

UNIOESTE – Universidade Estadual do Oeste do Paraná

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Integrando o *Framework i ao Processo de Gerência de
Tempo**

Felipe Augusto Henn

CASCABEL

2012

FELIPE AUGUSTO HENN

**INTEGRANDO O *FRAMEWORK* i* AO PROCESSO DE GERÊNCIA DE
TEMPO**

Monografia apresentada como requisito parcial
para obtenção do grau de Bacharel em Ciência
da Computação, do Centro de Ciências Exatas
e Tecnológicas da Universidade Estadual do
Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Dr. Victor Francisco Araya
Santander

CASCADEL

2012

FELIPE AUGUSTO HENN

**INTEGRANDO O *FRAMEWORK* i* AO PROCESSO DE GERÊNCIA DE
TEMPO**

Monografia apresentada como requisito parcial para obtenção do Título de *Bacharel em Ciência da Computação*,
pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos
professores:

Prof. Dr. Victor Francisco Araya Santander
(Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Elder Elisandro Schemberger
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Esp. Cristiano Ferreira de Souza
Centro Técnico Superior do Oeste Paraense

Cascavel, 17 de Outubro de 2012.

AGRADECIMENTOS

Agradeço primeiramente a meus pais José Henn Neto e Juraci Salete Holodniak Henn, e meus irmãos Marcos Vinicius Henn e Caroline Henn, por todo o apoio e por tudo o que me ensinaram. Agradeço também a todos meus amigos, mas principalmente Cauê Schunch Lopes, Rodrigo Trage, Bruno Eduardo Soares e Heitor Faccioni, que estiveram ao meu lado nos bons e maus momentos, e me fizeram entender por que uma graduação é lembrada como um período tão especial. Por fim agradeço a todos os meus professores, em especial Victor Francisco Araya Santander, por me auxiliaram tanto no meu crescimento profissional quanto pessoal.

Lista de Figuras

2.1 Exemplo de EAP para definir casos de teste.....	11
2.2 Exemplo de um cronograma.....	25
3.1 Legenda dos principais componentes do <i>framework</i> i*.....	32
3.2 Exemplo de diagrama SD.....	33
3.3 Possíveis associações entre atores.....	33
3.4 Exemplo de diagrama SR.....	35
3.5 Possíveis ligações de contribuição.....	36
4.1 Representação de atores participantes de um projeto.....	44
4.2 Representação de dependência do tipo tarefa entre dois atores.....	45
4.3 Representação de dependência do tipo objetivo entre dois atores.....	46
4.4 Representação de dependência do tipo recurso entre dois ator.....	46
4.5 Representação de dependência do tipo objetivo-soft entre dois atores.....	47
4.6 Modelo SD do projeto com foco nos atores envolvidos no projeto.....	48
4.7 Modelo SD do sistema com foco nos atores envolvidos no projeto.....	49
4.8 Modelo SR do projeto com foco nos atores envolvidos no projeto.....	51
4.9 Modelo SR do sistema com foco nos atores envolvidos no projeto.....	52
4.10 Detalhamento de recursos.....	53
4.11 Casos de uso obtidos com base na Figura 4.7.....	67
4.12 Cronograma para o modelo SR apresentado na Figura 4.10.....	71
5.1 Modelo SD do projeto.....	76

5.2 Modelo SD do sistema.....	77
5.3 Modelo SR do projeto.....	78
5.4 Modelo SR do sistema.....	79
5.5 Cronograma do sistema a ser desenvolvido.....	83
6.1 Relação entre diretrizes e passos.....	88

Lista de Tabelas

2.1 Exemplo de dicionário de EAP para definir casos de teste.....	11
4.1 <i>Template</i> para listar atividades.....	54
4.2 <i>Template</i> da atividade gerar <i>check-list</i> de testes, após listagem de atividades.....	55
4.3 <i>Template</i> da atividade gerar <i>check-list</i> de testes, após estimativa de recursos.....	59
4.4 <i>Template</i> da atividade gerar <i>check-list</i> de testes, após estimativa de duração.....	68
5.1 <i>template</i> da atividade S – 7.2 – definir tipo de selo.....	80
5.2 <i>template</i> da atividade P – 5.2 – alterar documentação.....	80
5.3 <i>template</i> da atividade S – 2 – cadastro de selos.....	82

