Em (RUHE *et al.*, 2002), é apresentado um método de apoio à decisão em negociação de requisitos (*Quantitative Win-Win*), que inclui métodos quantitativos como base para decisões melhores e mais objetivas;
- Em (RUS *et al.*, 2002), é proposta uma abordagem de combinação entre estudos experimentais e simulação de processo para apoiar a análise de decisão no desenvolvimento de software;
- Em (DONZELLI *et al.*, 2003), é abordada a questão do apoio à decisão para o uso de inspeções de software;
- Em (RUS *et al.*, 2003), aborda-se a questão da tomada de decisão no planejamento de verificação e validação.

Uma das possíveis maneiras de se representar um problema de decisão é descrevê-lo como um problema de satisfação de restrições. Muitos problemas relacionados ao desenvolvimento de software podem ser encarados como problemas de satisfação de restrições, onde há muitas alternativas a selecionar com base em um conjunto de restrições. A dificuldade maior em se fazer isso é dispor do conhecimento necessário para determinar essas restrições e o que as satisfaz. Por exemplo, na seleção de um modelo de ciclo de vida, é difícil determinar quais características exigidas pelo software

são contempladas por cada um dos modelos existentes. No caso da seleção de profissionais para uma atividade de um projeto de software, é difícil determinar quais características o profissional deve possuir para desempenhar a atividade de forma adequada. Caso essas relações possam ser determinadas, é possível que a seleção entre diversas opções possa ser apoiada por algoritmos de satisfação de restrições. Uma alternativa para a falta de conhecimento entre as relações existentes é confiar no conhecimento e experiência do gerente de projetos para determinar esses fatores.

Podemos citar vários exemplos no desenvolvimento de software de problemas que podem ser considerados de satisfação de restrições, tais como:

- A seleção das atividades que irão fazer parte de um processo de software em um determinado projeto. Um projeto possui uma série de requisitos funcionais e não funcionais, que podem ser encarados como sendo as restrições. Incluir ou excluir atividades pode implicar em satisfazer ou não as restrições impostas, dado que cada inclusão de atividade pode contribuir para o atendimento de um ou mais requisitos;
- A seleção de fornecedores de produtos de software para um projeto ou organização como um todo. Cada fornecedor tem suas características e o contratante exige que seus fornecedores tenham um certo conjunto de características (restrições). Resolver o problema é conseguir a melhor combinação possível entre as características necessárias e as dos fornecedores;
- Seleção de profissionais para contratação. O contratante espera que os profissionais a serem contratados possuam uma série de características (restrições). Cada profissional tem suas características. Resolver o problema é conseguir a melhor combinação possível entre esses fatores;
- Seleção de profissionais na organização para desempenharem atividades de um projeto. Neste trabalho, pretende-se apoiar a resolução deste problema. Cada atividade requer que o profissional possua uma série de características (restrições). Cada profissional tem suas características. Resolver o problema é, como no caso anterior, conseguir a melhor combinação possível entre esses fatores;
- Diversos outros, tais como: a seleção de componentes de software a serem utilizados no projeto, seleção de técnicas a serem utilizadas nas mais diferentes fases do projeto com base nos requisitos de cada fase, entre outros.

Na próxima seção, são apresentados os problemas de satisfação de restrições, uma vez que se acredita poder modelar problemas de decisão dessa forma.

3.4 Problemas de Satisfação de Restrições

Muitos problemas na área de inteligência artificial (GAVANELLI, 2002) e na área de engenharia, como o problema do escalonamento de processos, podem ser classificados como sendo problemas de satisfação de restrições (CHU e NGAI, 1988). Além desses exemplos, há aplicações para os problemas de satisfação de restrições nas áreas de computação gráfica, sistemas de bancos de dados, biologia molecular, aplicações comerciais, engenharia elétrica, projeto de circuitos, alocação de recursos, entre outros.

Um problema de satisfação de restrições pode ser considerado como uma tupla $S = (V, D, R)$, onde $V = \{x_1, x_2, \dots, x_n\}$ é um conjunto finito de variáveis, também chamadas de variáveis de domínio finito, $D = \{D_1, D_2, \dots, D_n\}$ é um conjunto finito de domínios e R é um conjunto de restrições, limitando os valores que as variáveis podem receber simultaneamente. A cada variável x_i está associado um conjunto finito D_i com todos os possíveis valores que podem ser assumidos pela variável.

A solução de um problema de satisfação de restrições é uma atribuição de valores a todas as variáveis de V , de forma que todas as restrições de R sejam satisfeitas. Normalmente, há várias soluções que satisfazem às restrições definidas. Elas têm diferentes qualidades, definidas por funções de custo associadas. Em muitos problemas, interessa a solução ótima, que minimiza ou maximiza sua função de custo (KUCHCINSKY, 2003).

Um problema clássico que pode ser encarado como um problema de satisfação de restrições é o problema das N Rainhas. O problema consiste em achar todas as formas de se colocar N rainhas em um tabuleiro de tamanho N por N (N linhas por N colunas), de forma que nenhuma delas ataque outra. Uma rainha ataca outra rainha se está na mesma linha, na mesma coluna, ou em uma das diagonais de outra rainha.

O problema das N -Rainhas é bastante antigo e já é estudado há mais de um século. Originalmente conhecido como o problema das Oito Rainhas, ele foi estudado por muitos matemáticos famosos, incluindo o matemático alemão Karl Friedrich Gauss (1777-1855). O problema foi generalizado para N , em 1850, por Franz Nauck. Desde 1960, com os rápidos avanços na computação, o problema tem sido usado como

exemplo de algoritmos de *backtracking*, da geração de permutações, do paradigma dividir para conquistar, dos problemas de satisfação de restrições, de programação inteira e outros.

Vários algoritmos já foram propostos para resolver problemas de satisfação de restrições. Uma pesquisa sobre isso é feita em (TSANG, 1993).

Um dos algoritmos que podem ser usados é o algoritmo de *backtracking* (GOLOMB e BAUMERT, 1965). No *backtracking*, valores dos domínios são atribuídos às variáveis, um de cada vez. A ordem na qual os valores são atribuídos às variáveis é definida pelo algoritmo. Depois que um valor é atribuído a uma variável, é feita uma checagem de consistência para determinar se o valor foi atribuído corretamente, ou seja, se está de acordo com as restrições. Se sim, o processo de atribuição continua, ou seja, é atribuído um valor a uma outra variável. Se não, um valor alternativo é selecionado e atribuído à variável e a checagem de consistência é feita novamente. Caso se chegue a um ponto em que a variável não pode receber valor nenhum que seja consistente com as restrições, o processo precisa retornar para uma outra variável e deve ser tentado outro valor para essa variável. A busca continua até que uma solução seja encontrada ou todas as alternativas sejam tentadas (CHU e NGAI, 1988). Achar uma solução significa que foi possível atribuir valores a todas as variáveis respeitando as restrições.

O problema do *backtracking* é que para muitos problemas pode haver um número grande de variáveis, cada uma com um grande número de valores em seu domínio. Como resultado, o espaço total de busca pode se tornar muito grande. Um exemplo de busca sem necessidade que o *backtracking* realiza é quando as atribuições feitas até um certo ponto inviabilizam a atribuição de uma variável que será considerada mais adiante. Assim, o algoritmo irá tentar uma série de atribuições até descobrir que não há solução. Portanto, para uma busca mais eficiente, o ideal é que o algoritmo detecte que uma solução não é possível o mais cedo possível e possa tentar um outro caminho.

Uma forma de melhorar o desempenho do *backtracking* é através do uso do *forward checking* (HARALICK e ELLIOTT, 1980). Nesse algoritmo, toda vez que um valor é atribuído, o algoritmo exclui dos domínios das outras variáveis os valores que não podem mais ser atribuídos. Se algum dos domínios esvazia, então a busca volta imediatamente, evitando o problema citado acima. *Forward Checking*, normalmente é executado mais rapidamente que o *backtracking* simples e é, geralmente, simples de implementar.

Forward checking é um caso especial da checagem de consistência de arcos (RUSSELL e NORVIG, 2003). Um estado é arco-consistente quando cada variável tem um valor em seu domínio que é consistente com cada uma das restrições para aquela variável. A consistência de arco pode ser obtida através de exclusões sucessivas de valores que sejam inconsistentes com alguma restrição. Ao serem excluídos os valores, outros valores podem se tornar inconsistentes porque dependiam dos valores excluídos. A consistência de arco, portanto, é uma forma de propagação de restrições, uma vez que as escolhas disponíveis vão gradualmente diminuindo. Em alguns casos, atingir a consistência de arco é suficiente para resolver o problema, porque todas as variáveis podem passar a ter apenas um valor em seu domínio. A consistência de arco, normalmente, é usada como um passo de pré-processamento, mas pode também ser usada durante a busca.

Caso se utilize *backtracking* para resolver um problema de satisfação de restrições, outra questão importante é a ordem na qual as variáveis são escolhidas para receberem valores. Experimentos e análises de vários pesquisadores mostram que esta ordem pode ter impacto substancial na complexidade da busca. Muitas heurísticas foram desenvolvidas e analisadas para selecionar a ordem das variáveis. Uma heurística muito eficiente, desenvolvida por BITNER e REINGOLD (1975), é normalmente usada com o algoritmo *forward checking*. Nessa heurística, a variável com o menor número de valores possíveis restantes no domínio é selecionada para receber valor. Essa heurística foi chamada *search-rearrangement method*, ou método de reorganização de busca (também chamada de *most constrained variable*, ou variável mais restrita). Purdom e Brown estudaram extensivamente essa heurística, assim como suas variantes, tanto experimentalmente como analiticamente (PURDOM 1983; PURDOM e BROWN 1981, 1982; PURDOM *et al.*, 1981). Seus resultados mostram que para várias classes de problemas, *backtracking* com a reorganização de busca é uma melhoria substancial sobre o método de *backtracking* simples. Para o problema das N-Rainhas, STONE e STONE (1986) mostraram experimentalmente que essa heurística levou a uma melhoria de dezenas de ordens de magnitude para valores grandes de N. Com essa heurística, foram capazes de resolver o problema para $N \leq 96$. O *backtracking* simples não era capaz de resolver nem mesmo para $N = 30$ (KUMAR, 1992).

Uma outra heurística, também bastante efetiva, é a *most-constraining-variable*. Ela tenta reduzir o fator de ramificação em escolhas futuras, escolhendo as variáveis que

estão envolvidas no maior número de restrições em outras variáveis ainda sem valor atribuído (RUSSELL e NORVIG, 2003).

Uma vez que a variável que irá receber o valor seja escolhida, vários valores em seu domínio podem ser atribuídos. A ordem em que esses valores é considerada, também, pode ter um grande impacto na obtenção da solução. Uma possível heurística é preferir aqueles valores que maximizam o número de opções disponíveis para futuras atribuições (HARALICK e ELLIOTT, 1980). Ao incorporar essa heurística no algoritmo de Stone e Stone para resolver o problema das N-Rainhas, KALE (1990) desenvolveu um algoritmo baseado em *backtracking* que pode ser usado para resolver o problema para valores grandes de N (= 1000). Sem a heurística, o algoritmo era incapaz de resolver o problema para N pouco maior que 100 (KUMAR, 1992). Essa heurística é chamada de *least-constraining-value* em (RUSSELL e NORVIG, 2003).

Até o momento, foi considerado o caso em que existe pelo menos uma solução para o problema e se deseja encontrar uma solução. No entanto, em muitas aplicações importantes as restrições podem ser muito fortes, de forma que uma solução completa não seja possível. Nesse caso, soluções parciais ainda podem ser úteis se um número suficiente das restrições mais importantes for satisfeito. Um exemplo desse tipo de problema é o problema da máxima satisfação de restrições, no qual o objetivo é achar atribuições de valores a variáveis de modo a atender ao número máximo de restrições possíveis (WALLACE e FREUDER, 1996).

Em alguns casos, achar uma solução viável não é o bastante e uma preferência entre as soluções deve ser expressa. Essa preferência, normalmente, é dada por uma função que associa custos às soluções. O algoritmo comumente usado para resolver esse tipo de problema é o *Branch-and-Bound* (LAWLER e WOOD, 1966). O *Branch-and-Bound* pode ser definido, em linhas gerais, como sendo um algoritmo que acha a solução ótima, mantendo a melhor solução encontrada até o momento. Se alguma solução parcial não pode melhorar a melhor solução, é abandonada a solução parcial.

Há muitos exemplos de utilização das técnicas que foram apresentadas nesta seção. Em (KUCHCINSKY, 2003), é proposto o escalonamento e alocação de recursos baseados em restrições. É feito uso de *forward-checking*, da heurística *most-constrained-variable*, e para otimização uma adaptação do *branch-and-bound*. Em (CHU e NGAI, 1988), é proposta uma abordagem de resolução de problemas de satisfação de restrições baseada em um algoritmo de busca ordenada dinâmico orientado a restrições. A abordagem proposta é extremamente similar ao uso de *forward checking*

e *most-constrained-variable*. Em (GAVANELLI, 2002), são abordados problemas de satisfação de restrições interativos para a visão artificial.

Dentro da engenharia de software, também, é possível identificar o uso de problemas de satisfação de restrições em alguns trabalhos. Em (ALBIN-AMIOT *et al.*, 2001), são utilizados problemas de satisfação de restrições para a identificação de padrões de projeto em código fonte de software. Em (WOODS e YANG, 1996), é proposta a solução de problemas de entendimento de programas (BALMAS *et al.*, 2000) como problemas de satisfação de restrições. Em (SY e DEVILLE, 2003), é apresentada uma abordagem para geração automática de dados para testes de programas, fazendo uso da satisfação de restrições.

3.5 Conclusão

Os sistemas de apoio à decisão têm sido cada dia mais utilizados. Sua importância e o apoio que fornecem também tem crescido bastante.

Há diversas formas de modelar problemas de decisão e de implementar sistemas de apoio à decisão. Uma forma bastante utilizada é a modelagem de situações de decisão como problemas de satisfação de restrições, caso suas características permitam isso. Neste trabalho, acredita-se que modelar o problema da alocação de recursos humanos como um problema de satisfação de restrições e resolvê-lo de forma a dar apoio ao gerente nessa tarefa seja uma boa abordagem.

Neste capítulo, foi abordada a questão das decisões e da análise de decisões, bem como o apoio à decisão de uma forma geral. Foi considerada, também, a questão do apoio à decisão no contexto do desenvolvimento de software, procurando explicar sua necessidade, características e citar exemplos de uso. Foram ainda abordados também os problemas de satisfação de restrições, que são muito usados na área de inteligência artificial e em muitas outras áreas. Explicou-se o que são eles, em que são usados, como podem ser resolvidos e algumas de suas variações.

No próximo capítulo será apresentada a modelagem do problema da alocação de recursos humanos como um problema de satisfação de restrições e a solução para ele proposta neste trabalho.

CAPÍTULO 4

Modelagem e Resolução do Problema de Alocação de Recursos Humanos em Projetos de Software como um Problema de Satisfação de Restrições

4.1 Introdução

A alocação de recursos humanos em um projeto é uma das principais atividades a serem realizadas pelo gerente no processo de desenvolvimento de software. Essa tarefa nem sempre é simples: são muitos fatores a considerar, muitas combinações diferentes de equipes podem ser geradas, muitas restrições impostas pela qualificação dos profissionais e períodos em que estarão disponíveis contra a qualificação necessária para executar cada tarefa e o período em que a tarefa precisa ser executada, conforme visto no capítulo 2. Dessa forma, supõe-se que fornecer apoio ao gerente de projetos na atividade de composição de equipe para um projeto de software seja de grande ajuda para a execução da atividade, uma vez que alguns dos passos para a solução do problema podem ser automatizados, possivelmente diminuindo o esforço necessário para sua execução. Além disso, é de se esperar que o apoio forneça ao gerente um melhor embasamento para a tomada de decisão, uma vez que possibilita que seja considerado um universo maior de fatores e alternativas. O apoio sugerido aqui diz respeito a propor ao gerente, dado um conjunto de restrições de projeto, profissionais disponíveis, tempo e orçamento, uma equipe que satisfaça a essas restrições e que otimize algum dos fatores envolvidos no problema. Caso não seja possível satisfazer todas as restrições, a abordagem proposta deve apresentar a equipe que mais se aproxime de satisfazer as restrições impostas.

Neste capítulo, é apresentada a abordagem de apoio à decisão gerencial na alocação de recursos humanos em projetos de software através da modelagem e resolução do problema como um problema de satisfação de restrições. A seção a seguir (4.2) apresenta a modelagem do problema de alocação de recursos humanos em projetos de software como um problema de satisfação de restrições. A seção 4.3 apresenta a resolução proposta para o problema, detalhando os algoritmos utilizados e as variações propostas para a solução do problema. A seção 4.4 apresenta um exemplo de uma

situação de uso da abordagem proposta. A seção 4.5, por fim, apresenta as conclusões deste capítulo.

4.2 Definição do Problema de Alocação de Recursos Humanos em Projetos de Software

Diversos problemas, nas mais variadas áreas, são modelados e resolvidos como problemas de satisfação de restrições, como visto no capítulo anterior. Modelar um problema dessa forma tem a vantagem de possibilitar a utilização de todo o conhecimento que vem sendo adquirido ao longo dos anos sobre essa classe de problemas. Dessa forma, podem ser utilizadas técnicas e algoritmos que têm sido propostos e avaliados para esse tipo de problema, sem a necessidade do desenvolvimento de novas técnicas e algoritmos específicos para um problema em questão.

O problema da alocação de recursos humanos envolve, basicamente, um conjunto de profissionais e um conjunto de atividades. Cada profissional possui uma série de características (habilidades, conhecimentos, experiências, formação acadêmica, certificações, posição na organização, etc), cada qual com uma determinada intensidade (por exemplo, grande conhecimento em banco de dados, noções de linguagens de programação, entre outros). Além disso, profissionais estão indisponíveis em diferentes períodos do tempo. Estar indisponível em um período significa que o profissional não pode ser alocado a nenhuma atividade de projeto naquele período. Essa indisponibilidade pode ser por diversos motivos, entre eles férias, alocação a outra atividade no mesmo período, entre outros.

Outra questão importante para o problema é a produtividade, que pode variar de profissional para profissional, influenciando no tempo necessário para a conclusão das tarefas. Neste trabalho, será dada ao gerente a possibilidade de escolher diferentes modelos de produtividade, ou seja, diferentes formas de influenciar a duração das atividades pela produtividade dos profissionais.