Sumário

Lista de figuras	vi
Lista de tabelas	viii
Sumário	ix
Resumo	xiii
1 Introdução	1
1.1 Contexto.....	1
1.2 Motivações.....	3
1.3 Proposta.....	4
1.4 Contribuições esperadas.....	5
2 Gerenciamento de Tempo	6
2.1 A gerência de projetos.....	6
2.2 A gerência de tempo.....	8
2.2.1 Definir atividades.....	9
2.2.1.1 Entradas para definir as atividades.....	9
2.2.1.2 Ferramentas e técnicas para definir as atividade.....	13
2.2.1.3 Saídas da definição.....	13
2.2.2 Sequenciar as atividades.....	14
2.2.2.1 Entradas para sequenciar as atividades.....	14
2.2.2.2 Ferramentas e técnicas para sequenciar as atividades.....	16
2.2.2.3 Saídas do sequenciamento de atividades.....	17

2.2.3 Estimar recursos das atividades.....	17
2.2.3.1 Entradas para estimar os recursos das atividades.....	17
2.2.3.2 Ferramentas e técnicas para estimar os recursos das atividades.....	18
2.2.3.3 Saídas da estimativa de recursos das atividades.....	19
2.2.4 Estimar duração das atividades.....	20
2.2.4.1 Entradas para estimar a duração das atividades.....	20
2.2.4.2 Ferramentas e técnicas para estimar a duração das atividades.....	21
2.2.4.3 Saídas da estimativa de duração das atividades.....	22
2.2.5 Desenvolver o cronograma.....	22
2.2.5.1 Entradas para desenvolver o cronograma.....	22
2.2.5.2 Ferramentas e técnicas para desenvolver o cronograma.....	23
2.2.5.3 Saídas do desenvolvimento do cronograma.....	24
2.2.6 Controlar o cronograma.....	24
2.2.6.1 Entradas para controlar o cronograma.....	25
2.2.6.2 Ferramentas e técnicas para controlar o cronograma.....	26
2.2.6.3 Saídas do controle de cronograma.....	27
2.3 Considerações finais.....	27
3 O <i>Framework</i> i*	29
3.1 Modelagem organizacional.....	29
3.2 <i>Framework</i> i*.....	31
3.2.1 Modelo de dependências estratégicas (SD).....	32
3.2.2 Modelo de razões estratégicas (SR).....	34
3.3 Considerações finais.....	37
4 Integrando o <i>Framework</i> i* ao processo de gerência de tempo	39
4.1 Introdução.....	39

4.2 As Diretrizes.....	41
4.3 A proposta.....	41
4.3.1 Passo 1: gerar os modelos de dependências estratégicas SD.....	43
4.3.1.1 Passo 1 – Diretriz 1: identificar os atores participantes do projeto.....	43
4.3.1.2 Passo 1 – Diretriz 2: identificar dependências entre os atores.....	44
4.3.1.3 Passo 1 – Diretriz 3: gerar os modelos de dependências estratégicas com foco nos atores envolvidos no processo.....	47
4.3.2 Passo 2: gerar os modelos de razões estratégicas SR.....	49
4.3.2.1 Passo 2 – Diretriz 4: gerar os modelos de razões estratégicas com foco nos atores envolvidos no processo.....	50
4.3.2.2 Passo 2 – Diretriz 5: detalhar razões estratégicas de recursos.....	53
4.3.3 Passo 3: listar as atividades.....	54
4.3.3.1 Passo 3 – Diretriz 6: aplicar o <i>template</i> para todas as atividades.....	55
4.3.4 Passo 4: estimar os recursos das atividades.....	57
4.3.4.1 Passo 4 – Diretriz 7: realizar estimativas de recursos necessários.....	59
4.3.5 Passo 5: sequenciar atividades.....	60
4.3.5.1 Passo 5 – Diretriz 8: sequenciar as atividades.....	60
4.3.6 Passo 6: estimar a duração das atividades.....	61
4.3.6.1 Passo 6 – Diretriz 9: realizar estimativa de duração das atividades do sistema.....	65
4.3.6.2 Passo 6 – Diretriz 10: realizar estimativa de duração das atividades do projeto.....	68
4.3.7 Passo 7: gerar o cronograma do projeto.....	67
4.3.7.1 Passo 7 – diretriz 11: Gerar os cronogramas.....	70
4.4 Considerações finais.....	72
5 Estudo de Caso	74