A primeira opção é não utilizar modelo algum, caso o gerente julgue que os modelos oferecidos são inadequados, ou simplesmente não deseje considerar este fator. Dessa forma, a produtividade não será considerada no problema e as atividades terão duração fixa.

A segunda opção é utilizar um modelo fixo. Neste caso, o gerente não poderá configurar os índices de produtividade dos profissionais e esses índices serão calculados automaticamente. Será considerada uma adaptação do modelo de ABDEL-HAMID e MADNICK (1991), que determina que um desenvolvedor experiente produz o dobro que um desenvolvedor inexperiente. Tal constatação é comprovada pelo resultado de entrevistas e por uma pesquisa na literatura de Engenharia de Software, conforme apresentado no livro que descreve o modelo. Na adaptação sugerida, assumimos que um desenvolvedor que possui as características requeridas na intensidade exata em que são exigidas por uma atividade é capaz de realizar esta atividade no prazo estipulado e tem produtividade 1. A partir desta premissa, sendo X um modificador de produtividade, consideramos que:

$$\text{Produtividade Máxima: } 1 + X$$

$$\text{Produtividade Mínima: } 1 - X$$

Pelo modelo de ABDEL-HAMID e MADNICK (1991), temos:

$$(1 - X) \cdot 2 = 1 + X$$

$$2 - 2X = 1 + X$$

$$X = 1/3 = 0.333$$

Assim:

$$\text{Produtividade Máxima: } 1 + X = 1.333$$

$$\text{Produtividade Mínima: } 1 - X = 0.667$$

$$\text{Produtividade} = (1 - 0.337) + 2 \cdot 0.337 \cdot \text{Experiência}$$

Considera-se que a experiência de um profissional varia de acordo com as características possuídas pelo profissional que excedem as características exigidas pela atividade. Essa diferença faz com que a experiência varie entre 0.5 (caso em que o resultado da expressão de produtividade acima seria 1) e 1 (produtividade máxima). Assim sendo, um profissional com produtividade 1 é capaz de realizar a atividade no prazo planejado. Conforme essa produtividade cresce, mais rápido o profissional termina a tarefa. Assim, o novo período da atividade passa a ser o número de dias planejado, dividido pela produtividade do profissional que a executa. Essa alteração pode fazer com que as pós atividades da atividade que teve seu período alterado possam ser iniciadas mais cedo, o que levaria a uma redução no tempo do projeto.

A terceira opção de modelo de produtividade permite que o gerente determine, para cada profissional, sua produtividade para todas as atividades. Isso permite que o gerente indique o valor da produtividade de cada profissional, procurando adequar esses valores

a realidade da organização e do projeto em questão. Neste caso, considera-se que a produtividade do profissional será a mesma para qualquer tarefa que este execute.

A quarta, e última opção, permite que o gerente determine a produtividade de cada profissional para cada atividade. Esta opção, como a anterior, permite que o gerente ajuste os valores da produtividade de acordo com seu sentimento a respeito da situação específica.

O custo da hora de trabalho do profissional e o número de horas, por dia, por ele trabalhadas são também informações relevantes ao problema da alocação, uma vez que essas informações influem, respectivamente, no custo do projeto e na disponibilidade do profissional para alocação. Cada atividade, por sua vez, possui um conjunto de características exigidas, ou seja, quais características um profissional precisa possuir para executar a tarefa e, se relevante, em que intensidade. Além disso, uma atividade indica o período em que precisa ser realizada (informação essa contida no cronograma do projeto) e quantas horas por dia precisam ser dedicadas a sua execução nesse período.

Com base nessas informações, o problema da alocação de recursos humanos implica em atribuir um profissional a cada atividade, levando em consideração as seguintes regras:

- Para que um profissional seja atribuído a uma atividade, precisa possuir todas as características exigidas pela atividade, em intensidade igual ou superior ao exigido;
- Para que um profissional seja atribuído a uma atividade, este não pode estar indisponível no período em que a atividade será executada;
- Um profissional está indisponível para realizar uma atividade:
 - Se houver alguma indisponibilidade registrada para o período da atividade (férias, licença, viagem, etc);
 - Se estiver alocado a outra atividade no período e o número de horas trabalhadas por dia nessa atividade for igual ao número de horas máximo trabalhado por dia pelo profissional;
 - Se estiver alocado a outra atividade no período e o número de horas trabalhadas por dia nessa atividade for menor que o número de horas máximo trabalhado por dia pelo profissional, mas o número de horas

restante não for suficiente para alocar o profissional na atividade desejada no período.

Expressando de maneira formal:

- Seja P um conjunto de projetos. Cada elemento $P_i \in P$ é composto por um nome e por um conjunto de atividades (A_{ij});

$$P = \{ P_i \}$$

$$P_i = [\text{nome}, \{A_{ij}\}]$$

- Seja RH um conjunto de profissionais da organização que podem ser alocados a atividades de seus projetos. Cada $RH_i \in RH$ é composto por um nome, pelo valor da hora de trabalho, pela quantidade de horas trabalhadas por dia pelo profissional, por um conjunto de períodos de indisponibilidade (PI_{ij}) e por um conjunto de características possuídas (CRH_{ij});

$$RH = \{ RH_i \}$$

$$RH_i = [\text{nome}, \$\text{hora}, \#\text{hora}, \{PI_{ij}\}, \{CRH_{ij}\}]$$

- Seja A_i um conjunto de atividades que pertence a um dado projeto P_i . Cada $A_{ij} \in A_i$ é composta por um nome, pelas datas inicial e final em que a atividade precisa ser realizada, pelo número de horas que devem ser dedicadas por dia à atividade, por um conjunto de pré-atividades (PA_{ijk}), e por um conjunto de características requeridas para que o profissional realize a atividade (CA_{ijk}).

$$A_{ij} = [\text{nome}, \text{data_inicio}, \text{data_final}, \#\text{horas}, \{PA_{ijk}\}, \{CA_{ijk}\}]$$

- Seja PI_i um conjunto de períodos de indisponibilidade que pertence a um dado profissional RH_i . Cada $PI_{ij} \in PI_i$ é composto pelas datas inicial e final do período e pela quantidade de horas por dia no período em que o profissional estará indisponível.

$$PI_{ij} = [\text{data_inicio}, \text{data_final}, \#\text{horas}]$$

- Seja C um conjunto de características que pertence a um profissional RH_j ou que são demandadas para a execução de uma atividade A_k . Cada $C_i \in C$ é composto por um nome, um valor máximo que a característica pode assumir e o valor (apresentado, no caso do profissional, ou demandado, no caso da atividade) da característica.

$$C_i = [\text{nome}, \#\text{máximo}, \#\text{requerido/apresentado}]$$

Assim, resolver o problema é encontrar soluções para a expressão descrita a seguir:

$$\forall A_{ij} \in Ai \subset Pi \in P(\exists RH_k \in RH(\alpha \wedge \beta))$$

A formulação acima indica que, para toda atividade que pertence ao conjunto de atividades de um determinado projeto que pertence ao conjunto de projetos da organização, existe algum profissional que pertence ao conjunto de profissionais da organização para o qual as condições α e β são verdadeiras. A primeira condição é a seguinte:

$$\alpha \leftarrow \forall (C_m \in C \subset A_i) (\exists ((C_l \in C \subset RH_k) \wedge C_m = C_l \wedge Valor(C_l) \geq Valor(C_m)))$$

A fórmula acima estabelece que, para toda característica que faz parte do conjunto de características exigidas por uma atividade, existe uma característica que faz parte do conjunto de características possuídas por um profissional, de forma que as duas características são, na verdade, a mesma e o valor da característica no profissional é maior ou igual ao valor exigido na atividade.

A segunda condição expressa na primeira fórmula é a seguinte:

$$\beta \leftarrow \neg \exists (PI_n \in PI \subset RH_k) \left(\begin{array}{l} ((DataIni(A_{ij}) \geq DataIni(PI_n)) \wedge \\ (DataIni(A_{ij}) \leq DataFim(PI_n)) \\ \vee \\ (DataFim(A_{ij}) \geq DataIni(PI_n)) \wedge \\ (DataFim(A_{ij}) \leq DataFim(PI_n)) \\ \vee \\ (DataIni(A_{ij}) \leq DataIni(PI_n)) \wedge \\ (DataFim(A_{ij}) \geq DataFim(PI_n)) \\ \wedge MaxHorasDia(RH_k) < NumHoras(PI_n) + NumHoras(A_{ij})) \end{array} \right)$$

A expressão acima estabelece que não existe um período de indisponibilidade para o profissional tal que o período de indisponibilidade esteja compreendido todo ou em parte dentro do período em que a atividade precisa ser realizada. E que, além disso, o número máximo de horas trabalhadas por dia pelo profissional seja menor que o número de horas de indisponibilidade por dia, somado ao número de horas que devem ser dedicados por dia à atividade.

Conforme mostrado no capítulo 3, um problema de satisfação de restrições pode ser considerado como uma tupla $S = (V, D, R)$. Dessa forma, pode-se modelar o problema da alocação de recursos humanos em projetos de software como sendo um problema de satisfação de restrições no qual as atividades são modeladas como variáveis (V), os

valores que essas variáveis podem assumir são os profissionais da organização (D) e todas as regras envolvidas (de qualificação e de data) são as restrições (R). Resolver o problema implica em escolher profissionais (valores) para todas as atividades (variáveis), de forma a satisfazer todas as regras definidas para o problema (restrições).

Dessa forma, pode-se considerar R como sendo as regras expressas nas equações acima. V e D podem ser definidos, formalmente, da seguinte maneira:

$$D = \{ D_i \} \therefore D_i = [\{RH_{ij}\}]$$

$$V = \{ V_i \} \therefore V_i = [A_{ij}, D_{ij}]$$

Normalmente, há várias soluções para um problema de satisfação de restrições. Elas têm diferentes qualidades, definidas por funções de custo. Em muitos problemas, interessa uma solução ótima, que minimiza ou maximiza sua função de custo. No problema da alocação de recursos humanos em projetos de software, podem ser definidas algumas funções de custo com base em fatores do problema. Cada função de custo terá uma solução ótima própria, potencialmente distinta das soluções ótimas geradas para outras funções de custo. Neste trabalho, são consideradas as seguintes funções:

- Menor Custo: esta função é responsável por determinar a alocação que satisfaz a todas as restrições do problema e, além disso, representa o menor custo para o projeto. O custo é calculado a partir do valor da hora de trabalho dos profissionais alocados, multiplicado pelo número de horas que devem ser dedicadas às atividades do projeto:

- Seja E o conjunto de possíveis equipes que satisfazem a todas as restrições. Seja $Custo(E_k)$ uma função que retorne o custo de uma dada equipe E_k para o projeto. A equipe selecionada $E_i \in E$ é escolhida de acordo com a seguinte equação:

$$\neg \exists (E_j \in E) (Custo(E_j) < Custo(E_i))$$

- Menor Equipe: esta função é responsável por determinar a alocação que satisfaz a todas as restrições do problema e, além disso, representa a equipe em que menos profissionais estão alocados ao projeto:

- Seja E o conjunto de possíveis equipes que satisfazem a todas as restrições. Seja $Tamanho(E_k)$ uma função que retorna o tamanho (número de diferentes membros) de uma equipe E_k . A equipe selecionada $E_i \in E$ é escolhida de acordo com a seguinte equação:

$$\neg \exists (E_j \in E) (Tamanho(E_j) < Tamanho(E_i))$$

- Menor Sub-Alocação: esta função é responsável por determinar a alocação que satisfaz a todas as restrições do problema e, além disso, representa a equipe em que há o menor “índice de sub-alocação” no projeto. Esse índice é baseado no valor numérico dos valores das características exigidas pelas atividades e possuídas pelos profissionais. Cada valor de característica exigida pela atividade e que seja possuído pelo profissional excedendo o valor exigido pela atividade soma essa diferença no índice. É um índice relativo, uma vez que nem todas as características possuem o mesmo valor máximo:

- Seja E o conjunto de possíveis equipes que satisfazem a todas as restrições. Seja $DistQual(E_k)$ uma função que retorna a distância entre a qualificação necessária e a possuída por uma equipe E_k . A equipe selecionada $E_i \in E$ é escolhida de acordo com a seguinte equação:

$$\neg \exists (E_j \in E) (DistQual(T_j) < DistQual(T_i))$$

- Maior Qualificação: esta função é responsável por determinar a alocação que satisfaz a todas as restrições do problema e, além disso, representa a equipe em que as melhores pessoas disponíveis foram alocadas. Nesta função, utiliza-se o mesmo índice de sub-alocação, só que, ao invés de selecionar a equipe com o menor índice, é selecionada a equipe com o maior valor para esse índice, ou seja, a maior sub-alocação:

$$\neg \exists (E_j \in E) (DistQual(T_j) > DistQual(T_i))$$

- Menor Tempo: esta função é responsável por determinar a alocação que satisfaz a todas as restrições do problema e, além disso, representa a equipe que é capaz de concluir o projeto mais rapidamente. Nesta função, utiliza-se um dos modelos de produtividade descritos acima:

- Seja E o conjunto de possíveis equipes que satisfazem a todas as restrições. Seja $Duracao(E_k)$ uma função que retorna a duração esperada do projeto dada uma equipe E_k . A equipe selecionada $E_i \in E$ é escolhida de acordo com a seguinte equação:

$$\neg \exists (E_j \in E) (Duracao(E_j) < Duracao(E_i))$$

Conforme dito no capítulo 3, em muitas aplicações importantes, as restrições podem ser muito fortes, de forma que uma solução completa não seja possível. Nesse caso, soluções parciais ainda podem ser úteis se um número suficiente das restrições mais importantes for satisfeito. No caso do problema da alocação de recursos humanos em projetos de software, essa situação pode ocorrer, ou seja, as restrições das atividades não poderem ser satisfeitas pelos profissionais disponíveis na organização. Neste caso, nem sempre a organização irá optar pela contratação ou capacitação de profissionais e irá, na verdade, procurar atender ao maior número possível de restrições. Assim, no problema da alocação, caso não haja uma solução que satisfaça a todas as restrições, uma abordagem é retornar a alocação que quebre o menor número possível de restrições. Neste trabalho, essa abordagem foi escolhida. No entanto, deu-se também a possibilidade de retornar a solução parcial que teve profissionais alocados ao maior número possível de atividades sem quebrar nenhuma restrição.

Na seção a seguir, será discutida a solução proposta para o problema da alocação, explicando os algoritmos e técnicas que foram utilizados nessa solução.

4.3 Solução do Problema de Alocação de Recursos Humanos em Projetos de Software

O problema da alocação de recursos humanos, conforme mencionado na seção 4.1, é um problema de alta complexidade, uma vez que são possíveis diversas soluções diferentes, com vários arranjos entre profissionais e atividades, além dos muitos fatores a considerar no problema.

Como exemplo, pode-se citar a situação em que existam dez atividades não paralelas a serem executadas e dez profissionais na organização que possam realizá-las. Neste caso, serão 10^{10} diferentes casamentos entre profissionais e atividades, ou melhor, dez bilhões de possibilidades. Em um exemplo mais simples, com duas atividades e dois profissionais, pode-se mostrar facilmente que o número de possibilidades é igual ao número de profissionais elevado ao número de atividades. Nesse caso, as possibilidades seriam: o primeiro profissional realizar a primeira atividade e o segundo profissional a segunda atividade, o segundo profissional realizar a primeira atividade e o primeiro profissional a segunda atividade, o primeiro profissional realizar as duas atividades e o segundo profissional realizar as duas atividades. Assim, pode-se dizer que o problema é exponencial: $O(p^a)$, onde p é o número de profissionais e a o número de atividades.

No entanto, apesar de o problema ser bastante complexo, é de se esperar que as restrições diminuam bastante seu número de possíveis soluções, tornando-o possível de ser resolvido em um tempo razoável: daí a importância das restrições no problema. Voltando ao exemplo das dez atividades e dez profissionais, é possível que as restrições determinem, por exemplo, que algumas atividades são paralelas (portanto, o mesmo profissional não pode realizá-las, o que diminui o número de possibilidades). Além disso, nem todos os profissionais vão ter qualificação necessária para executar todas as atividades (o que reduzirá ainda mais o número de possíveis combinações). Suponha que devido a essas restrições, tenha-se chegado a uma situação em que cada profissional possa, inicialmente, realizar apenas cinco atividades. Assim, o número de diferentes soluções cairia de dez bilhões para cem mil (10^5), ou seja, 0,001% da quantidade original. No entanto, cem mil seria o número máximo, uma vez que ao se atribuir profissionais às atividades, novas restrições são criadas (por exemplo, este profissional não pode mais realizar outras atividades no período, os outros profissionais não podem mais realizar a atividade, entre outros) e o número de soluções provavelmente cairá ainda mais.

A abordagem proposta para a solução do problema da alocação de recursos humanos, neste trabalho, contempla diferentes possibilidades, conforme dito na seção anterior. São elas:

- N soluções quaisquer: retornar uma quantidade N de soluções quaisquer que satisfaçam às restrições;
- Problema ótimo: retornar as soluções que satisfaçam às restrições e que representem as soluções de melhor custo dado um fator (menor custo, menor equipe, menor sub-alocação, melhor qualificação, menor tempo);
- Problema parcial ótimo quanto às restrições: retornar as soluções parciais que quebrem o menor número de restrições do problema;
- Problema parcial ótimo quanto às atividades: retornar as soluções parciais que atribuam profissionais ao maior número de atividades.

A solução proposta para cada uma dessas possibilidades será discutida nas próximas subseções. Alguns passos dos algoritmos utilizados merecem atenção especial e serão explicados a seguir.

O primeiro passo para se resolver o problema da alocação como um problema de satisfação de restrições é determinar o domínio inicial das variáveis. Isso quer dizer determinar quais profissionais (valores) são capazes de realizar cada atividade (variáveis). Pode-se considerar que esse passo faz a consistência de arco (RUSSELL e NORVIG, 2003), já mencionada no capítulo 3. Neste trabalho, essa determinação de domínio inicial é feita da seguinte forma:

```
Para cada profissional:
  Para cada atividade:
    checa se o profissional possui as características exigidas pela atividade;
    checa se o profissional não está indisponível no período da atividade;
    armazena se o profissional pode ou não realizar a atividade;
  Fim para.
Fim para.
```

Figura 4.1 – Algoritmo de determinação do domínio inicial

Outro passo importante é a seleção da próxima atividade para a qual será escolhido um profissional. Para isso, é utilizada a heurística *most-constrained-variable* (BITNER e REINGOLD, 1975), que implica em selecionar sempre como próxima atividade a ser considerada aquela que tiver o menor domínio, ou seja, a que puder ser realizada pelo menor número de profissionais.

Uma das partes mais importantes do problema é atribuir as alocações propriamente ditas, ou seja, determinar que um dado profissional será alocado a uma certa atividade. Uma vez o domínio inicial tendo sido determinado e a atividade inicial escolhida, pode-se começar as tentativas para determinar qual profissional pode realizar a atividade até que todas as atividades tenham algum profissional associado ou não seja possível retornar uma solução. Isso foi feito utilizando-se as duas funções descritas a seguir:

```
Função Boolean FixaAlocacao (Profissional P, Atividade A):

  Fixa alocação do profissional à atividade A;
  Atualiza rede de tarefas com base na produtividade do profissional;
  Incrementa funções de custo;
  Insere período de indisponibilidade relativo à atividade para o profissional;
  Atualiza domínios;

  Se (domínio de alguma variável foi esvaziado):
    retorna Falso;
  Senão
    retorna Verdadeiro;
  Fim Se.

Fim Função.
```

Figura 4.2 – Algoritmo FixaAlocação

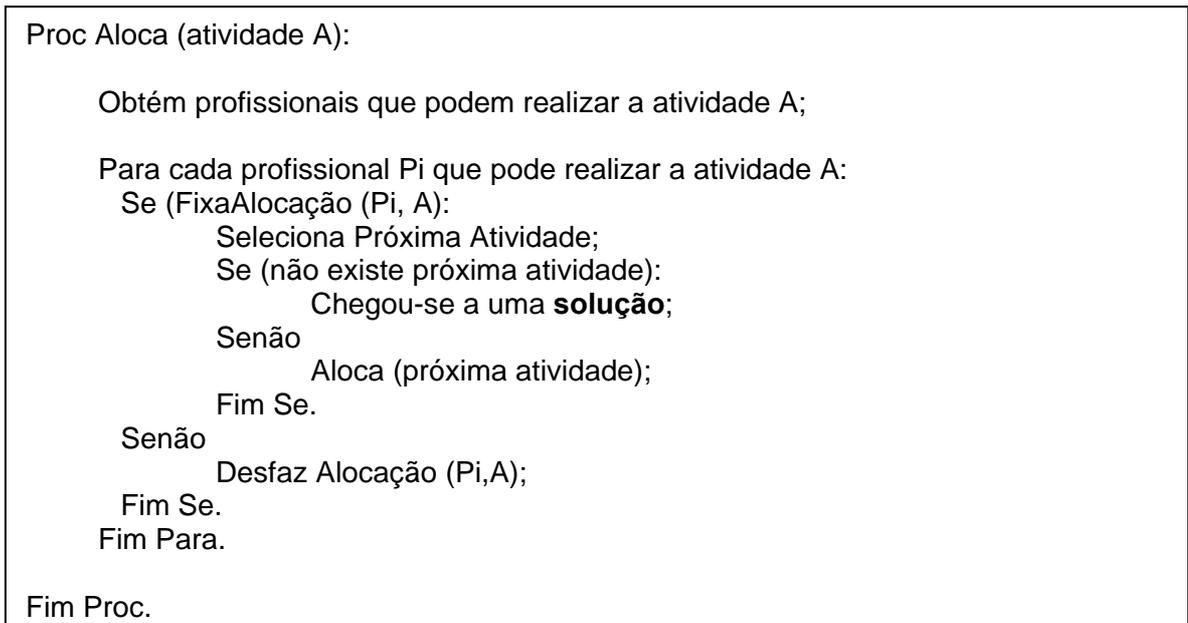


Figura 4.3 – Algoritmo Aloca

A função *FixaAlocacao*, primeiramente, determina que o profissional irá realizar a atividade (modifica o domínio para refletir isso). Depois disso, atualiza a rede de tarefas, ou seja, modifica as datas da atividade de acordo com a produtividade do profissional, se for o caso, e re-escalona as demais atividades. O próximo passo é atualizar as funções de custo consideradas no problema. Depois, os domínios são atualizados, já que os outros profissionais não podem mais realizar a tarefa e o profissional pode ter ficado impedido de realizar alguma outra tarefa no mesmo período. Por fim, é feito o teste relativo ao *forward checking*, que verifica se o domínio de alguma variável esvaziou, ou seja, se há alguma atividade para a qual não se possa atribuir nenhum profissional. Caso isso aconteça, o caminho escolhido não levará a uma solução e deve ser abandonado.

A função *Aloca* é recursiva, sendo responsável por chamar *FixaAlocacao* diversas vezes, para determinar se as alocações fixadas levam a uma solução. No caso de todas as alocações serem fixadas com sucesso, chegou-se a uma solução. Se em algum momento for possível determinar que o caminho escolhido não levará a uma solução, esta é abandonada e a alocação fixada é desfeita, para que outro caminho possa ser experimentado.

Assim, foi explicado o funcionamento básico do algoritmo de alocação proposto neste trabalho. Nas próximas subseções, aborda-se cada uma das possibilidades de alocação suportadas pelos algoritmos propostos neste trabalho.

4.3.1 N Soluções Quaisquer

Neste caso, o objetivo é apenas retornar um número N de soluções quaisquer para o problema, ou seja, quaisquer alocações, podendo inclusive serem todas em que sejam atribuídos profissionais a todas as atividades, satisfazendo todas as restrições.

O número de passos necessários e o tempo de execução deste caso estão diretamente ligados ao número de soluções escolhido. Caso o número de soluções escolhido seja baixo, o problema tende a ser bastante simples. No entanto, é possível que, caso o número de soluções escolhido seja muito alto, a solução para o problema se torne inviável. Voltando ao exemplo citado acima, com dez profissionais e dez atividades sem restrição alguma imposta, deveriam ser gerados dez bilhões de soluções diferentes, o que dá uma dimensão do custo da solução. Portanto, se o objetivo do emprego da abordagem proposta é facilitar o trabalho do gerente, não faz muito sentido retornar para ele bilhões de possibilidades para que escolha uma, e assim limitar o número de soluções pode ser interessante.

Utiliza-se o algoritmo *backtracking* (GOLOMB e BAUMERT, 1965) com *forward checking* (HARALICK e ELLIOTT, 1980), além da heurística para a seleção de variáveis *most constrained variable* (BITNER e REINGOLD, 1975).

O algoritmo básico, neste caso, apenas determina o domínio inicial (Figura 4.1), seleciona a próxima variável a ser considerada e a partir daí executa o algoritmo *Aloca* (Figura 4.3). A particularidade deste caso é que, uma vez que se chegue ao número desejado de soluções (a menos que se selecione gerar todas), o algoritmo retorna esse conjunto de soluções. Em geral, obter essas soluções tende a ser rápido, principalmente se o número de soluções escolhidas for pequeno. No entanto, é possível que o problema se torne bastante complexo, dependendo dos dados de entrada. Em uma situação com poucas restrições (muitos caminhos a serem testados) em que sempre sejam tomadas as decisões erradas (escolhendo-se valores que não levem à solução e também não gerem restrições), o problema pode ter um custo de solução muito alto. Contudo, ao utilizar-se *forward-checking* e *most-constrained-variable* espera-se que situações como as descritas não ocorram, como explicado no capítulo 3.

4.3.2 Problema Ótimo

Neste caso, o objetivo é retornar todas as soluções para o problema que tenham o melhor custo para uma dada função de custo.

Utiliza-se o algoritmo *backtracking* (GOLOMB e BAUMERT, 1965) com *forward checking* (HARALICK e ELLIOTT, 1980), além da heurística para seleção de variáveis *most constrained variable* (BITNER e REINGOLD, 1975). Para otimizar o fator utiliza-se *branch-and-bound* (LAWLER e WOOD, 1966), ou seja, a cada momento uma solução parcial não pode ter custo pior que a solução de melhor custo encontrada até o momento.

Na verdade, o que é feito é apenas impor mais uma restrição ao problema, ou seja, encontra-se uma solução qualquer e determina-se que ela tem o melhor custo até o momento e o processo continua buscando soluções. Se em algum ponto da busca por uma solução o custo acumulado dessa solução não puder mais ser melhor que o da solução de melhor custo, abandona-se essa tentativa de solução, uma vez que com certeza essa não seria a solução ótima. Caso se prossiga na busca por uma solução, até que se encontre uma cujo custo seja melhor que o da solução de melhor custo até o momento, atualiza-se o melhor custo e a solução encontrada passa a ser a de melhor custo. Quando não for mais possível encontrar uma equipe de melhor custo, a última encontrada realmente é a que tem o melhor custo para o problema. Ou seja, parte-se de uma solução e esta é melhorada ao longo da busca.

O algoritmo básico neste caso apenas determina o domínio inicial (Figura 4.1), seleciona a próxima variável a ser considerada e a partir daí executa o algoritmo *Aloca* (Figura 4.3). No entanto, o algoritmo *FixaAlocacao* (Figura 4.2) é alterado para que, a cada alocação que é fixada, o custo da solução seja atualizado e verifique-se se o custo acumulado da solução não é maior (para o caso em que se deseja o valor mínimo) que o custo da melhor solução. Se isso acontecer, *FixaAlocacao* retorna Falso e a busca naquela solução não continua. No algoritmo *Aloca* (Figura 4.3), sempre que se chega a uma solução, o custo da melhor solução é atualizado, se for o caso.

4.3.3 Problema parcial ótimo quanto às restrições

Neste caso, o objetivo é retornar todas as soluções parciais para o problema que representem aquelas que quebram o menor número possível de restrições do problema.

Este caso é um pouco diferente dos demais, exatamente pelo fato de restrições serem quebradas. Uma vez que os casos anteriores se baseavam fortemente nas restrições do problema, essa é uma diferença importante. Fica claro que o *forward-checking* (HARALICK e ELLIOTT, 1980) não pode mais ser empregado, uma vez que se baseia

no fato de domínios não poderem ser esvaziados, o que acontecerá no caso de não haver soluções.

Neste trabalho, o problema foi resolvido com *backtracking* (GOLOMB e BAUMERT, 1965) e *branch and bound* (LAWLER e WOOD, 1966), utilizando como função de custo o número de restrições quebradas e procurando minimizar essa função.

Para uma utilização mais eficiente do *branch and bound* é preciso partir de uma boa solução inicial, ou seja, uma solução que tenha custo próximo ao custo da melhor solução. Isso permite que o algoritmo volte de caminhos errados mais cedo e perca menos tempo em caminhos que não conduzem à melhor solução.

Assim, para tentar partir de uma boa solução, o algoritmo proposto neste trabalho primeiramente determina o domínio inicial (Figura 4.1). Os valores que são determinados neste passo representam custo zero, ainda que não seja possível atribuir valor para todas as variáveis (senão seria uma solução). A partir desse domínio inicial, atribui-se valores às variáveis que não receberam valor, procurando quebrar menos restrições. Com isso, chega-se a uma primeira solução e a um primeiro valor de custo, que se acredita ser um bom valor, uma vez que algumas alocações tiveram custo zero (mas nada garante que essa solução realmente seja próxima da melhor). A partir dessa solução inicial, inicia-se o algoritmo propriamente dito, que procura em todo o espaço de busca pela melhor solução, desconsiderando apenas aqueles caminhos que ultrapassarem o custo do melhor custo atual.

4.3.4 Problema parcial ótimo quanto às atividades

Neste caso, são retornadas todas as soluções parciais para o problema em que foram atribuídos profissionais ao maior número de atividades possível, considerando-se que para as demais atividades outras medidas serão tomadas (contratação, treinamento, etc.).

Este caso é similar ao anterior, uma vez que não se pode basear tanto nas restrições, uma vez que com certeza algumas serão quebradas. Da mesma forma que no caso anterior, o *forward-checking* (HARALICK e ELLIOTT, 1980) não pode ser utilizado. No entanto, uma variação do algoritmo pode ser empregada. Só será executada essa modalidade do algoritmo no caso de uma tentativa mal sucedida de obter soluções para o problema. Assim, podem-se aproveitar informações dessa tentativa de solução na solução do problema parcial. Isso pode ser feito armazenando-se o número máximo de atividades às quais se conseguiu atribuir profissionais. Pode-se afirmar que pelo menos o mesmo número de atividades a solução parcial terá, e soluções com menos atividades

não são as melhores. Dessa forma, pode-se utilizar uma variação do *forward-checking* em que, a cada momento, se o número de domínios vazios for maior que a diferença entre o número total de atividades e o melhor número de atividades até o momento, aquela tentativa de solução pode ser abandonada, pois não chegaria à solução parcial onde o maior número de atividades teve algum profissional atribuído, sem quebrar as restrições do problema.

Portanto, o que foi feito neste trabalho foi resolver o problema com *backtracking* (GOLOMB e BAUMERT, 1965) e *branch and bound* (LAWLER e WOOD, 1966), além da variação do *forward-checking* explicada acima, utilizando como função de custo o número de atividades para as quais foi atribuído algum profissional (sem quebrar as restrições) e procurando maximizar essa função.

O algoritmo proposto neste trabalho, como no caso anterior, primeiramente determina o domínio inicial (Figura 4.1). Então, tem início o algoritmo propriamente dito, que procura em todo o espaço de busca pela melhor solução, desconsiderando apenas aqueles caminhos que falhem no teste da variação do *forward-checking*.

4.4 Exemplo de Utilização da Abordagem de Alocação Baseada em Satisfação de Restrições

Para possibilitar um melhor entendimento da abordagem de alocação proposta neste trabalho, nesta seção, é apresentado um exemplo de sua utilização. É definida uma situação com um projeto, atividades e profissionais fictícios e são mostradas as possíveis soluções para o problema, dadas as restrições impostas.

Sejam, portanto, as seguintes características e seus domínios:

Tabela 4.1 – Características do exemplo e seus domínios

Características	Domínios	
	Valor Numérico	Descrição
Relacionamento com o Público Negociação Trabalho em Equipe	1	Tem treinamento
	2	Possui Habilidade
	3	Possui Grande Habilidade
Técnicas de Lev. de Requisitos Análise Orientada a Objetos Banco de Dados Projeto Orientado a Objetos Java Técnicas de Testes	1	Conhece e Sabe Executar com Supervisão
	2	Conhece e Executa sem Supervisão
	3	Possui Expertise e Mentoria
Experiência em Telecomunicações	1	Entre 2 e 6 meses
	2	Entre 6 meses e 1 ano
	3	Entre 1 e 3 anos
	4	Acima de 3 anos

A figura 4.4 apresenta a rede de atividades utilizada no exemplo. Cada atividade é apresentada com seu nome e a lista das características exigidas, juntamente com a intensidade da característica que o profissional deve possuir para realizar a atividade. As intensidades são representadas através de valores numéricos que variam de 1 a 4. Os profissionais são apresentados na figura 4.5. Para cada profissional são apresentados: o nome, o valor da hora de trabalho do profissional e a lista de características possuídas pelo profissional.

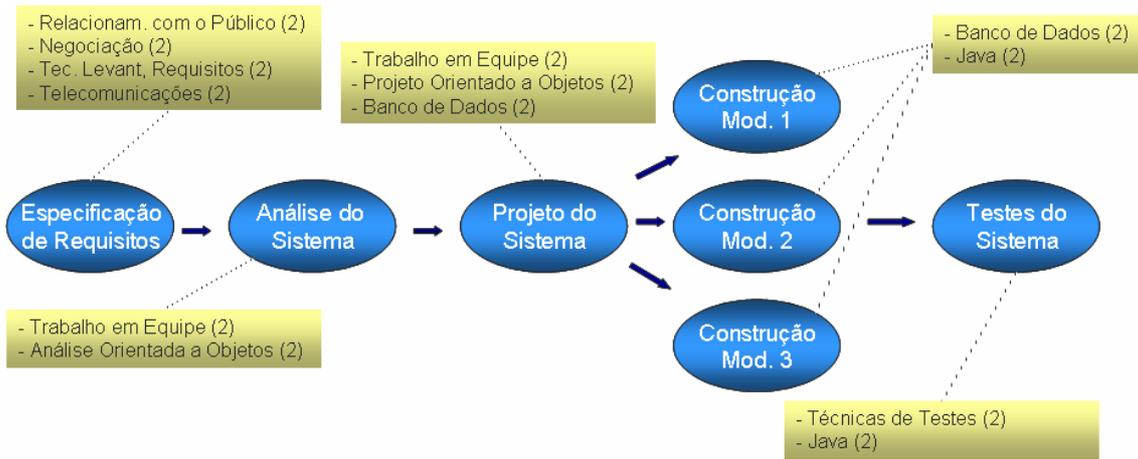


Figura 4.4 – Atividades do exemplo



Figura 4.5 – Profissionais do exemplo

A figura 4.6 apresenta o número de profissionais que podem realizar cada uma das atividades do exemplo, considerando-se apenas a qualificação necessária para realizar as atividades contra a qualificação possuída pelos profissionais.

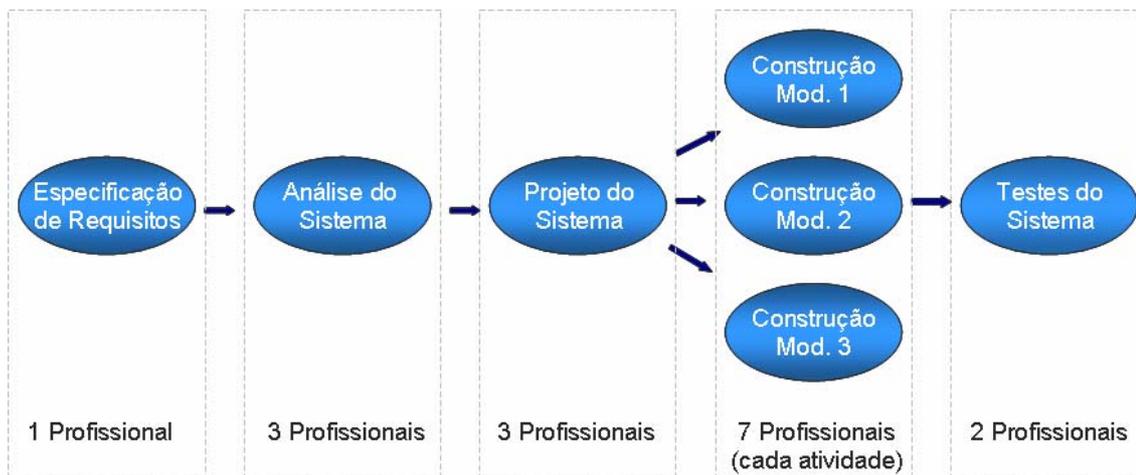


Figura 4.6 – Profissionais qualificados para realizar as atividades

Considerando-se as atividades e profissionais descritos nas figuras 4.4 e 4.5, é possível montar 2160 equipes que atendam às restrições. Se o simples fato de determinar uma equipe que atendesse aos requisitos impostos sem nenhum apoio já seria difícil, mais difícil ainda seria gerar essas 2160 equipes, calculando fatores como custo, tamanho e sub-alocação em cada uma delas. Vale salientar que, se fossem retiradas as características exigidas das atividades e não houvesse sobreposição de atividades, seriam possíveis mais de 800000 alocações diferentes.