5.1 Contexto.....	74
5.2 Aplicação da Proposta.....	75
5.2.1 Passo 1: gerar os modelos de dependências estratégicas SD.....	75
5.2.2 Passo 2: gerar os modelos de razões estratégicas SR.....	77
5.2.3 Passo 3: listar as atividades.....	79
5.2.4 Passo 4: estimar os recursos das atividades.....	80
5.2.5 Passo 5: sequenciar as atividades.....	81
5.2.6 Passo 6: estimar a duração das atividades.....	81
5.2.7 Passo 7: gerar o cronograma do projeto.....	82
5.3 Considerações Finais.....	83
6 Considerações Finais	85
6.1 Vantagens do uso da proposta.....	85
6.1.1 Validação de requisitos.....	85
6.1.2 Base de conhecimento.....	86
6.1.3 Rastreabilidade.....	86
6.2 Contribuições futuras.....	86
6.2.1 Experimento.....	87
6.2.2 Ferramenta.....	87
6.3 Considerações finais.....	87
Glossário	89
Referências bibliográficas	90

Resumo

A gerência de tempo é umas das nove áreas do conhecimento na gerência de projetos, sendo de grande importância por definir e sequenciar atividades, definir estimar recursos e tempo, além da construção do cronograma. Apesar de sua importância, a gerência de tempo muitas vezes é negligenciada por gerentes de projeto que consideram tais tarefas muito onerosas.

Neste contexto, a modelagem organizacional pode ser de grande auxílio para o desenvolvimento de uma abordagem que considere motivações, intencionalidades e dependências entre atores estratégicos participantes do projeto, além de fornecer uma missão macro do projeto.

Neste trabalho a técnica de modelagem organizacional i* será integrada ao processo de gerência de tempo, a fim de reduzir a o número de documentos necessários para tal gerência, e também visando considerar os aspectos organizacionais que irão interferir no desenvolvimento do projeto.

Palavras-chave: gerência de tempo, modelagem organizacional, *framework* i*.

Capítulo 1

Introdução

Neste Capítulo será feita uma breve introdução ao contexto deste trabalho, com o intuito de relevar a importância do tema no âmbito do desenvolvimento de software e da gerência de projetos.

Na seção 1.1 serão apresentadas as motivações que levaram a apresentar uma proposta que auxilie a gerência de tempo em projetos. Na seção 1.2 a proposta será contextualizada, apresentando um relato sobre a área de gerência de projetos no âmbito da engenharia de software, bem como uma breve introdução sobre ferramentas de modelagem organizacional. Na seção 1.3 é feita uma breve introdução sobre a proposta de integrar a gerência de tempo, conforme proposta pelo PMBOK [1] com *framework* de modelagem organizacional *i**. Por fim na seção 1.4 fala-se sobre as contribuições esperadas com o presente trabalho.

1.1 Contexto

O desenvolvimento de software é uma atividade não trivial cuja complexidade aumenta em decorrência do porte do sistema a ser desenvolvido. Ao contrário de outras áreas nas quais é possível obter um resultado previsível seguindo métricas e diretrizes estabelecidas, na engenharia de software é difícil a replicação de um resultado, visto que o mesmo depende de múltiplas variáveis tais como equipe, prazo, orçamento e contexto. As dificuldades em definir passos e estabelecer diretrizes para o desenvolvimento de software torna necessária a aplicação de conceitos de engenharia.

O desenvolvimento de software objetiva a criação de softwares como produtos finais de um processo definido por fatores como os acima citados. Logo é de grande importância que

seja feito um projeto do sistema, visando obter qualidade e padronização no processo. O desenvolvimento de um sistema começa por um projeto bem definido. De acordo com o PMI [2], um projeto é um empreendimento com início e fim definidos, que faz uso de recursos, para atingir objetivos estabelecidos, considerando prazos e custos previamente estabelecidos.

No contexto de projetos de software, cabe destacar os princípios de uma boa gerência de projetos como proposta pelo *Project Management Book of Knowledge* PMBOK [1]. Neste documento são apresentadas nove áreas de conhecimento, sendo que cada uma abrange diferentes aspectos que devem ser considerados para planejar e executar um projeto com qualidade. As nove áreas são:

- Gerenciamento de integração do projeto
- Gerenciamento de escopo do projeto
- Gerenciamento de tempo do projeto
- Gerenciamento de custo do projeto
- Gerenciamento de qualidade do projeto
- Gerenciamento de recursos humanos do projeto
- Gerenciamento de comunicação do projeto
- Gerenciamento de riscos do projeto
- Gerenciamento de aquisição do projeto

Neste trabalho, contudo, será enfatizada a área de gerenciamento de tempo devido a sua importância em propor diretrizes que visem auxiliar em tarefas como definição de atividades, estimativa de recursos e de tempo e desenvolvimento do cronograma.