Neste exemplo, as funções de custo que podem ser definidas para o problema variam da seguinte forma:

Tabela 4.2 – Variação dos fatores no exemplo

Fator	Valor Mínimo	Valor Máximo	Δ Percentual
Tempo	39 dias	46 dias	17,9%
Custo	\$ 10360	\$ 26360	154,4%
Ind. Sub-alocação	91	223	145,0%
Tamanho	4 pessoas	7 pessoas	42,9%

Com isso, percebe-se que há uma grande variação em todos os fatores considerados. Uma vez que o gerente dificilmente seria capaz de considerar todas as possibilidades e otimizar esses fatores sem apoio, fica claro o quanto se pode desviar de uma solução ótima.

Exemplificando, caso o objetivo do gerente seja determinar uma equipe que atenda às restrições e procure diminuir os gastos do projeto, é possível que ele, sem apoio, se distancie da equipe que representa o menor custo em até 16.000,00 e a solução de maior custo representa quase três vezes o valor da solução de menor custo.

Assim, é possível ilustrar o quanto é difícil considerar todas as opções e satisfazer a todas as restrições de forma *ad-hoc*, e o quanto se pode distanciar de um valor ótimo para algum fator envolvido no problema, mesmo em um exemplo que representa um projeto pequeno, com poucas pessoas e atividades, além da curta duração. A figura 4.7 ilustra os conjuntos das possíveis soluções para o problema. Sem nenhum apoio, o gerente parte do conjunto maior de soluções, no qual o número de soluções a considerar para que seja escolhida uma é muito grande. Ao considerar as equipes que satisfazem as restrições do problema, o número de equipes a considerar tende a cair drasticamente. Além disso, é de se esperar que ao escolher uma equipe desse conjunto, todos os profissionais sejam qualificados para realizar as atividades às quais foram alocados. Por fim, ao considerar apenas as soluções ótimas quanto a algum fator, o número de equipes a considerar tende a cair ainda mais. Além disso, as equipes não apenas são qualificadas, mas também minimizam ou maximizam o fator selecionado para o projeto.

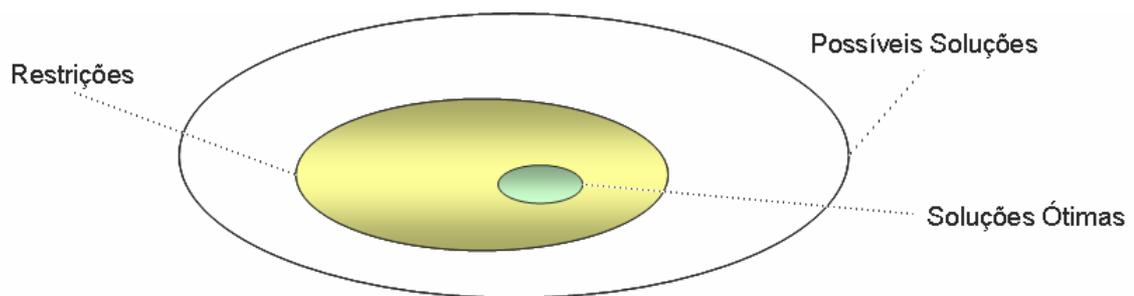


Figura 4.7 – Conjuntos de soluções para o problema

4.5 Conclusão

Alocar pessoas em projetos de software muitas vezes pode ser uma tarefa bastante complexa, devido ao número de fatores a considerar e o grande número de diferentes combinações possíveis de equipes.

Portanto, neste trabalho, defende-se a idéia de que, devido à complexidade da montagem de equipe em projetos de software, é necessário apoiar o gerente de projetos para que este possa realizar a tarefa de forma mais rápida e precisa.

A abordagem proposta para fornecer esse apoio foi descrita neste capítulo. Foram apresentados: a definição formal do problema que se pretende resolver, os algoritmos que podem ser utilizados e as diferentes opções de solução e como cada uma delas pode ser implementada. Além disso, foi mostrado um exemplo de uma situação em que a abordagem poderia ser utilizada e a que resultados se chegaria com sua aplicação.

No capítulo a seguir, será descrita a ferramenta implementada para fornecer apoio gerencial através da utilização da abordagem proposta, e o estudo experimental realizado para avaliar essa abordagem.

CAPÍTULO 5

Um Estudo Experimental para Investigar a Relevância do Uso da Abordagem Proposta

5.1 Introdução

Para avaliar a relevância do apoio à decisão na atividade de alocação de recursos humanos em projetos de software, foi planejado e executado um estudo experimental. Este estudo teve como objetivo investigar se os executantes da atividade de alocação de recursos humanos nas tarefas componentes de um projeto de software encontram dificuldades para escolher as equipes e priorizar os fatores relevantes para o projeto sem a utilização do apoio proposto. Caso isso aconteça, considera-se que haverá indícios de que apoiar a execução da atividade de alocação de recursos humanos, utilizando, por exemplo, uma abordagem como a desenvolvida no capítulo 4 pode trazer benefícios ao gerente de projeto.

Com o objetivo de demonstrar o uso da abordagem proposta, foi desenvolvido um protótipo de uma ferramenta que implementa as funções e algoritmos descritos no capítulo 4 e representa a materialização do apoio à decisão gerencial proposto. Dessa forma, o gerente de projetos não precisa conhecer ou se preocupar com a modelagem e com as técnicas de resolução do problema, uma vez que essa complexidade é encapsulada na ferramenta. A ferramenta foi necessária, também, para a execução do estudo experimental realizado.

Na seção a seguir (5.2), apresentamos a ferramenta desenvolvida para fornecer o apoio à decisão proposto neste trabalho. Na seção 5.3, apresentamos o planejamento, a execução e a análise dos resultados do estudo experimental. Por fim, na seção 5.4 apresentamos as considerações finais deste capítulo.

5.2 Uma Ferramenta de Apoio à Alocação de Recursos Humanos em Projetos de Software

A modelagem do problema de alocação de recursos humanos em atividades de um projeto de desenvolvimento de software, a proposta e formalização de um conjunto de funções de custo e algoritmos que podem ser utilizados para sua solução pode não ser suficiente para auxiliar um gerente de projetos na execução da atividade de alocação de recursos humanos. Para fornecer esse auxílio, é necessário esconder do gerente a

complexidade do problema e disponibilizar para ele um meio automatizado de obter o apoio para a solução do problema de forma simples e rápida.

Para fornecer essa automatização, foi desenvolvido um protótipo de uma ferramenta de apoio à abordagem apresentada no último capítulo. A ferramenta sugere ao gerente possíveis equipes para as atividades de seu projeto, considerando-se um conjunto de restrições envolvido no problema, conforme descrito no capítulo 4. Essa ferramenta implementa a modelagem, os algoritmos e funções descritos naquele capítulo.

Portanto, a ferramenta permite que o gerente introduzir as informações que descrevem as atividades do projeto, os profissionais e as restrições envolvidas (características, datas, indisponibilidades) e escolha uma das opções implementadas: (i) uma ou todas as soluções (no entanto, a escolha de todas as soluções é desaconselhada devido à possibilidade de retornar bilhões de soluções); ou (ii) otimizar ou não algum fator. Caso se deseje otimizar, selecionar o fator.

De acordo com as opções escolhidas, diferentes algoritmos são utilizados para gerar as equipes que atendam a essas opções. Vale salientar que a implementação garante que todas as possíveis combinações sejam consideradas, o que seria talvez impossível sem o apoio computacional (principalmente, quando muitos profissionais e atividades estão envolvidos no problema).

O primeiro passo para utilizar a ferramenta desenvolvida corresponde a fornecer as informações de entrada. Para isso, foram desenvolvidos cadastros para todas as informações necessárias. A ferramenta dispõe dos seguintes cadastros:

- Cadastro de Escalas: Neste cadastro é possível incluir, alterar, excluir e visualizar escalas e seus valores. Uma escala é composta por um nome e uma lista de valores. Um valor de escala é composto por uma descrição e um valor numérico;
- Cadastro de Tipos de Característica: Neste cadastro é possível incluir, alterar, excluir e visualizar tipos de característica. Exemplos de tipos de característica são: conhecimento, habilidade, experiência, entre outros. Um tipo de característica é composto por um nome e uma escala. Os tipos de características são usados apenas para classificar as características;
- Cadastro de Características: Neste cadastro é possível incluir, alterar, excluir e visualizar características. Uma característica pode ser exigida por uma atividade ou possuída por um profissional. É composta por um nome e um tipo de

característica. Exemplos de características são: conhecimento em banco de dados, habilidade de liderança, entre outros;

- Cadastro de Profissionais: Neste cadastro é possível incluir, alterar, excluir e visualizar profissionais. É possível informar o nome, o valor da hora de trabalho do profissional e o número de horas trabalhadas por dia, em média, pelo profissional. Além disso, devem-se informar as características possuídas pelo profissional e a intensidade em que as possui. Caso seja necessário, podem-se informar, ainda, os períodos de indisponibilidade do profissional;
- Cadastro de Atividades: Neste cadastro é possível incluir, alterar, excluir e visualizar as atividades componentes de um projeto de software. É possível informar o nome, as datas previstas para o início e encerramento da atividade e o número de horas que deve ser dedicado à atividade por dia. A janela do cadastro de atividades é mostrada na figura 5.1.

Nome	Data Início	Data Fim	Horas/Dia
Especificação de Requisitos	22/03/2005	29/03/2005	8
Análise do Sistema	30/03/2005	07/04/2005	8
Projeto do Sistema	08/04/2005	14/04/2005	8
Construção do Módulo 1	15/04/2005	28/04/2005	8
Construção do Módulo 2	15/04/2005	30/04/2005	8
Construção do Módulo 3	15/04/2005	29/04/2005	8

Figura 5.1 – Tela de Cadastro de Atividades

A ferramenta pode, também, importar as informações de entrada ou salvar as informações cadastradas utilizando arquivos XML. Assim, é possível obter as informações necessárias de alguma outra fonte existente e organizar essas informações no formato XML que a ferramenta espera, eliminando-se a necessidade do cadastro manual das informações. Também é possível salvar os dados registrados no sistema em

um arquivo XML, que poderá ser carregado em outras execuções da ferramenta, uma vez que muitas das informações mudam com pouca frequência (escalas, características e outras).

O próximo passo é determinar, caso necessário, qual modelo de produtividade deve ser utilizado na geração das equipes. São dadas quatro opções de modelo, conforme descrito no capítulo 4. São elas: não utilizar modelo algum; utilizar um modelo fixo de produtividade, baseado na experiência dos profissionais; utilizar um modelo configurável, em que se atribuem valores de produtividade para cada profissional, independentemente das atividades; e, por fim, um modelo configurável onde se atribuem valores de produtividade para cada profissional em cada atividade. A tela de configuração de produtividade é ilustrada na figura 5.2.

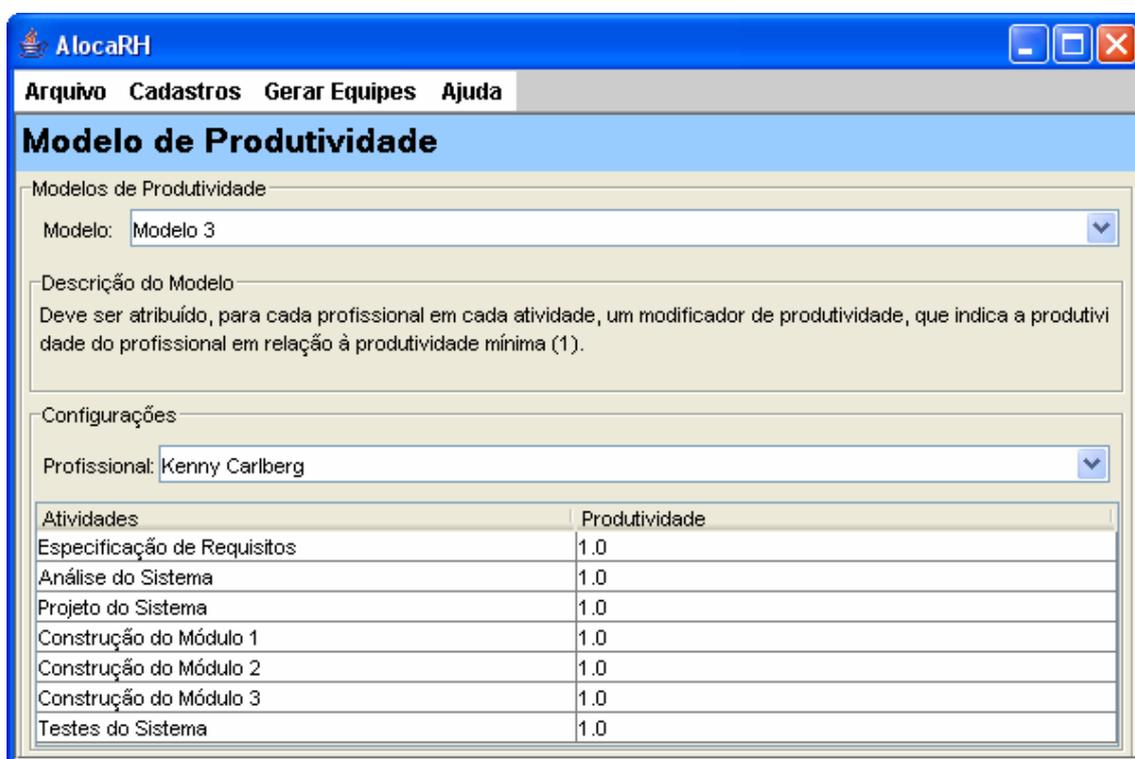


Figura 5.2 Tela de Seleção e Configuração de Modelos de Produtividade

Uma vez tendo sido fornecidas as informações de entrada necessárias (seja por cadastro ou por arquivo) e tendo sido escolhido que o modelo de produtividade será usado, é possível acessar a principal tela da ferramenta, ou seja, a tela de geração de equipes. O aspecto inicial da tela de geração de equipes é ilustrado na figura 5.3. Nesta tela, deve-se selecionar como as equipes devem ser geradas e solicitar a geração das equipes. Os algoritmos de seleção serão executados e o sistema exibirá uma lista das equipes, detalhando o valor de cada um dos fatores envolvidos. Ao se selecionar uma

das equipes, serão exibidos os detalhes da equipe (qual profissional deve realizar qual atividade, o período da atividade e o número de horas que deve ser dedicado à atividade por dia).

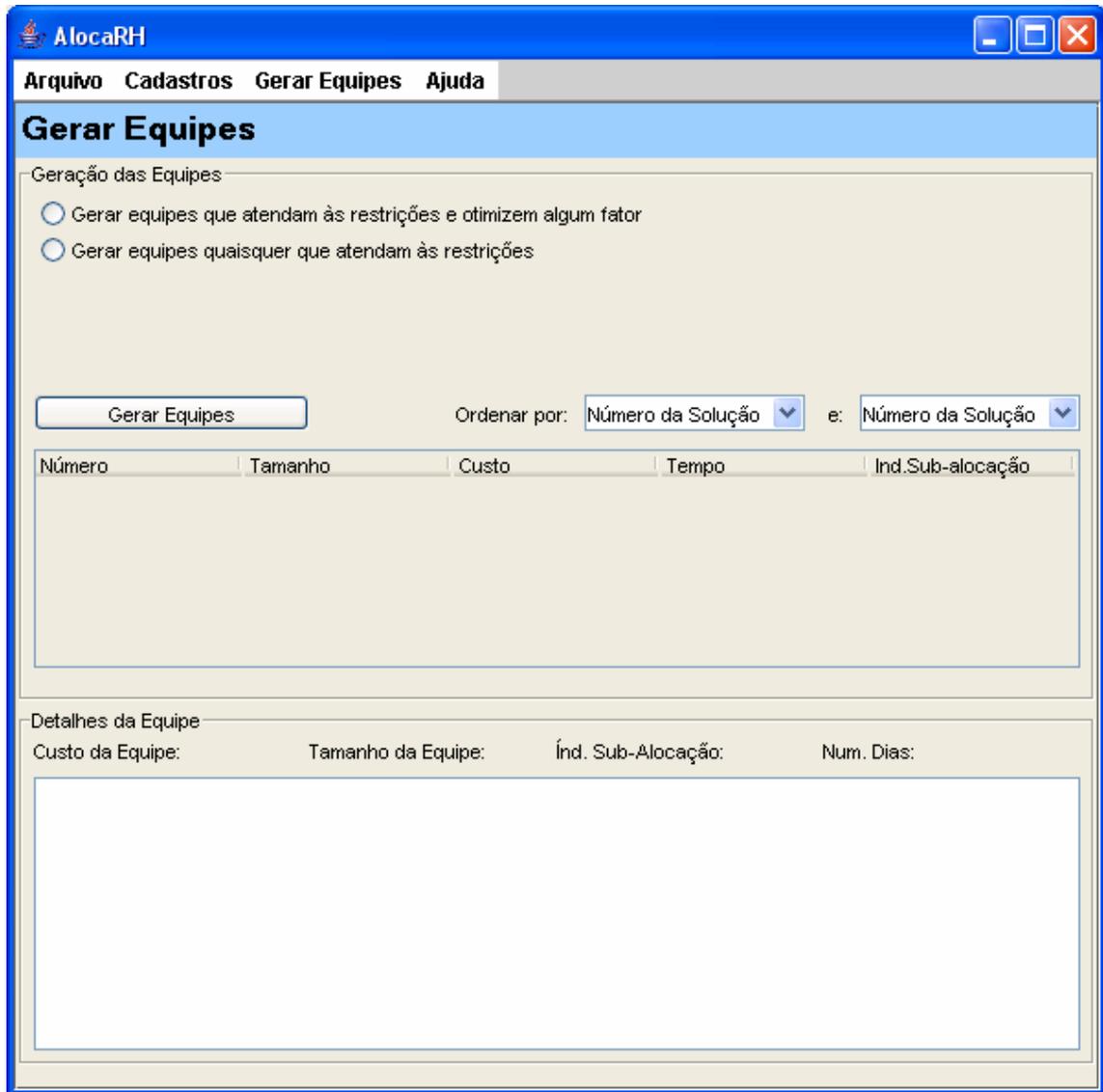


Figura 5.3 – Tela de Geração de Equipes (Aspecto Inicial)

Na tela de geração de equipes, inicialmente o usuário seleciona uma das duas opções: (i) gerar equipes que atendam às restrições e otimizar algum fator; ou (ii) gerar equipes quaisquer que atendam às restrições.