Segundo o PMBOK [1], o objetivo da gerência de tempo é descrever os processos requeridos para o término do projeto, garantindo que o mesmo cumpra com os prazos definidos em um cronograma de atividades.

Uma vez especificado o acordo para o desenvolvimento de um software, a gerência de tempo preocupa-se em cumprir os requisitos estipulados para o sistema dentro do prazo definido, levando em conta fatores como orçamento, prazos, manutenções e alterações no projeto a fim de obter um produto documentado e com qualidade.

Considerando estes fatores fundamentais a um projeto, é destacada a importância do uso de técnicas ou métodos para a gerência de tempo. Como a gerência de tempo não é uma

atividade de fácil execução, visto que há muitos fatores a se considerar, será aplicada a gerência de tempo, como proposta pelo PMBOK [1], utilizando ferramentas de modelagem organizacional a fim de criar uma abordagem mais objetiva e que considere motivações e intencionalidades entre os envolvidos no projeto.

Por meio da aplicação da gerência de tempo via modelagem organizacional, espera-se não só propor uma abordagem que considere fatores organizacionais no processo, mas também permita uma visão mais simplificada e objetiva do processo por meio de sua representação gráfica.

1.2 Motivações

Tendo o projeto do sistema em mãos, uma boa gerência de tempo se faz necessária para definir um cronograma de atividades, alocar recursos e tempo de modo a cumprir as atividades listadas no cronograma. Neste contexto evidenciamos a necessidade do uso de uma abordagem bem definida para um gerenciamento eficiente.

Muitas vezes práticas propostas pelo PMBOK [1] são negligenciadas por gerentes de projeto, que veem demasiado formalismo ou muita dificuldade em seguir as regras impostas para uma gerência eficiente. Dentre os principais fatores que motivam tal negligência estão a pressa em iniciar o desenvolvimento do projeto, a suposição de que todos os envolvidos no projeto já possuem pleno conhecimento dos requisitos e das atividades do projeto e a premissa de que o cronograma não demanda alterações após definido.

Observando estes problemas, foi vislumbrada a possibilidade de apontar soluções para os mesmos, norteando o processo de gerência de tempo a partir de modelos organizacionais que permitam modelar informações estratégicas para este processo.

A dificuldade na aplicação da gerência de tempo conforme proposta pelo PMBOK [1] foi um fator de grande relevância para a definição do tema deste trabalho, haja visto que por meio da modelagem organizacional o processo de gerência de tempo pode ser aplicado sobre uma visão macro do projeto.

Por meio da aplicação da gerência de tempo baseada na modelagem organizacional espera-se criar uma abordagem que considere intencionalidades, motivações e fatores internos da equipe como parte do processo de desenvolvimento do sistema.

O uso da modelagem organizacional pode ainda auxiliar no processo, pois por meio de uma abordagem centrada nas pessoas envolvidas no projeto podem-se identificar e estimar tarefas, recursos e objetivos com foco nas relações entre os envolvidos nas atividades.

1.3 Proposta

Propõe-se neste trabalho uma abordagem para auxiliar o processo tradicional de gerência de tempo, de modo que sejam considerados os fatores organizacionais. Propõe-se integrar a gerência de tempo conforme descrita pelo PMBOK [1] por meio de uma técnica de modelagem organizacional.

De acordo com Alencar [3] *apud* Pádua e Cazarini [4], a Modelagem organizacional auxilia no entendimento dos requisitos de gestão que interferirão no desenvolvimento de software, além de facilitar a identificação de alternativas viáveis para os vários processos da organização de modo que se faça um referencial competitivo para a tomada de decisões, a fim de fazer um melhor uso do potencial da organização.

Para realizar a modelagem organizacional foi escolhido o *framework* *i** (istar). De acordo com Yu [5] o *framework* *i** permite descrever aspectos de intencionalidade e motivações envolvendo atores em um ambiente organizacional.

Por meio do *framework* *i** e com base nos requisitos funcionais e não funcionais do sistema, pode-se modelar via diagramas as tarefas, os objetivos, os recursos, as dependências e as pessoas envolvidas no projeto, nos dando uma visão organizacional por meio dos diagramas de dependências e de razões estratégicas.

Os diagramas gerados pelo *framework* *i** permitem uma maior visão de todas as etapas do projeto. Considerando as práticas de gerência de tempo do PMBOK [1] aplicada aos modelos de razões e dependências estratégicas gerados pelo *framework* *i**, Espera-se apoiar o processo de realização de estimativas de tempo proposto no referido modelo.