Ao selecionar a opção “Gerar equipes que atendam às restrições e otimizem algum fator” será exibida uma lista com os fatores que podem ser otimizados (menor custo, menor tempo, menor equipe, menor sub-alocação e maior qualificação). Deve-se, então, selecionar um dos fatores a ser otimizado e solicitar a geração das equipes (botão “Gerar Equipes”). Serão geradas as equipes que satisfaçam às restrições e otimizem o fator

escolhido, exibindo-se, para cada uma delas, o valor de cada um dos fatores envolvidos. Ao se selecionar uma das equipes, suas informações serão detalhadas. Caso se deseje, é possível ordenar a lista de soluções por algum fator. A figura 5.4 ilustra a tela de geração de equipes. Na situação mostrada, foram geradas as equipes que atendem às restrições do problema e otimizam o fator custo (ordenadas por tempo).

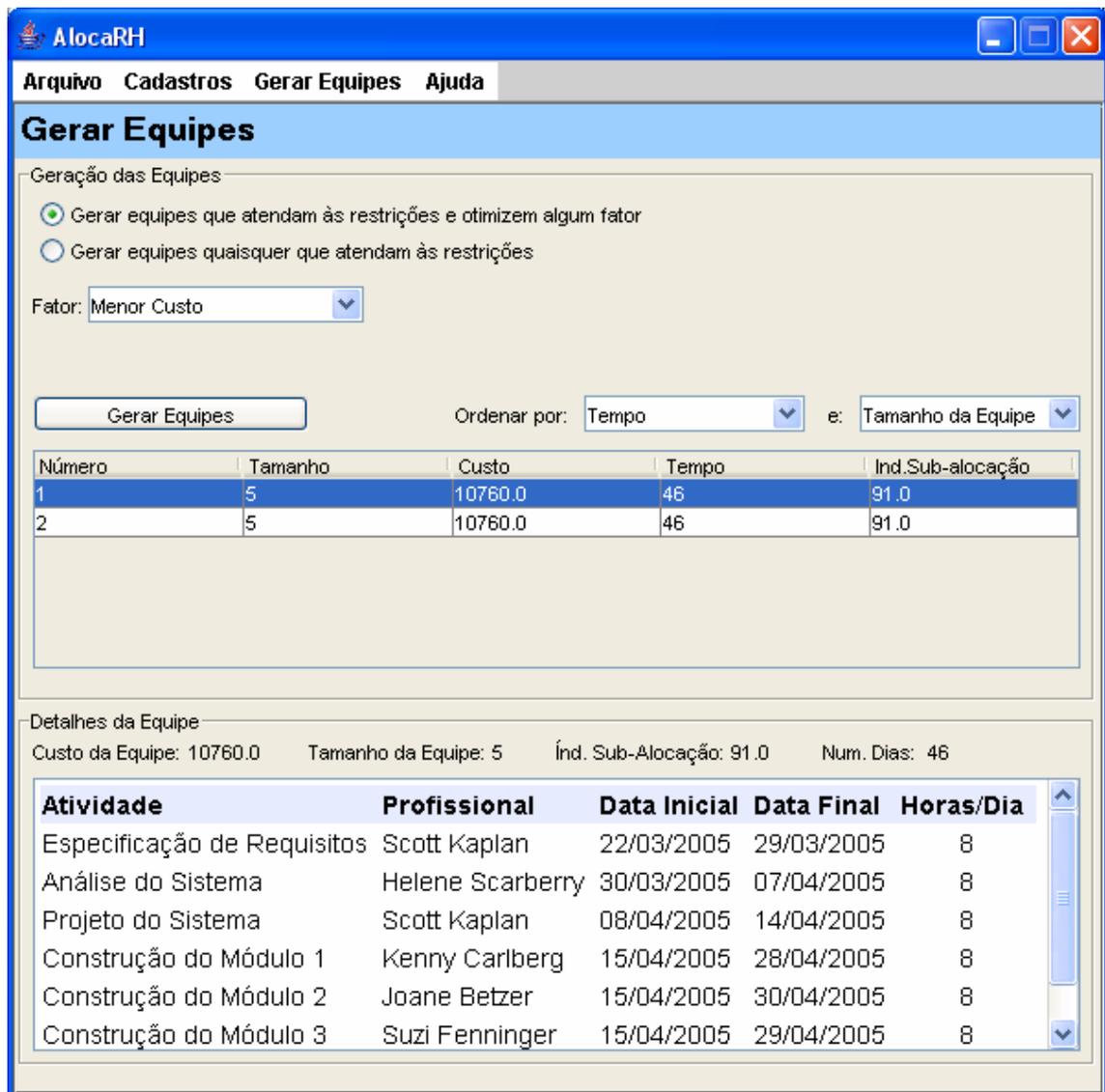


Figura 5.4 – Tela de Geração de Equipes, otimizando o fator custo

Ao selecionar a opção “Gerar equipes quaisquer que atendam às restrições”, será dada a opção de se gerar todas as possíveis equipes, ou de limitar o número de soluções a serem geradas (uma vez que, dependendo das particularidades do problema, o número de soluções pode ser muito grande). Deve-se escolher o número de soluções e solicitar a geração das equipes (botão “Gerar Equipes”). Serão listadas todas ou o número escolhido de equipes que satisfaçam às restrições do problema, exibindo-se, para cada

uma delas, o valor de cada um dos fatores envolvidos. Ao se selecionar uma das equipes, suas informações serão detalhadas. Como no caso anterior, é possível ordenar a lista de soluções por algum fator.

5.3 Estudo Experimental

Experimentação é o centro do processo científico. Ela oferece um modo sistemático, disciplinado, computável e controlado para a avaliação de novos métodos, técnicas, linguagens e ferramentas. Estes não deveriam ser sugeridos, publicados ou apresentados para venda sem experimentação e avaliação (AMARAL, 2003).

Assim, com o objetivo de tentar obter maiores indícios sobre a relevância do uso da abordagem proposta neste trabalho, foi planejado e executado um estudo experimental. Nas próximas seções, descrevemos o plano e a operação do estudo, assim como os resultados de sua análise.

5.3.1 Definição dos Objetivos

5.3.1.1 – Objetivo Global

Caracterizar a relevância da utilização da abordagem de alocação de recursos humanos em projetos de desenvolvimento de software baseada em satisfação de restrições. Para isso, será feita uma comparação entre os resultados da execução da atividade realizada de forma *ad-hoc* e utilizando a automatização proposta neste trabalho, apresentada na seção 5.2.

5.3.1.2 – Objetivo de Medição

Tendo como base os resultados gerados pela execução da atividade de alocação de recursos humanos em um projeto de software através da ferramenta proposta neste trabalho, este estudo tem como objetivo caracterizar se alguma abordagem *ad-hoc* para a execução da atividade com as mesmas características é capaz de chegar a resultados iguais ou melhores que os resultados gerados pela ferramenta, no que diz respeito à qualidade da equipe escolhida (considerando o custo monetário e o tempo do projeto).

5.3.1.3 – Objetivo do Estudo

Analisar a execução da atividade de alocação de recursos humanos em projetos de software

Com o propósito de *caracterizar a relevância da utilização do apoio proposto, ou seja, da abordagem de alocação de recursos humanos em projetos de software baseada em satisfação de restrições*

Referente aos *ganhos obtidos por sua utilização*

Do ponto de vista do *pesquisador*

No contexto de *estudantes de mestrado e doutorado em Engenharia de Software.*

5.3.1.4 – Questões

Q1: A aplicação de algum procedimento *ad-hoc* para a execução da atividade de alocação de recursos humanos em projetos de software encontra uma equipe com melhor custo monetário e tempo esperado para a conclusão do projeto do que a equipe escolhida pelo pesquisador utilizando o apoio proposto?

Métrica1: Custo monetário para o projeto sendo utilizado no estudo, baseado no custo da hora de trabalho dos profissionais.

Métrica2: Tempo necessário para a conclusão do projeto sendo utilizado no estudo, considerando a produtividade dos profissionais escolhidos para as atividades.

5.3.2 Planejamento do Estudo

5.3.2.1 – Seleção do Contexto

Os participantes do estudo serão estudantes de mestrado e doutorado em Engenharia de Software. Eles serão requisitados a realizar a atividade de seleção de equipe para um projeto de software definido em laboratório (estudo *In-vitro*). O mesmo projeto será apresentado a todos os participantes e o objetivo de cada participante será determinar uma equipe que atenda às restrições impostas pelo projeto (qualificação dos profissionais para realizar as atividades, datas em que as atividades precisam ser realizadas), seja capaz de concluir o projeto proposto em um determinado número de dias e com o menor custo monetário possível (baseando-se no custo da hora de trabalho dos profissionais). As equipes montadas pelos participantes serão comparadas com a equipe gerada pela ferramenta de apoio à abordagem proposta. O processo utilizado é *off-line*, porque as heurísticas não são aplicadas durante um ciclo de desenvolvimento real do projeto de um software.

5.3.2.2 – Definição das Hipóteses

5.3.2.2.1 – Em relação à qualidade das equipes geradas

Hipótese nula (H0): o custo monetário e o tempo de projeto quando executado pelas equipes geradas através da utilização de abordagens *ad-hoc* é igual ou menor que o custo monetário e o tempo para o projeto quando executado pela equipe escolhida pelo pesquisador através do uso da abordagem proposta?

C(P)_{ad-hoc}: custo monetário – procedimentos *ad-hoc* – participantes; e

C(G)_{automatizado}: custo monetário – ferramenta de apoio – pesquisador.

T(P)_{ad-hoc}: tempo de projeto – procedimentos *ad-hoc* – participantes; e

T(G)_{automatizado}: tempo de projeto – ferramenta de apoio – pesquisador.

H0: $C(P)_{ad-hoc} \leq C(G)_{automatizado}$ ou $T(P)_{ad-hoc} \leq T(G)_{automatizado}$

Hipótese alternativa (H1): a utilização do apoio proposto melhora o custo e o tempo do projeto relativo à equipe gerada se comparado ao custo e ao tempo do projeto da equipe gerada para o projeto por outros procedimentos *ad-hoc* quando aplicados no mesmo projeto.

H1: $C(P)_{ad-hoc} > C(G)_{automatizado}$ e $T(P)_{ad-hoc} > T(G)_{automatizado}$

5.3.2.3 – Seleção de Indivíduos

Os participantes do estudo serão estudantes de mestrado e doutorado do Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ. A escolha dos indivíduos será baseada em princípios não probabilísticos, sendo a população determinada por conveniência.

5.3.2.4 – Variáveis

A variável independente é a forma como a atividade será realizada. Os possíveis valores para essa variável são:

- 1- A ferramenta que apóia a abordagem proposta, que será utilizada pelo pesquisador.
- 2- Os procedimentos *ad-hoc* não automatizados, utilizados pelos participantes.

As variáveis dependentes são:

- 1- Custo monetário relativo à equipe selecionada pela execução da atividade.
- 2- Tempo do projeto relativo à equipe selecionada pela execução da atividade.

5.3.2.5 – Validade do Estudo

5.3.2.5.1 - Validade Interna

A validade interna se refere a garantir que é o tratamento que produz os resultados do estudo, e não algum outro fator desconhecido ou sobre o qual não se tenha controle. A validade interna do estudo é dependente do número de participantes executando o estudo. Prevemos a participação de cerca de quinze participantes, o que garante um bom nível de validação interna do estudo. Certamente, um número maior de participantes melhoraria a validade interna. Outro ponto que pode influenciar o resultado do estudo é a troca de informações entre os participantes que já realizaram o estudo e os que ainda não o realizaram. Para evitar este problema, requisitaremos explicitamente que os participantes não troquem informações a respeito do projeto. Tentaremos ainda realizar o estudo em poucas sessões, onde os participantes trabalharão paralelamente, sem trocar informações. Finalmente, o projeto não será apresentado como uma competição, inibindo a comparação de resultados entre os participantes.

5.3.2.5.2 – Validade Externa

A validade externa mede a capacidade de o estudo refletir o mesmo comportamento em outros grupos de participantes e profissionais da indústria, ou seja, em outros grupos além daquele em que o estudo foi aplicado. O maior problema em relação à validade externa do estudo seria a falta de interesse dos participantes no estudo. Alguns indivíduos podem realizar o estudo de forma descomprometida, sem um interesse real na obtenção de boas equipes, ao contrário do que aconteceria na indústria durante um projeto real. As técnicas foram utilizadas fora do ambiente de desenvolvimento de software, portanto, não é possível generalizar os resultados obtidos na indústria de forma mais ampla.

5.3.2.5.3 – Validade de Construção

A validade de construção considera os relacionamentos entre a teoria e a observação, ou seja, se o tratamento reflete bem a causa e o resultado reflete bem o

efeito (WOHLIN, 2000). Neste estudo, é empregado o mesmo projeto tanto na realização automatizada da atividade pelo pesquisador quanto nos procedimentos *ad-hoc* pelos participantes, garantindo que o tratamento reflète a causa.

5.3.2.5.4 – Validade de Conclusão

Foi utilizado o mesmo projeto pelo pesquisador e pelos participantes, sendo possível acreditar que os custos monetários e os tempos encontrados para o projeto refletem o resultado dos procedimentos e/ou técnicas empregados individualmente.

5.3.3 Definição dos Objetivos

5.3.3.1 – Preparação

Nesta etapa os participantes preencheram o Formulário de Caracterização (formação acadêmica e experiência) apresentado no anexo I. Além disso, foi entregue aos participantes a descrição do problema (apresentado no anexo II), a lista das atividades e dos profissionais envolvidos no projeto que foi utilizado como base do estudo (também apresentados no anexo II). Foi feita uma breve exposição oral do que precisava ser feito e o instrumento do estudo foi apresentado aos participantes (figura 5.5).

Para servir de instrumento do estudo, foi desenvolvido um software para a realização da atividade de alocação. Este software permite ao participante escolher as equipes (quantas vezes desejar) e informa se esta equipe satisfaz ou não às restrições (de características e datas) do problema. Percebendo a seleção de uma equipe válida, o software exibe o custo e o tempo necessário para a equipe concluir o projeto proposto. Os dados da execução do estudo são salvos em arquivos XML com o histórico das operações (equipes escolhidas, tempo gasto, número de tentativas e outros). Este software foi desenvolvido apenas para ser o instrumento do estudo, mas poderia também ser útil a um gerente de projetos buscando determinar se uma dada equipe que ele julgue ser uma boa equipe satisfaz a todas as restrições envolvidas no problema, uma vez que mesmo essa verificação seria difícil de ser realizada sem algum apoio.

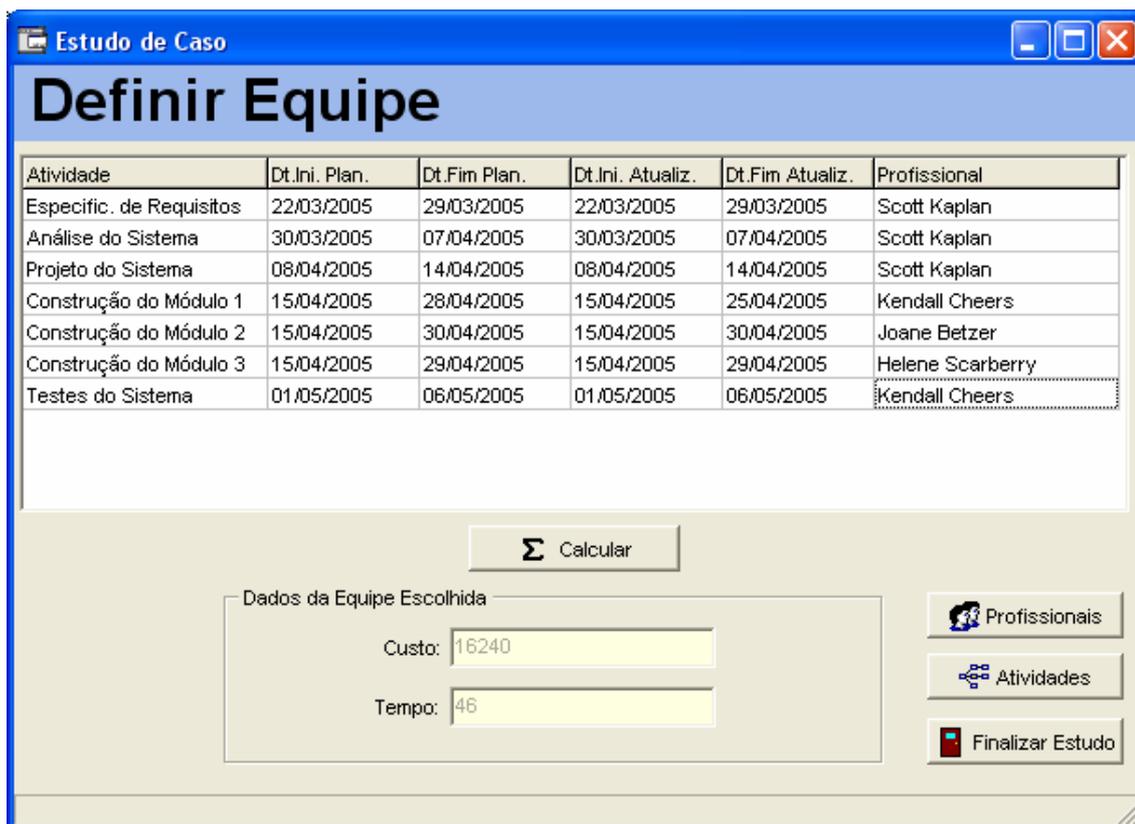


Figura 5.5 – Instrumento do Estudo

5.3.3.2 – Pesquisador: Utilizando o Apoio Proposto

Para obter os resultados através do apoio proposto, a ferramenta apresentada na seção 5.2 foi executada tendo como dados de entrada (apresentados no anexo II) os mesmos dados do projeto apresentado aos participantes, que realizaram o estudo sem utilizar esta ferramenta.