Por meio da integração do *framework* i* com a gerência de tempo, é visada a criação uma abordagem que facilite a estimativa de tempo. Em um primeiro momento verifica-se que a aplicação dos conceitos do PMBOK [1] sobre a modelagem organizacional pode tornar mais fácil a identificação de sub-tarefas e dependências que afetariam as mesmas.

1.4 Contribuições Alcançadas

Com a conclusão deste trabalho conseguimos:

- Criar uma abordagem passível de ser aplicada em projetos de qualquer porte, para realização de estimativas de tempo com base em relações entre as partes envolvidas no projeto modeladas via o *framework* de modelagem organizacional i*.
- Propor um conjunto de diretrizes para auxiliar o processo de gerência de tempo considerando como base as intencionalidades modeladas graficamente via *framework* i*. Este processo permite que as atividades associadas a esta área de conhecimento do PMBOK[1], possam ser realizadas de forma mais clara, objetiva e considerando expectativas de atores estratégicos envolvidos.
- Permitir que os envolvidos no processo de gerência de tempo possam obter um entendimento das relações organizacionais importantes que podem afetar a identificação de atividades, sequência de realização das mesmas, estimativas de recursos humanos e computacionais, entre outros.
- Demonstrar a aplicação do método por meio de um estudo de caso.

Capítulo 2

Gerenciamento de Tempo

Nesta seção apresenta-se a gerência de tempo como proposta pelo PMBOK [1], a fim de explicar sua importância na execução de um projeto. Será abordado o tema com relação ao processo tradicional de gerência de projetos, deixando para os Capítulos seguintes a abordagem via modelagem organizacional. A seção 2.1 apresenta uma introdução à gerência de projetos, de forma mais abrangente. Na seção 2.2 será dado enfoque para a área da gerência de tempo dentre as nove áreas do conhecimento propostas pelo PMBOK [1], por ser o foco principal deste estudo. Por fim serão feitas as considerações finais na seção 2.3.

2.1 A Gerência de Projetos

A aplicação de conceitos e técnicas para gerência de projetos interfere de maneira positiva no desenvolvimento, na execução e na qualidade do processo de acordo com Patah [6]. Na área de desenvolvimento de software essa importância se torna evidente à medida que aumenta a complexidade do projeto.

A atividade de gerência de projetos no âmbito do desenvolvimento de software difere das outras áreas em três aspectos principais segundo Sommerville [7]:

- **O produto é intangível:** o fato de um sistema ser um produto não físico dificulta a tarefa de gerenciá-lo, o que torna ainda mais importante sua correta documentação, visto que é a única forma de manter um controle sobre o processo.

- **Não há processo de software padrão:** não existe um processo unificado para o desenvolvimento de software, haja visto que o mesmo será definido por seu escopo, custo, equipe, contexto, orçamento e diversas outras variáveis.
- **Grandes projetos de software são em geral únicos:** o número de variáveis que irá afetar o projeto é muito grande, o sistema pode ser dedicado, ou possuir dependência de hardware, softwares ou informações do cliente.

Segundo Vargas [8] “o gerenciamento de projetos se refere a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de satisfazer seus requisitos, e é realizado com o uso de processos tais como iniciar, planejar, controlar e encerrar”.

De acordo com o *Project Management Institute* - PMI [2], um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. Ou seja, é um processo composto por múltiplas tarefas definidas entre o início e o fim do projeto. Existem vários modelos propostos para a gerência de projeto: será dado enfoque ao processo de acordo com o PMI, apesar da existência de outros modelos, tais como os propostos pelo *software engineering institute* SEI [9], ou o *international Standards organization* ISO [10].

De acordo com Russo [11] o gerente de projetos é uma peça chave para o sucesso do mesmo. O gerente de projetos deverá ser alguém envolvido com o processo desde o início, sendo atribuída a responsabilidade pelo sucesso do projeto a sua capacidade de administração e gerência, muitas vezes não sendo necessário que o gerente de projetos seja um especialista no contexto em que o sistema irá abordar.

De acordo com Pressman [12], para que um projeto seja bem sucedido, é necessário analisar corretamente alguns parâmetros, tais como recursos necessários, o escopo do sistema, as atividades necessárias para sua conclusão, métricas empregadas no seu desenvolvimento, tempo estimado entre outros.

Dentre todos os parâmetros relativos à gerência de projetos dá-se enfoque para a gerência de tempo, devido a sua importância com relação às estimativas de desenvolvimento baseada em recursos e tarefas, a definição do cronograma e a alocação de recursos conforme demandarem as atividades.

Segundo o PMBOK [1], um projeto é composto por cinco fases: iniciação, planejamento, execução, monitoramento/controlado e encerramento. Cada uma das nove áreas está envolvida

em uma ou mais fases do projeto, sendo a gerência de tempo executada na fase de planejamento e no controle e monitoramento do projeto.

2.2 A Gerência de Tempo

Segundo Alencar, Castro [3], o sucesso da equipe de desenvolvimento está relacionada ao gerente de projeto, de modo que o desempenho do projeto está diretamente ligado ao desempenho do gerente, ressaltando dentre algumas das características desejáveis ao um bom gerente o espírito de liderança, motivação, a capacidade resolver conflitos e negociar.

Dentre as áreas propostas pelo PMBOK[1], a gerência de tempo abrange atividades pertinentes ao início, desenvolvimento e conclusão do projeto em uma etapa de planejamento, a qual visa fornecer um melhor conhecimento sobre o projeto para o gerente e a equipe.

De acordo com o PMBOK [1] a gerência do tempo de projeto é responsável por descrever os processos requeridos para o término do projeto, garantindo que o mesmo cumpra com os prazos definidos em um cronograma de atividades.

A gerência de tempo é dividida em seis passos, sendo os cinco primeiros executados durante a fase de planejamento de projeto e o último realizado durante a fase de monitoramento e controle. De acordo com o PMBOK [1] a gerência de tempo é constituída pelas seguintes atividades:

- Definição de atividades
- Sequenciamento das atividades
- Estimativa de recursos da atividade
- Estimativa de duração da atividade
- Desenvolvimento do cronograma
- Controle do cronograma.

Como proposto pelo PMBOK [1], há três tópicos consideráveis no contexto de execução de atividades, são eles:

- Entradas: deverão ser especificadas as atividades que serão realizadas, bem como devem ser relevadas suas documentações. Também devem ser definidas como entradas as estimativas de recursos.
- Ferramentas e técnicas: quais as técnicas e ferramentas necessárias para a execução e processamento dos dados de entrada.
- Saídas: resultado gerado após o término do processo pode ser um documento ou outro tipo de resposta resultante.

Para cada atividade da gerência de tempo, deve haver uma entrada, a aplicação de alguma técnica ou ferramenta e uma saída resultante do processo. Na próxima subseção será feita uma descrição de cada atividade, quais entradas ferramentas, técnicas e saídas são propostas pelo PMBOK [1]. A fim de esclarecer quais elementos são necessários como entradas, cada atividade será exemplificada no contexto de um projeto de um sistema para gerenciamento de cronotacógrafos. Este projeto será apresentado em detalhes no Capítulo 5. Basicamente envolve o cadastramento e manutenção de cronotacógrafos e selos para os mesmos, além de funcionalidades para geração de relatórios acerca de tais atividades. Um cronotacógrafo deve ser vinculado ao chassi de um veículo de carga ou transporte de passageiros, podendo ter um conjunto de selos contendo especificações do INMETRO associado.

2.2.1 Definir atividades

Em um primeiro momento, na fase de planejamento do projeto, deve-se identificar quais são as atividades necessárias para concluir as etapas do desenvolvimento do sistema, sejam versões ou o produto final. Pode ser necessário o desenvolvimento de centenas de atividades para conclusão de um projeto e listá-las de forma aleatória não é uma tarefa fácil tampouco eficiente.

2.2.1.1 Entradas para definir as atividades

A fim de definir as atividades necessárias à conclusão do projeto, como propõe o PMBOK [1], são necessários os seguintes documentos:

- **Linha de base do escopo do projeto:** a linha base possui três componentes, sendo a seguir descrito cada um deles:
 - **Declaração do escopo do projeto:** este documento abrange vários aspectos do sistema desde o fechamento do contrato de desenvolvimento, segundo o PMBOK [1] é composto:
 - **Critérios de aceitação do produto:** estabelecidos no contrato. **Exemplo:** o sistema deverá rodar na plataforma Windows 7, ser desenvolvido em java e usar banco de dados oracle.
 - **Entregas do projeto:** não apenas o produto final, mas os protótipos, versões e documentação devem estar estabelecidos. **Exemplo:** entregas de protótipos nos dias 25/07/12 e 25/08/12; entrega da versão final documentada dia 25/09/12.
 - **Exclusões do projeto:** define o que não será abrangido no projeto, pode conter funcionalidades ou partes do sistema que não sejam necessários ao cliente. **Exemplo:** por se tratar de uma expansão do software de gerenciamento de frotas, o sistema não conterà cadastro de chassis, pois o cliente já o possui.
 - **Restrições do projeto:** em geral as limitações do projeto já estão listadas no contrato de desenvolvimento, tais como datas de entrega, restrições de orçamento, porém as restrições não são definidas exclusivamente pelo cliente: podem existir limitações internas na empresa. **Exemplo:** poderão ser alocados 6 funcionários para o desenvolvimento do sistema.
 - **Premissas do projeto:** são as hipóteses relacionadas com o desenvolvimento do projeto, devem ser seguidas de seu impacto caso sejam verdadeiras ou não. Geralmente são levantadas e analisadas pela equipe de desenvolvimento. **Exemplo:** o desenvolvimento do sistema tem tolerância de atraso de uma semana.
 - **Estrutura analítica do projeto (EAP):** gerada a partir do documento de requisitos, do documento de escopo do projeto e opcionalmente pelos ativos de processos organizacionais tais como arquivos de projetos anteriores e políticas de procedimentos.