Na ferramenta, o pesquisador solicitou a geração das equipes escolhendo a opção de geração de todas as soluções, utilizando como opções de ordenação o tempo e o custo. A geração das equipes levou poucos segundos. Depois da geração, bastou percorrer a lista e selecionar a primeira equipe com o tempo determinado pelo problema. A tabela 5.1 apresenta os resultados da execução da ferramenta pelo pesquisador. O número de tentativas é o número de equipes avaliadas pelo usuário antes de chegar à equipe final (ou seja, são as equipes para as quais se calculou o custo e o tempo e foram melhoradas posteriormente). O tempo gasto é o tempo necessário para se chegar à melhor equipe encontrada. O custo e o tempo do projeto são características da equipe encontrada. A última coluna representa a diferença entre o custo da equipe escolhida e o custo da melhor equipe possível.

Tabela 5.1 – Resultado obtido pelo pesquisador

1º Tratamento	Nº de Tentativas	Tempo Gasto	Custo Equipe	Tempo Projeto	Δ Custo
Pesquisador	1	N/A	14040	42	0

5.3.3.3 – Participantes: Procedimentos *Ad-Hoc*

O outro tratamento foi apresentado aos 16 participantes do estudo. A execução do estudo foi realizada no laboratório da COPPE/UFRJ em dois dias. Em cada dia foram realizadas entre duas a três sessões, com diferentes participantes. Os resultados dos participantes são apresentados na tabela 5.2.

Tabela 5.2 – Resultados obtidos pelos participantes

Participantes	Nº de Tentativas	Tempo Gasto	Custo Equipe	Tempo Projeto	Δ Custo
Participante 1	6	24	14080	42	40
Participante 2	11	12	19760	42	5720
Participante 3	24	20	14040	42	0
Participante 4	47	28	16800	42	2760
Participante 5	16	30	16760	42	2720
Participante 6	31	38	14040	42	0
Participante 7	5	13	15480	42	1440
Participante 8	7	13	15480	42	1440
Participante 9	8	14	14040	42	0
Participante 10	2	13	21080	42	7040
Participante 11	27	36	14040	42	0
Participante 12	4	23	19640	42	5600
Participante 13	24	27	14080	42	40
Participante 14	4	20	19640	42	5600
Participante 15	8	26	15240	42	1200
Participante 16	1	6	10400	45	3640
Médias	14,1	21,4	15912,5	42,2	2327,5

5.3.4 Análise dos Resultados

Os resultados deste estudo foram avaliados através de uma análise qualitativa, dado que o objetivo do estudo não é provar de forma irrefutável suas hipóteses, mas obter indícios de que o uso da abordagem sugerida neste trabalho pode ser útil ao gerente de projetos que precise alocar recursos humanos. Assim, não consideramos necessário o uso de testes estatísticos ou técnicas quantitativas de análise (que exigiriam, inclusive, uma maior amostra de participantes do que o disponível) para se chegar às conclusões abaixo relacionadas.

O primeiro aspecto considerado na análise dos resultados foi o processo de solução utilizado pelos participantes do estudo, ou seja, quais soluções foram encontradas, em que seqüência e em quanto tempo. Analisando essa questão, foi possível perceber que

os participantes tiveram certa dificuldade em considerar as duas metas envolvidas no problema (tempo e custo monetário). Aparentemente, todos se preocuparam em encontrar uma equipe capaz de realizar a tarefa no tempo determinado e, uma vez encontrando esta equipe, muitos reduziram seus esforços para minimizar o custo. Cerca de 25% dos participantes pararam na primeira equipe encontrada com o tempo igual ao tempo solicitado. Para outros 25%, o número de tentativas realizadas após encontrar uma equipe com o tempo determinado foi menor que o número de tentativas realizadas antes de se encontrar essa equipe. Além disso, cerca de 8% pararam na primeira solução encontrada (nem no tempo determinado, nem no melhor custo). Essa análise indica uma dificuldade em se encontrar o ponto ótimo no processo de busca, além de uma tendência natural das pessoas em parar quando uma solução parece ser “boa o suficiente”. No entanto, não se pode negar que a parada pode estar relacionada à falta de benefícios diretos aos participantes, caso estes chegassem a uma melhor equipe.

Um outro aspecto considerado foi a experiência dos participantes. Foram feitos agrupamentos de duas formas diferentes, considerando a formação acadêmica e a experiência no desenvolvimento de software. Na tabela 5.3, são apresentados os resultados dos participantes doutorandos e dos mestrandos. Os participantes também foram agrupados por sua experiência na atividade de alocação de recursos humanos. Na tabela 5.4, são apresentados os resultados dos participantes agrupados pela experiência na alocação de recursos humanos.

Tabela 5.3 – Resultados obtidos pelos participantes agrupados por formação

Doutorandos	Nº de Tentativas	Tempo Gasto	Custo Equipe	Tempo Projeto	Δ Custo
Participante 1	6	24	14080	42	40
Participante 2	11	12	19760	42	5720
Participante 3	24	20	14040	42	0
Participante 4	47	28	16800	42	2760
Participante 5	16	30	16760	42	2720
Participante 6	31	38	14040	42	0
Participante 7	5	13	15480	42	1440
Participante 8	7	13	15480	42	1440
Participante 9	8	14	14040	42	0
Participante 16	1	6	10400	45	3640
Médias	15,6	19,8	15088	42,3	1776
Mestrandos	Nº de Tentativas	Tempo Gasto	Custo Equipe	Tempo Projeto	Δ Custo
Participante 10	2	13	21080	42	7040
Participante 11	27	36	14040	42	0
Participante 12	4	23	19640	42	5600
Participante 13	24	27	14080	42	40

Participante 14	4	20	19640	42	5600
Participante 15	8	26	15240	42	1200
Médias	11,5	24,2	17286,7	42	3246,7

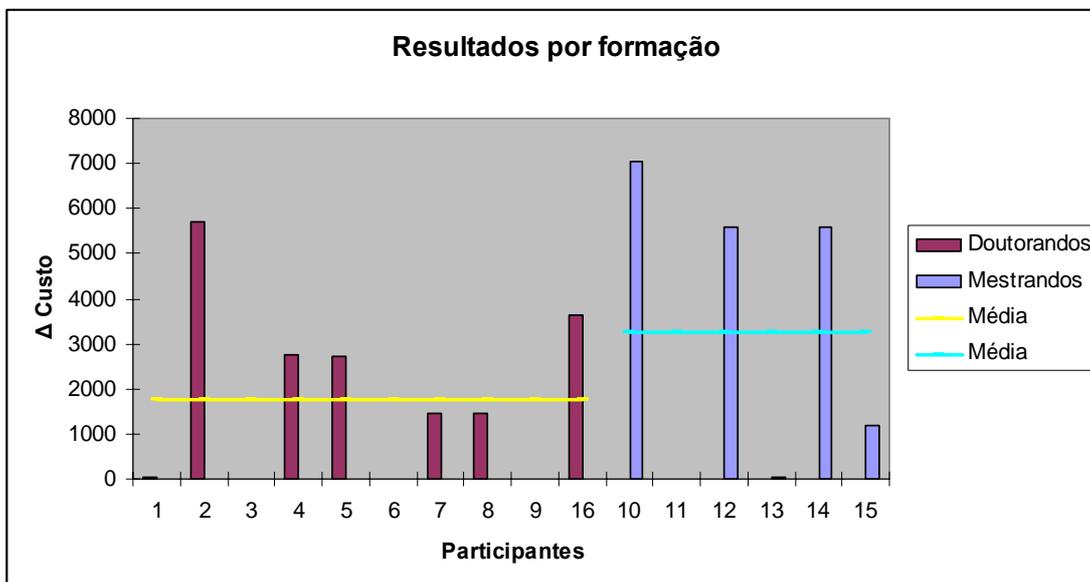


Figura 5.6 – Resultados agrupados por formação acadêmica dos participantes

Tabela 5.4 – Resultados obtidos pelos participantes agrupados por experiência na alocação de recursos humanos

Alguma Experiência Prát.	Tentativas	Tempo Gasto	Custo Equipe	Tempo Projeto	Δ Custo
Participante 3	24	20	14040	42	0
Participante 4	47	28	16800	42	2760
Participante 5	16	30	16760	42	2720
Participante 6	31	38	14040	42	0
Médias	29,5	29	15410	42	1370
Nenhuma Experiência Prát.	Tentativas	Tempo Gasto	Custo Equipe	Tempo Projeto	Δ Custo
Participante 1	6	24	14080	42	40
Participante 2	11	12	19760	42	5720
Participante 7	5	13	15480	42	1440
Participante 8	7	13	15480	42	1440
Participante 9	8	14	14040	42	0
Participante 10	2	13	21080	42	7040
Participante 11	27	36	14040	42	0
Participante 12	4	23	19640	42	5600
Participante 13	24	27	14080	42	40
Participante 14	4	20	19640	42	5600
Participante 15	8	26	15240	42	1200
Participante 16	1	6	10400	45	3640
Médias	8,9	18,9	16080	42,3	2646,7



Figura 5.7 – Resultados agrupados por experiência no desenvolvimento de software dos participantes

Os resultados mostrados nas tabelas acima nos levam a perceber que o custo médio da equipe, além da diferença média em relação à melhor equipe, são significativamente maiores para os mestrandos do que para os doutorandos. Além disso, os participantes sem experiência prática na alocação de recursos humanos chegaram a valores maiores (logo, piores) que aqueles com experiência na atividade. Portanto, percebem-se indícios de que a abordagem proposta pode ser útil para pessoas com menor experiência. Além disso, o fato dos melhores resultados estarem ligados a um maior investimento de esforço na realização da atividade de alocação de recursos humanos (isso pode ser visto tanto pelo número de tentativas quanto pelo tempo gasto pelo participante) pode indicar que a abordagem é mais útil quando o tempo para a tomada de decisão é pequeno. A questão da experiência pode ser reforçada também se analisarmos a experiência no desenvolvimento de software, onde é possível perceber que os resultados das pessoas com experiência foram melhores que os resultados das pessoas sem experiência.

Foi analisada, também, a relação entre a formação de melhores equipes e a experiência gerencial do participante. Entretanto, percebeu-se, durante a análise, que parece ter havido inconsistências nos dados informados nos questionários de caracterização, relativas à experiência dos profissionais em gerência de projetos. Essas inconsistências nos levaram a desconsiderar essas informações durante a análise.

Foi analisada, ainda, a questão do esforço de execução da atividade de alocação de recursos humanos em relação à qualidade das equipes encontradas. Foi realizado um agrupamento dos participantes por tempo gasto no estudo, apresentado na tabela 5.5.

Tabela 5.5 – Resultados obtidos pelos participantes agrupados por tempo de execução do estudo

15>Tempo Gasto>=0	Tempo Gasto	Custo Equipe	Δ Custo
Participante 2	12	19760	5720
Participante 9	14	14040	0
Participante 10	13	21080	7040
Participante 16	6	10400	3640
Participante 7	13	15480	1440
Participante 8	13	15480	1440
Médias	11,8	16040	3213,3
25>Tempo Gasto>=15	Tempo Gasto	Custo Equipe	Δ Custo
Participante 1	24	14080	40
Participante 3	20	14040	0
Participante 14	20	19640	5600
Participante 12	23	19640	5600
Médias	21,8	16850	2810
Tempo Gasto>=25	Tempo Gasto	Custo Equipe	Δ Custo
Participante 4	28	16800	2760
Participante 5	30	16760	2720
Participante 6	38	14040	0
Participante 15	26	15240	1200
Participante 11	36	14040	0
Participante 13	27	14080	40
Médias	30,8	15160	1120

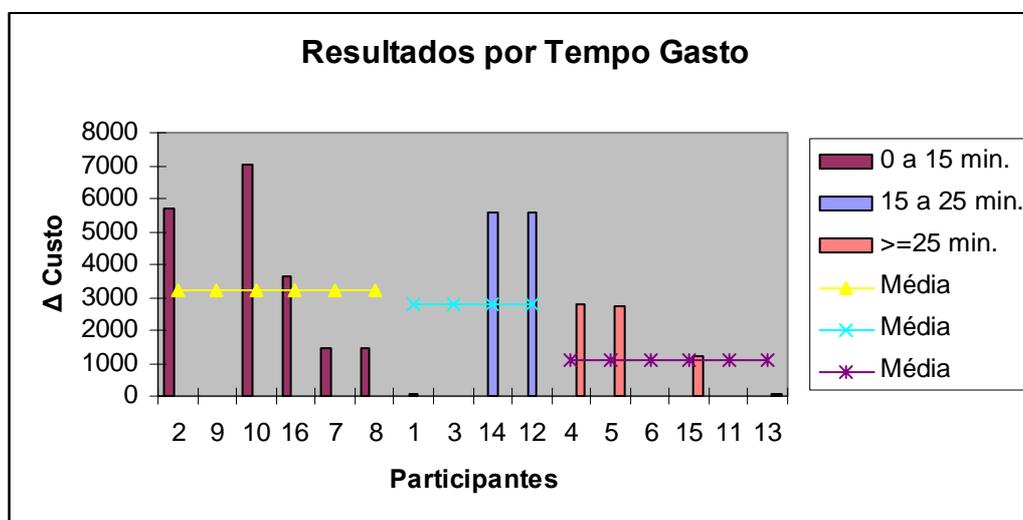


Figura 5.8 – Resultados agrupados por experiência no desenvolvimento de software dos participantes

Os resultados acima nos mostram que, na média, quanto maior o tempo gasto para realizar a atividade, melhor o resultado encontrado, ou seja, menor a distância do custo encontrado para o melhor custo. Isso reforça a idéia de que o apoio automatizado pode ser mais útil em situações onde se disponha de menos tempo para tomar a decisão de

alocação de recursos humanos, ou seja, em que o esforço para realizar a atividade tenha que ser menor.

O gráfico da figura 5.9 apresenta as soluções encontradas e todas as possíveis soluções para o problema. Os pontos roxos representam todas as possíveis soluções para o problema e os triângulos azuis as soluções encontradas pelos participantes do estudo.

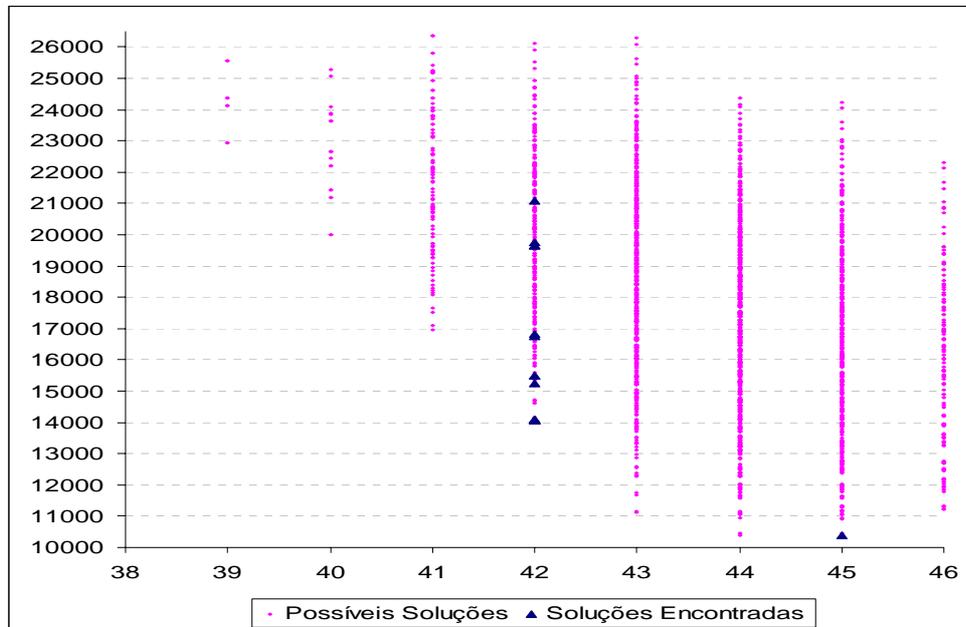


Figura 5.9 – Resultados do Estudo

A partir do gráfico, é possível perceber que, apesar de alguns participantes terem conseguido encontrar a melhor solução (25% encontraram a solução de menor custo para 42 dias), a maior parte dos participantes se distanciou bastante dessa solução. Além disso, é possível perceber o quanto se poderia distanciar da solução ótima, mesmo em uma situação de projeto tão simples (sete atividades e sete profissionais).

Por fim, é importante salientar que mesmo em uma situação de projeto simples, apenas 25% dos participantes conseguiu chegar à melhor solução. Além disso, mesmo aqueles que conseguiram necessitaram de um grande esforço para isso. Apenas um dos participantes que chegou à melhor solução levou menos de vinte minutos (catorze) para encontrar essa solução. Portanto, uma vez que a ferramenta proposta sempre é capaz de chegar à melhor solução (considerando-se que os algoritmos utilizados tenham sido corretamente implementados) e os participantes foram capazes em apenas 25% das vezes, há fortes indícios de que a primeira hipótese alternativa do estudo tenha sido confirmada, ou seja, a utilização da ferramenta proposta melhora a qualidade da equipe gerada (reduz o custo monetário e termina o projeto no tempo determinado para o

projeto) se comparada à qualidade da equipe gerada para o projeto por outros procedimentos *ad-hoc* quando aplicados no mesmo projeto.

5.4 Conclusão

Neste capítulo, apresentamos um estudo experimental que foi realizado para investigar se o apoio sugerido neste trabalho apresenta algum benefício para o gerente de projetos na atividade de alocação de recursos humanos. Foi apresentada, ainda, uma ferramenta de apoio à alocação de recursos humanos em projetos de software, que foi desenvolvida como parte deste trabalho, sendo um dos requisitos para a execução do estudo experimental.

O estudo realizado apresentou indícios de que o apoio fornecido é útil para o gerente no momento da realização da atividade de alocação de recursos humanos em projetos de desenvolvimento de software, uma vez que os algoritmos utilizados na ferramenta garantem que a melhor solução será encontrada e os participantes do estudo nem sempre (75%) conseguiram chegar à melhor solução.

Durante a análise dos resultados, foi identificado, também, que há indícios de que o apoio seja mais útil para profissionais com menor experiência e em situações onde se disponha de pouco tempo para realizar a atividade. Entretanto, para chegarmos a conclusões mais abrangentes e precisas, seria necessária a realização de outros estudos, inclusive em projetos reais, para avaliar se as tendências *in-vitro* se confirmam.

No próximo capítulo, serão apresentadas as conclusões e considerações finais deste trabalho, contribuições e limitações, além de suas perspectivas futuras.

CAPÍTULO 6

Conclusão

6.1 Considerações Finais

As pessoas desempenham um papel fundamental nos projetos de software. Portanto, alocar pessoas, ou seja, determinar quem irá realizar cada atividade em um projeto de software é uma atividade de grande importância.

Neste trabalho, foi apresentada uma proposta para apoiar o gerente de projetos na atividade de alocação de recursos humanos em projetos de software. A proposta consiste em sugerir ao gerente possíveis equipes para o projeto que atendam a uma série de restrições impostas (pela qualificação dos profissionais, pelas datas, entre outras) e que, se necessário, otimizem algum fator envolvido no problema, como o custo ou o tempo do projeto.

Foi desenvolvida uma abordagem baseada em satisfação de restrições e em otimização, incluindo a definição de funções de custo e algoritmos para resolver o problema. Foi, também, desenvolvida uma ferramenta para implementar a abordagem proposta e facilitar o uso pelo gerente do apoio sugerido.

Foi, ainda, planejado e executado um estudo experimental para investigar a relevância do uso da alocação de recursos humanos baseada em satisfação de restrições, proposta neste trabalho.

A abordagem proposta representa um exemplo do uso de técnicas de otimização e inteligência artificial para apoiar a execução de atividades complexas do processo de desenvolvimento de software. Abordagens similares poderiam ser usadas para apoiar outras atividades, tais como a seleção de fornecedores, seleção de componentes para reutilização, escolha da arquitetura de sistemas, escolha de modelos de ciclo de vida, entre outras.

A proposta considera, de certa forma, a preocupação tanto com questões técnicas (quando considera as características) quanto com os objetivos do negócio (quando prioriza fatores como custo e tamanho de equipe). Da mesma forma, abordagens similares poderiam ser adotadas em outras atividades do processo de desenvolvimento

de software, de forma que na execução de cada tarefa houvesse uma maior preocupação com a integração entre as questões técnicas e os objetivos do negócio.

Além disso, apesar de neste trabalho o foco ter sido a alocação em projetos de software, a abordagem poderia ser usada também em outros contextos mais gerais do que este. Os mesmos algoritmos e ferramenta desenvolvidos poderiam ser usados em qualquer problema de alocação de pessoas, não apenas em projetos de software. Com algumas modificações na modelagem e na ferramenta, outras situações de alocação ou de satisfação de restrições poderiam ser resolvidas (não apenas de pessoas).

Nossa abordagem parece ser mais relevante para situações onde haja um número maior de profissionais e atividades envolvidas, ainda que o estudo experimental tenha mostrado que mesmo nas situações mais simples a tarefa é bastante complexa.

A realização do estudo experimental contribuiu para reforçar os indícios de que apoiar a atividade, como se propõe neste trabalho, pode produzir benefícios para um gerente de projetos que precise alocar pessoas a um projeto de software. Este benefício pode ser tanto relativo ao esforço necessário para realizar a tarefa, quanto relativo à qualidade da equipe escolhida, considerando alguns dos fatores associados (como tempo e custo). O estudo encontrou indícios, também, de que o apoio proposto produz um ganho maior para pessoas menos experientes, dado que pessoas com menos experiência tiveram resultados piores em sua execução.

Vale lembrar que o trabalho desenvolvido não se propõe a substituir o gerente de projetos na execução da atividade de alocação de recursos humanos em projetos de software. O objetivo é auxiliar na tomada de decisão, sugerindo opções de alocação que satisfaçam as restrições informadas para problema. Essas sugestões podem funcionar como uma segunda opinião para o gerente, ou uma opção de alocação inicial que deve ser ajustada para as necessidades específicas de cada situação, caso seja necessário, baseado no conhecimento tácito do gerente. Além disso, as equipes sugeridas funcionam como uma forma de justificar decisões ou até mesmo determinar o quanto uma equipe escolhida se desvia da equipe que otimiza um fator envolvido no problema e se isso é aceitável.

6.2 Contribuições

Podemos destacar como principais contribuições deste trabalho:

- Definição, formalização e implementação do algoritmo de alocação baseado em satisfação de restrições e otimização;
- Definição, formalização e implementação de funções de custo associadas à alocação de recursos humanos;
- Desenvolvimento de uma ferramenta que implementa os algoritmos e funções propostos e disponibiliza ao gerente o apoio sugerido neste trabalho;
- Definição, planejamento, execução e análise de um estudo experimental para avaliar a viabilidade da abordagem proposta.

6.3 Limitações e Perspectivas Futuras do Trabalho

É possível que o apoio sugerido no trabalho não seja adequado a todas as situações em que se precise alocar recursos humanos a atividades de um projeto de software. Isso acontece porque diferentes organizações podem adotar diferentes critérios para determinar quem irá realizar cada tarefa. Por exemplo, o gerente pode não ter a liberdade de escolher quem fará parte da equipe, mas sim contar apenas com as pessoas que estiverem disponíveis, não levando em consideração se as pessoas possuem a qualificação necessária para executar as atividades. Mas mesmo neste caso, o apoio proposto poderia ser usado para determinar quem mais se aproxima da qualificação necessária. Em outras situações, a organização pode simplesmente não ter como prática a determinação da qualificação necessária para realizar cada atividade e tomar essas decisões baseando-se apenas em impressões ou no quanto se conhece informalmente de cada profissional, ainda que isso não seja considerado pela literatura uma boa prática. Assim, há a consciência de que para algumas situações a abordagem proposta talvez não seja aplicável. No entanto, isso é bastante aceitável, uma vez que o objetivo do trabalho é fornecer um apoio que possa trazer benefícios em muitas situações, mas não resolver o problema definitivamente. Além disso, é possível que organizações que executem a atividade de maneira *ad-hoc*, uma vez dispostas de um mecanismo de apoio, passem a realizar a atividade de maneira mais disciplinada.

Podemos citar algumas outras limitações do trabalho, que geram oportunidades para estender o trabalho:

- Não é dado apoio na determinação de quais características são necessárias para realizar cada atividade: apenas se possibilita o cadastramento dessas

informações, assumindo-se que o conhecimento necessário para essa determinação é do gerente. Assim, um possível trabalho futuro seria investigar uma forma de guiar o gerente no momento de determinar quais características são necessárias para executar cada atividade do processo de desenvolvimento de software, reunindo e complementando o conhecimento existente no assunto, utilizando os resultados de pesquisas como a de (ACUÑA e JURISTO, 2003), que procura determinar quais habilidades são necessárias para executar cada atividade.

- No algoritmo de alocação não se consideram fins de semana ou feriados, além de ser possível otimizar apenas um fator por vez. Portanto, outro trabalho futuro seria estender o algoritmo de alocação para uso de otimização multidimensional, ou seja, que considere na otimização mais de um fator ao mesmo tempo, como é feito em (DUGGAN *et al.*, 2004). Além disso, considerar também os fins de semana e feriados.
- Na ferramenta de apoio, não se gera nenhum documento descrevendo a equipe escolhida. Esse documento poderia ser parte do plano de alocação de recursos humanos do projeto. Além disso, não se dá apoio a todas as atividades de um processo de gerência de alocação de recursos humanos. Logo, uma perspectiva futura seria estender a ferramenta para que sejam também consideradas as demais atividades de um processo de gerência de alocação de recursos humanos, como o de Schnaider (2003).
- A abordagem proposta não trata a questão do ajuste dos valores utilizados nos modelos envolvidos no problema. Assim sendo, outra perspectiva futura seria fornecer apoio ao ajuste dos modelos, ou seja, auxiliar no ajuste dos valores relativos à produtividade e à exigência de qualificação das atividades, permitindo uma retro-alimentação dos modelos.
- A abordagem proposta e a ferramenta de apoio desenvolvida não foram utilizadas em um ambiente de desenvolvimento de software real. Assim, utilizar a ferramenta de apoio proposta em um projeto de desenvolvimento de software real, procurando determinar se o apoio fornecido é adequado e em que poderia ser aperfeiçoado seria interessante.
- Na ferramenta de apoio, a função de alocação parcial não foi implementada, apesar de ter sido descrita e formalizada neste trabalho. Portanto, implementar

essas funções na ferramenta desenvolvida seria mais uma perspectiva futura do trabalho.

Outras possíveis evoluções do trabalho podem ser citadas:

- Considerar, além das características dos profissionais, características da equipe como um todo, ou seja, determinar características que uma equipe deveria possuir, sendo que cada profissional contribuiria para atender ou não a essas características;
- Melhorar a exibição dos resultados da execução do algoritmo de alocação, permitindo o uso de outras funcionalidades, como filtros, gráficos, e outros;
- Permitir ao gerente escolher uma equipe e checar se a equipe satisfaz a todas as restrições do problema;
- Retornar não apenas equipes que otimizem algum fator, mas que tenham um valor limite para o fator selecionado. Por exemplo, no caso em que se disponha de um certo valor a ser gasto com o pessoal e que se deseje saber quais são as equipes em que o custo limite não é ultrapassado.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDEL-HAMID, T., MADNICK, S.E., 1991, *Software project dynamics: an integrated approach*, Prentice Hall, Upper Saddle River, Estados Unidos.
- ACUÑA, S.T., JURISTO, N., 2003, “Modelling Human Competencies in the Software Process”, In: *ProSim - Software Process Simulation and Modeling Workshop*, Portland, Estados Unidos, Maio.
- ADENSO-DIAZ, B., TUYA, J., CABAL, M.J.S., GOITIA-FUERTES, M., 2003, “DSS for Rescheduling of Railway Services Under Unplanned Events”, In: MORA, M., FORGIONE, G.A., GUPTA, J.N.D. (eds.), *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*, 1ª Edição, IDEA Group Publishing, Hershey, Estados Unidos.
- AGARWAL, R., FERRAT, T.W., 2000, “Retention and the career motives of IT professionals”, In: *Proceedings of the 2000 ACM SIGCPR conference on Computer personnel research*, pp. 158-166, Chicago, Estados Unidos, Abril.
- ALBIN-AMIOT, H., COINTE, P., GUÉHÉNEUC, Y., JUSSIEN, N., 2001, “Instantiating and Detecting Design Patterns: Putting Bits and Pieces Together”, In: *Proceedings of the 16th IEEE conference on Automated Software Engineering (ASE'01)*, pp. 166-173, San Diego, Estados Unidos, Novembro.
- AMARAL, E.A.G., *Empacotamento de Experimentos em Engenharia de Software*, Tese de Mestrado, COPPE/UFRJ, Programa de Engenharia de Sistemas e Computação, Rio de Janeiro, 2003.
- BALMAS, F., WERTZ, H., SINGER, J., 2000, “Understanding Program Understanding”, In: *Proceedings of the 8th International Workshop on Program Comprehension (IWPC'00)*, pp. 256-256, Limerick, Irlanda, 2000.
- BARRETO, A.S., BARROS, M.O., WERNER, C.M.L., 2005, “Apoio à Alocação de Recursos Humanos em Projetos de Software: Uma Abordagem Baseada em Satisfação de Restrições”, In: *Anais do IV Simpósio Brasileiro de Qualidade de Software*, pp. 79-93, Porto Alegre, Brasil, Junho.
- BARRETO, A.S., BARROS, M.O., WERNER, C.M.L., 2005, “Staffing a Software Project: a Constraint Satisfaction Approach”, In: *7th International Workshop on Economics-Driven Software Engineering Research*, St. Louis, Estados Unidos, Maio.

- BARROS, M.O., *Gerenciamento de Projetos Baseado em Cenários: uma Abordagem de Modelagem Dinâmica e Simulação*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2001.
- BAYES, T., 1763, “An essay toward solving a problem in the doctrine of chances”, In: *Philosophical Transactions of London*, V. 53, pp. 370–418.
- BELL, D. E., RAIFFA, H., TVERSKY, A., 1988, *Decision Making: Descriptive, Normative, and Prescriptive Interactions*, 1ª Edição, Cambridge University Press, Nova York, Estados Unidos.
- BITNER, J., REINGOLD, E. M., 1975, “Backtrack Programming Techniques”, *Communications of the ACM*, v.18, n. 11, pp. 651–655.
- BOEHM, B., SULLIVAN, K., 2000, “Software economics: a roadmap”, In: *Proceedings of the Conference on The Future of Software Engineering*, pp. 319—344, Limerick, Irlanda.
- BROOKS, F., 1987, “No Silver Bullet: Essence and accidents of software engineering”, *IEEE Computer*, V. 20, N. 4, pp. 10-20.
- BROOKS, F., 1987, Report on the Defense Science Board Task Force on Military Software, Office of the Undersecretary of Defense for Acquisition, Dept. of Defense, Washington, Estados Unidos, Setembro.
- CENTENO, M.A., CARRILLO, M., 2001, “Challenges of Introducing Simulation as a Decision Making Tool”, In: *Proceedings of the 2001 Winter Simulation Conference*, 2001, pp. 17-21, Arlington, Estados Unidos, Dezembro.
- CHU, W.W., NGAI, P.H., 1988, “A Dynamic Constraint-Directed Ordered Search Algorithm for Solving Constraint Satisfaction Problems”, In: *Proceedings of the first international conference on Industrial and engineering applications of artificial intelligence and expert systems*, v. 1, pp. 116-125, Tullahoma, Estados Unidos.
- CLEMEN, R.; REILLY, T., 2001, *Making Hard Decisions With Decision Tools*, 1ª Edição, Duxbury Press, Belmont, Estados Unidos.
- COELLO, C.A.C, VELDHUIZEN, D.A.V., LAMONT, G.B., 2002, *Evolutionary Algorithms for Solving Multiobjective Problems*, 1ª Edição, Kluwer academic Publishers, Nova York, Estados Unidos.
- COLLOFELLO, J., RUS, I., HOUSTON, D., SYCAMORE, D., SMITH-DANIELS, D., 1998, “A System Dynamics Software Process Simulator for Staffing Policies

- Decision Support”, In: Proceedings of The Thirty-First Annual Hawaii International Conference on System Sciences, pp. 103-111, Kohala Coast, Estados Unidos, Janeiro.
- COLLOFELLO, J., RUS, I., HOUSTON, D., SYCAMORE, D., SMITH-DANIELS, D., 1998, “A System Dynamics Software Process Simulator for Staffing Policies Decision Support”, In: *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences*, pp. 103-111, Kohala Coast, Estados Unidos.
- DE FINETTI, B., 1937, “La prévision: Ses lois logiques, ses sources subjectives”, In: *Annales de l'Institut Henri Poincaré* 7, pp. 1–68, Paris, França.
- DINGSOYR, T., ROYRVIK, E., 2001, “Skills Management as Knowledge Technology in a Software Consultancy Company”, In: *Lecture Notes In Computer Science*, v. 2176, Springer-Verlag, pp. 96-105, Londres, Inglaterra.
- DoD, 1983, Software Initiative, Strategy for a Dod Software Initiative, Dept. of Defense Report, Washington, Estados Unidos, Novembro.
- DONZELLI P., RUS I., AND SHULL F., 2003, “Decision support for software inspection use - Combining process modeling and simulation with empirical studies”, In: *Proceedings of The 28th NASA/IEEE Software Engineering Workshop*, Greenbelt, Estados Unidos, Dezembro.
- DUGGAN, J., BYRNE, J., LYONS, G., 2004, “A Task Allocation Optimizer for Software Construction”, *IEEE Software*, V. 21, N. 3, pp. 76-82.
- FORGIONE, G.A., GUPTA, J.N.D., MORA, M., 2003, “Decision Making Support Systems: Achievements, Challenges and Opportunities”, In: MORA, M., FORGIONE, G.A., GUPTA, J.N.D., *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*, 1ª Edição, IDEA Group Publishing, Hershey, Estados Unidos.
- GARBER, M.F., 2002, *Estruturas Flutuantes para a Exploração de Campos de Petróleo no Mar (FPSO): Apoio à Decisão na Escolha do Sistema*, Tese de M. Sc., Universidade de São Paulo, São Paulo, Brasil.
- GAVANELLI, G., 2002, *Interactive Constraint Satisfaction Problems for Artificial Vision*, Tese de Ph. D., Department of Engineering, University of Ferrara, Itália.
- GERSON, M, CHIEN, I.S., RAVAL, V., EPPLEY, E.C., 1992, “Computer Assisted Decision Support Systems: Their Use in Strategic Decision Making”, In:

Proceedings of the 1992 ACM SIGCPR conference on Computer personnel research, pp.152-160, Cincinnati, Estados Unidos.

GIBSON, R., 2003, “Knowledge Management Support for Decision Making in the Pharmaceutical Industry”, In: MORA, M., FORGIONE, G.A., GUPTA, J.N.D., *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*, 1ª Edição, IDEA Group Publishing, Hershey, Estados Unidos.

GOLOMB, S.W., BAUMERT, L.D., 1965, “Backtrack programming”, *Journal of the ACM*, v. 12, n. 4, pp. 516–524.

GORRY, G.A., SCOTT-MORTON, M. S., 1971, “A framework for management information systems”, *Sloan Management Review*, v.3, n.1, pp. 55-71.

HARALICK, R.M., ELLIOTT, G.L, 1980, “Increasing tree search efficiency for constraint satisfaction problems”, *Artificial Intelligence*, v. 14, n. 3, pp. 263–313.

HUANG, L.C, HUANG, K.S., HUANG, H.P., JAW, B.S., 2004, “Applying Fuzzy Neural Network in Human Resource Selection System”, In: NAFIPS – North American Fuzzy Information Processing Society – 2004, Banff, Canadá.

ISO/IEC 12207:1995/Amd 1:2002, Information Technology – Software Life-Cycle Processes.

KAHNEMAN, D., SLOVIC, P., TVERSKY, A., 1982, *Judgment Under Uncertainty: Heuristics and Biases*, 1ª Edição, Cambridge University Press, Cambridge, Inglaterra.

KAHNEMAN, D., TVERSKY, A., 1979, “Prospect theory: An analysis of decision under risk”, *Econometrica*, v. 47, pp.263–291.

KALE, L.V., 1990, A Perfect Heuristic for the N Non-Attacking Queens Problem, *Information Processing Letters*, v. 34, n. 4, pp. 173–178.

KARLSTROM, D., RUNESON, P., 2002, “Decision Support for Extreme Programming Introduction and Practice Selection”, In: *Proceeding of the SEKE’02*, pp. 835-841, Ischia, Itália.

KOONTZ, H., O’DONNELL, C., K. WEIHRICH, H., 1980, *Management*, 7ª edição, Nova York, Estados Unidos, McGraw-Hill.