A EAP define o escopo do projeto e representa o trabalho como um todo organizado na forma de camada, cada qual composta por pacotes de atividades.

A EAP é organizada de maneira hierárquica e voltada para a entrega do projeto: as atividades serão sequenciadas na ordem em que devem ser executadas para a conclusão do projeto sendo que cada camada de um nível mais baixo possui detalhes das camadas de níveis mais altos. A Figura 2.1 exemplifica este cenário:



Figura 2.1: Exemplo de EAP para definir casos de teste.

- **Dicionário da EAP:** contem informações sobre a organização da EAP, como sua hierarquia, descrição e critérios de aprovação. Um exemplo de dicionário da EAP é exibido na Tabela 2.1:

ID	Pacote de trabalho	Descrição	Critério de aceitação
1	Definir casos de teste	Definir quais serão os testes aplicados sobre uma funcionalidade	Inexistência de erros ou situações não desejadas
1.1	Definir critérios de aprovação	Esclarecer quais os critérios para que a funcionalidade seja aprovada nos testes	A funcionalidade cumprir com os critérios definidos

1.2	Gerar <i>check-list</i>	Verificar o que já foi feito, o que está sendo feito e o que será feito	A funcionalidade deve ser gerida conforme critérios definidos
1.2.1	Mapear atividades	Identificar atividades e tarefas necessárias para execução dos testes	A funcionalidade executar todas as atividades e tarefas identificadas
1.2.2	Controlar atividades	Gerenciar as atividades mapeadas conforme necessário	A funcionalidade deve gerar apenas os resultados esperados ao termino do processo.
1.3	Revisar e formalizar documentação	A documentação gerada ou alterada pelos testadores deve ser revisada e formalizada	A documentação deve estar dentro dos padrões estabelecidos pela empresa

Tabela 2.1: Exemplo de dicionário de EAP para definir casos de teste.

- **Fatores ambientais da empresa:** fatores externos e internos que afetam direta ou indiretamente a empresa, tais como infraestrutura, cultura e estrutura da empresa normas e padrões da indústria ou do governo que tenham relação com a empresa. **Exemplo:** o gerente de projeto possui familiaridade com o problema.
- **Ativos de processos organizacionais:** quando houverem devem ser listados para a definição de atividades, pois podem ser relevantes ao processo. São exemplos de ativos organizacionais informações e conhecimentos inerentes à empresa, normas, políticas, processos e procedimentos para planejamento formal e informal das atividades. **Exemplo:** o projeto será gerenciado seguindo a metodologia SCRUM, com rápidas reuniões diárias no começo do expediente de trabalho, e uma reunião marcada para a equipe nas segundas-feiras.

2.2.1.2 Ferramentas e técnicas para definir as atividades

Tendo como entrada a estrutura analítica do projeto bem como seu dicionário e os ativos organizacionais seguem as técnicas e ferramentas para processar estas informações:

- **Decomposição:** tendo em mãos a EAP e o dicionário da EAP, agora cada pacote de atividades identificado anteriormente será subdividido em atividades menores e mais simples de serem gerenciadas.

Cada pacote de atividades irá gerar uma lista de atividades menores, as quais representam uma unidade de trabalho mais fácil de estimar e de se desenvolver. O objetivo nesta etapa é obter a lista de todas as atividades necessárias para conclusão do projeto organizadas de acordo com o pacote de tarefas que as geraram.