- KUCHCINSKY, K., 2003, “Constraints-Driven Scheduling and Resource Assignment”, *ACM Transactions on design Automation of Electronic Systems*, v. 8, n. 3, pp. 355-383.
- KUMAR, V., 1992, “Algorithms for Constraint-Satisfaction Problems: A Survey”, *Artificial Intelligence Magazine*, v.13, n. 1, pp. 32–44.
- LAWLER, E.L., WOOD, D.E., 1966, “Branch-and-bound methods: a survey”, *Operations Research*, v. 14, n. 4, pp. 699–719.
- LEE, P.C.B., 2002, “The social context of turnover among information technology professionals”, In: Proceedings of the 2002 ACM SIGCPR conference on Computer personnel research, pp. 145-153, Kristiansand, Noruega, Maio.
- LITTLE, J.D.C., 1970, “Models and managers: The concept of decision calculus”, *Management Science*, v. 16, pp. 466-485.
- MORA, M., FORGIONE, G.A., GUPTA, J.N.D., 2003, *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*, 1ª Edição, IDEA Group Publishing, Hershey, Estados Unidos.
- MURCH, R., 2000, *Project Management – Best Practices for IT Professionals*, 1ª Edição, Estados Unidos, Prentice Hall.
- NBR ISO 10006, 2000, *Gestão da Qualidade – Diretrizes para a Qualidade no Gerenciamento de Projetos*, Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil.
- NYKAKEN, P., 2003, “On the Ontology of a Decision Support System in Health Informatics”, In: MORA, M., FORGIONE, G.A., GUPTA, J.N.D., *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*, 1ª Edição, IDEA Group Publishing, Hershey, Estados Unidos.
- PARÉ, G., TREMBLAY, M., LALONDE, P., 2001, “Workforce retention: what do IT employees really want?”, In: Proceedings of the 2001 ACM SIGCPR conference on Computer personnel research, pp. 1-10, San Diego, Estados Unidos, Abril.
- PFLIEGER, S.L., 2004, *Engenharia de Software – Teoria e Prática*, 2ª edição, São Paulo, Prentice Hall.
- PMI, 2004, *PMBOK - Project Management Body of Knowledge*, 3ª Edição, Project Management Institute.

- PRATT, J. W., RAIFFA, H., SCHLAIFER, R., 1964, “The foundations of decision under uncertainty: An elementary exposition”, *Journal of the American Statistical Association*, v. 59, pp. 353–375.
- PURDOM, P., 1983, “Search Rearrangement Backtracking and Polynomial Average Time”, *Artificial Intelligence*, v. 21, pp. 117–133.
- PURDOM, P., BROWN, C, 1981, “An Average Time Analysis of Backtracking”, *SIAM Journal of Computing*, v.10, n. 3, pp. 583–593.
- PURDOM, P., BROWN, C., 1982, “An Empirical Comparison of Backtracking Algorithms”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 4, n.1, pp. 309–316.
- PURDOM, P., BROWN, C., ROBERTSON, E. L., 1981, “Backtracking with Multi-Level Dynamic Search Rearrangement”, *Acta Informatica*, v.15, pp. 99–113.
- RAFFO, D.M., HARRISON, W., VANDEVILLE, J., 2002, “Software Process Decision Support: Making Process Tradeoffs Using a Hybrid Metrics, Modeling and Utility Framework”, *Proceeding of the SEKE’02*, pp. 803-809, Ischia, Itália.
- RAIFFA, H, 1968, *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*, 1ª Edição, Random House, Nova York, Estados Unidos.
- RAMSEY, F.P., 1931, *The Foundations of Mathematics and other Logical Essays*, R.B. Braithwaite, Londres, Inglaterra.
- REIFER, D., 2002, “A Little Bit of Knowledge Is a Dangerous Thing”, *IEEE Software*, v. 19, n. 3 (Maio/Junho), pp. 14-15.
- RINALDI, F.M., BAIN, D., 2003, “Using Decision Support Systems to Help Policy Makers Cope With Urban Transport Problems”, In: MORA, M., FORGIONE, G.A., GUPTA, J.N.D., *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*, 1ª Edição, IDEA Group Publishing, Hershey, Estados Unidos.
- RUHE, G., 2003, “Software Engineering Decision Support - A New Paradigm for Learning Software Organizations”, *Proceedings of the 4th Workshop on Learning Software Organizations*, pp. 104-113, Chicago, Estados Unidos.
- RUHE, G., EBERLEIN, A., PFAHL, D., 2002, “Quantitative WinWin – A New Method for Decision Support in Requirements Negotiation”, In: *Proceeding of the SEKE’02*, pp. 159-166, Ischia, Itália.

- RUHE, G., *Software Engineering Decision Support*, ICORE Research Report, Vol. 2, pp. 153-162, 2003.
- RUS, I., HALLING, M., BIFFL, S., 2003, “Supporting Decision-Making in Software Engineering With Process Simulation and Empirical Studies”, *International Journal of Software Engineering and Knowledge Engineering*, v. 13, N. 5, pp. 531-545.
- RUS, I.; BIFFL, S.; HALLING, M., 2002, “Systematically Combining Process Simulation and Empirical Data in Support of Decision Analysis in Software Development”, In: *Proceeding of the SEKE’02*, pp. 827-833, Ischia, Itália.
- RUSKOVA, N.A., 2002, “Decision support system for human resources appraisal and selection”, In: First International Symposium “Intelligent Systems”, pp. 354-357, Varna, Bulgaria, Setembro.
- RUSSELL, S.; NORVIG, P., 2003, *Artificial Intelligence: A Modern Approach*, 2ª Edição, Prentice Hall, Englewood Cliffs, Estados Unidos.
- SAMSON, D., 1988, *Managerial Decision Analysis*, Irwin, Homewood, Estados Unidos.
- SAVAGE, L. J., 1954, *The Foundations of Statistics*, Wiley, Nova York, Estados Unidos.
- SCHNAIDER, L.R.C., 2003, Planejamento da alocação de recursos humanos em ambientes de desenvolvimento de software orientados à organização, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SCOTT-MORTON, M., 1971, *Management decision systems: Computer-based support for decision making*, Harvard University Press, Boston, Estados Unidos.
- SEI, 2001, People Capability Maturity Model Integration, Version 2.0, URL: <http://www.sei.cmu.edu>, verificado em julho de 2005.
- SEI, 2002, *Capability Maturity Model Integration, Version 1.1 Staged Representation*, URL: <http://www.sei.cmu.edu>.
- SMITH, J.E., WINTERFELDT, D.V., 2004, “Decision Analysis in Management Science”, *Management Science*, v. 50, n. 5, pp. 561–574.
- SOFTEX, 2005, MPS.BR - Melhoria de Processo do Software Brasileiro – Guia Geral, Versão 1.0, Abril.

- SOMMERVILLE, I., RODDEN, T., 1996, "Human, social and organisational influences on the software process", In: FUGGETA A., WOLF, A (eds), *Trends in Software Process*, 1ª Edição, Capítulo 4, John Wiley and Sons, Somerset, Estados Unidos.
- SOUZA, G.S., 2003, Representação da Distribuição do Conhecimento, habilidades e experiências através da estrutura organizacional, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SPRADLIN, T., 1997, *A Lexicon of Decision Making*, URL: faculty.fuqua.duke.edu/daweb/lexicon.htm, verificado em julho de 2005.
- SPRAGUE, R.H., CARLSON, E.D., 1982, *Building effective decision support systems*, Prentice Hall, Englewood Cliffs, Estados Unidos.
- STANDISH GROUP, 1994, *The Chaos Report (1994)*, Online: http://www.standishgroup.com/sample_research/chaos_1994_1.php, verificado em julho de 2005.
- STANDISH GROUP, 2003, *Latest Standish Group CHAOS Report Shows Project Success Rate Have Improved By 50%*, Online: <http://www.standishgroup.com/press/article.php?id=2>, verificado em julho de 2005.
- STONE, H.S., STONE, J., 1986, "Efficient Search Techniques: An Empirical Study of the N-Queens Problem", Technical Report RC 12057, IBM T. J. Watson Research Center, Yorktown Heights, Nova York, Estados Unidos.
- SUMNER, M. NIEDERMAN, F., 2002, "The impact of gender differences on job satisfaction, job turnover, and career experiences of information systems professionals", In: Proceedings of the 2002 ACM SIGCPR conference on Computer personnel research, pp. 154-162, Kristiansand, Noruega, Maio.
- SY, N.T., DEVILLE, Y., 2003, "Consistency techniques for interprocedural test data generation", In: Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering, v. 28, n. 5, pp. 108-117.
- THATCHER, J.B.; YONGMEI, L.; STEPINA, L.P., 2002, "The role of the work itself: an empirical examination of intrinsic motivation's influence on IT workers attitudes and intentions", In: Proceedings of the 2002 ACM SIGCPR conference on Computer personnel research, pp. 25-33, Kristiansand, Noruega, Maio.
- THAYER, R.H., 1997, *Software Engineering Project Management*, 2ª Edição, IEEE Computer Society Press, Los Alamitos, Estados Unidos.

- THAYER, R.H., 1997, *Software Engineering Project Management*, 2ª Edição, IEEE Computer Society Press, Los Alamitos, Estados Unidos.
- TSANG, E.P.K., 1993, *Foundation of constraint satisfaction*, Academic Press, Nova York, Estados Unidos.
- VILLELA, K., 2004, *Definição e Construção de Ambientes de Desenvolvimento de Software Orientados a Organização*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- VON NEUMANN, J., MORGENSTERN, O., 1944, *Theory of Games and Economic Behavior*, 1ª Edição, Princeton University Press, Princeton, Estados Unidos.
- WALLACE, R.J, FREUDER, E.C., 1996, “Heuristic Methods for Over-Constrained Constraint Satisfaction Problems”, In: *Lecture Notes In Computer Science*, v. 1106, pp. 207-216.
- WASTELL, D., ARBAOUI, S., LONCHAMP J., MONTANGERO, C., 1999, “The Human Dimension of the Software Process”, In: *Software Process: Principles, Methodology, Technology*, pp. 165-200, Springer-Verlag, Londres, Inglaterra.
- WEI, C., HU, P. J., CHEN, H., 2002, “Design and Evaluation of a Knowledge Management System”, *IEEE Intelligent Systems*, v. 19, n. 3 (Maio/Junho), pp. 56-59.
- WOHLIN, C., 2004, “Are Individual Differences in Software Development Performance Possible to Capture Using a Quantitative Survey?”, *Empirical Software Engineering*, V. 9, N.3, pp. 211-228.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M.C.; REGNELL, B.; WESSLÉN, A., *Experimentation in Software Engineering: an Introduction*. Kluwer Academic Publishers, Massachusetts, 2000.
- WOODS, S., YANG, Q., 1996, “The program understanding problem: analysis and a heuristic approach”, In: *Proceedings of the 18th international conference on Software engineering*, pp. 6-15, Berlim, Alemanha.
- WYSOCKI, R.K, YOUNG, J., 1990, *Information Systems: Management Principles in Action*, 1ª Edição, John Wiley & Sons, Nova York, Estados Unidos.

ANEXO I

Questionário para Estudo sobre Atividade de Alocação de Recursos Humanos em Projetos de Software

Q1	Questionário para Estudo sobre Atividade de Alocação de Recursos Humanos em Projetos de Software
-----------	---

Nome: _____

Determine seu nível de formação.

- | | |
|-------------------------------------|-----------------------------------|
| <input type="checkbox"/> Graduando | <input type="checkbox"/> Graduado |
| <input type="checkbox"/> Mestrando | <input type="checkbox"/> Mestre |
| <input type="checkbox"/> Doutorando | <input type="checkbox"/> Doutor |

Determine a sua experiência no desenvolvimento de software ?

- Eu nunca desenvolvi software
- Eu desenvolvi software para uso pessoal
- Eu desenvolvi software como parte de trabalhos de curso (____ projetos)
- Eu desenvolvi software em ambiente acadêmico (____ anos)
- Eu desenvolvi software em ambiente industrial (____ anos)

Determine a sua experiência na gerência de projetos de desenvolvimento de software ?

- Eu nunca fui gerente de projetos de software
- Eu fui líder de grupos de trabalho em projetos de curso
- Eu fui líder em poucos (≤ 3) projetos acadêmicos (____ anos)
- Eu fui líder em muitos (> 3) projetos acadêmicos (____ anos)
- Eu fui líder em poucos (≤ 3) projetos na indústria (____ anos)
- Eu fui líder em muitos (> 3) projetos na indústria (____ anos)

Determine a sua experiência na montagem e gerenciamento de equipes em projetos de software ?

- Nenhuma
- Estudo em cursos ou livros
- Praticado em projetos de curso
- Praticado em projetos acadêmicos
- Praticado em diversos (>3) projetos acadêmicos
- Praticado em projetos na indústria
- Praticado em diversos (>3) projetos na indústria

ANEXO II

Instruções da Execução do Estudo Experimental Distribuídas aos Participantes do Estudo

i - Descrição do Problema

Você é um gerente de projetos de sua empresa e está responsável por alocar recursos humanos, ou seja, determinar quais profissionais irão executar cada atividade de um projeto. O planejamento inicial, baseado na produtividade de profissionais com média experiência, estimou que o projeto será concluído em 46 dias. No entanto, é fundamental que o projeto seja concluído em, no máximo, 42 dias. A empresa deseja, ainda, que os gastos com o projeto sejam os menores possíveis, desde que as atividades sejam desempenhadas apenas por pessoas que tenham a qualificação necessária e estejam disponíveis nos períodos estimados para as atividades. Assim sendo, você precisa montar a equipe levando em consideração:

1. Um profissional só pode ser alocado a uma atividade se tiver pelo menos todas as características requeridas pela atividade, em intensidade igual ou superior à intensidade exigida.
2. Um profissional só pode ser alocado a uma atividade se não estiver indisponível no período em que a atividade precisa ser realizada. No caso deste estudo, um profissional está indisponível quando já está alocado a outra atividade no mesmo período. Vale salientar que o primeiro e o último dias dos períodos reservados para as atividades são considerados parte dos períodos. Portanto, os profissionais estão indisponíveis nesses dias. Por exemplo, uma atividade que vai de 10/03 a 15/03 tem 6 dias.
3. O tempo necessário para a conclusão do projeto não deve exceder 42 dias. Assim sendo, é preciso utilizar os profissionais mais experientes (ou seja, de maior produtividade) de forma a reduzir o tempo do projeto. Terminar o projeto em menos que 42 dias não representa vantagem alguma para a empresa, a não ser que o custo com a equipe não seja maior que o melhor custo possível com 42 dias.

4. É necessário atentar, também, para o custo do projeto relativo aos gastos com os profissionais escolhidos. Este custo é calculado com base no custo da hora do profissional e o número de horas da atividade (número de dias * 8). Para simplificar, fins de semana e feriados serão tratados como dias úteis.

Assim, o objetivo é chegar à equipe que atenda às exigências das atividades, faça com que o projeto seja concluído em, no máximo, 42 dias, e represente o menor custo possível para o projeto, dadas as exigências anteriores.

Você irá realizar o estudo através de um software desenvolvido para este fim. Neste software, estão listadas as atividades e os profissionais envolvidos no problema (você receberá também essas informações impressas). Você deverá escolher a equipe (escolher um profissional para cada atividade) e uma vez selecionada a equipe, você deve clicar no botão “Calcular”. Caso a equipe satisfaça as restrições, serão exibidos o custo e o tempo relacionados à equipe e você poderá finalizar o estudo ou tentar novamente. Caso contrário, será exibida uma mensagem indicando que a equipe escolhida não atende às restrições, e da mesma forma você poderá finalizar o estudo ou tentar novamente.

Para finalizar o estudo, basta clicar no botão “Finalizar Estudo”. Você deve preencher seu nome e indicar o tempo gasto no estudo (em minutos). O tempo calculado desde o início do uso do aplicativo é apresentado como padrão, mas pode ser modificado.

Muito Obrigado por sua participação!

ii – Lista de Profissionais do Projeto de Exemplo

Tabela i – Lista de Profissionais

Profissional	Valor / Hora	Características Possuídas	Valor
Kenny Carlberg	\$15,00	Banco de Dados	2
		Java	2
		Negociação	2
		Trabalho em Equipe	2
Scott Kaplan	\$30,00	Relacionamento com o Público	2
		Negociação	2
		Técnicas de Levantamento de Requisitos	2
		Telecomunicações	3
		Trabalho em Equipe	2
		Análise Orientada a Objetos	2
		Banco de Dados	2

		Projeto Orientado a Objetos	2
Kendall Cheers	\$50,00	Relacionamento com o Público	3
		Negociação	3
		Telecomunicações	3
		Trabalho em Equipe	3
		Banco de Dados	3
		Projeto Orientado a Objetos	3
		Java	3
		Técnicas de Testes	2
Helene Scarberry	\$20,00	Técnicas de Levantamento de Requisitos	2
		Trabalho em Equipe	2
		Análise Orientada a Objetos	3
		Banco de Dados	2
		Java	2
		Técnicas de Testes	2
Joane Betzer	\$10,00	Negociação	2
		Trabalho em Equipe	2
		Banco de Dados	2
		Java	2
Suzi Fenninger	\$15,00	Banco de Dados	2
		Java	3
Philip Mckeever	\$75,00	Relacionamento com o Público	3
		Negociação	3
		Telecomunicações	4
		Trabalho em Equipe	3
		Análise Orientada a Objetos	3
		Banco de Dados	3
		Projeto Orientado a Objetos	3
		Java	3

iii – Lista de Atividades do Projeto de Exemplo

Tabela ii – Lista de Atividades

Atividade	Data Início	Data Fim	Características Necessárias	Valor
Especificação de Requisitos	22/3/2005	29/3/2005	Relacionamento com o Público	2
			Negociação	2
			Técnicas de Levantamento de Requisitos	2
			Telecomunicações	2
Análise do Sistema	30/3/2005	7/4/2005	Trabalho em Equipe	2
			Análise Orientada a Objetos	2
Projeto do Sistema	8/4/2005	14/4/2005	Trabalho em Equipe	2
			Projeto Orientado a Objetos	2
			Banco de Dados	2

Construção do Módulo 1	15/4/2005	28/4/2005	Banco de Dados Java	2 2
Construção do Módulo 2	15/4/2005	30/4/2005	Banco de Dados Java	2 2
Construção do Módulo 3	15/4/2005	29/4/2005	Banco de Dados Java	2 2
Testes do Sistema	1/5/2005	6/5/2005	Técnicas de Testes Java	2 2