- **Planejamento em ondas sucessivas:** consiste em efetuar o planejamento gradativamente, à medida que o projeto for se desenvolvendo e novas informações estiverem disponíveis. Nesta fase é dada ênfase em realizar um planejamento mais completo para as etapas mais próximas do projeto e planejar em mais alto nível etapas que estão em uma parte mais avançada do ciclo de vida do sistema.
- **Modelos:** pode consistir de um padrão da própria empresa para a criação de listas de atividades ou um *framework* para tal tarefa. Pode ainda ser uma lista de um projeto anterior ou parte dela, que será usada como base para a criação da nova lista.
- **Opinião especializada:** outra técnica de definição de atividades consiste em deixar esta atividade para alguém com experiência e conhecimento em detalhamento de atividades, criação de EAPs e desenvolvimento de cronograma, podendo ser necessária a ajuda de alguém de fora da equipe.

2.2.1.3 Saídas da definição de atividades

Aplicando as técnicas sobre as entradas anteriormente descritas, deve-se segundo o PMBOK [1], obter as seguintes saídas:

- **Lista de atividades:** a lista consiste nas atividades obtidas dos pacotes de atividades passados como entrada e deve conter todas as atividades necessárias para conclusão do projeto e uma descrição detalhada atribuída a cada uma.
- **Atributos das atividades:** cada atividade além de sua descrição contendo o que deverá ser feito também possui outros atributos, tais como data limite para conclusão, lista de atividades das quais depende para poder ser concluída, restrições, atividades anteriores e subsequentes.
- **Lista de marcos:** um marco de controle pode ser um ponto de conclusão de alguma etapa do projeto, ou um ponto de importância relevante definido pela equipe. Deverá ser gerada uma lista contendo todos os marcos de controle e seu significado (reunião mensal, obrigatoriedade de atividade, revisão do projeto).

2.2.2 Sequenciar atividades

Uma vez obtida a lista de atividades, de acordo com o PMBOK [1] as mesmas devem ser colocadas na ordem em que serão desenvolvidas, considerando as dependências entre as tarefas, bem como fatores que possam limitar seu desenvolvimento. Todas as atividades anteriormente listadas devem possuir uma atividade anterior e uma posterior, com exceção da primeira e da última. Para sequenciar as atividades as seguintes entradas são necessárias:

2.2.2.1 Entradas para sequenciar as atividades

As seguintes entradas são necessárias para sequenciar as atividades:

- **Lista de atividades:** obtida anteriormente (ver seção 2.2.1.3). **Exemplo:**
 - Definir casos de teste: definir qual será a rotina de testes aplicada para cada funcionalidade.
 - Definir condições de aprovação: especificar o resultado esperado para aprovação da funcionalidade.
 - Gerar *check-list* de teste: listar o que já foi feito, controlar o que está sendo feito e definir o que será feito em termos de testes.

- Revisar e formalizar documentação: revisar, corrigir e padronizar (se necessário) a documentação escrita e repassada pelos testadores.
- Mapear atividades: identificar quais são as tarefas necessária para execução do teste (ao testar uma funcionalidade para atribuir um cronotacógrafo para um chassis, será necessário acessar a lista de chassis cadastrados).
- Controlar atividades: gerenciar as atividades mapeadas no item anterior.
- **Atributos das atividades:** obtidos anteriormente (ver seção 2.2.1.3). deve-se dar atenção a tal documento, pois é possível que haja dependências e limitações entre as atividades. **Exemplo:**
 - Definir casos de teste: data de entrega: 23/07/12; depende das atividades definir condições de aprovação, gerar *check-list* de teste, revisar e formalizar documentação, mapear atividades e controlar atividades para ser concluída.
 - Definir condições de aprovação: data de entrega: 23/07/12; deve ser a primeira atividade executada.
 - Gerar *check-list* de teste: data de entrega: 23/07/12; será a quarta atividade a ser executada; depende das atividades mapear atividades e controlar atividades para ser concluída.
 - Revisar e formalizar documentação: data de entrega 24/07/12; será a quinta atividade a ser executada;
 - Mapear atividades: data de entrega: 23/07/12; será a segunda atividade executada; para cada atividade mapeada deverá constar informação referente ao seu teste (para atribuir um cronotacógrafo a um chassis será necessário acessar seu cadastro; o cadastro de chassis já foi testado?).
 - Controlar atividades: data de entrega: 23/07/12; será a terceira atividade a ser executada; o tempo necessário para a conclusão de cada atividade mapeada deverá ser incluído na documentação da funcionalidade que está sendo testada.
- **Lista de marcos:** obtida anteriormente (ver seção 2.2.1.3). **Exemplo:** os marcos definidos foram: entregas das versões nos dias 25/07/12; 25/08/12; 25/09/12. Reunião com toda a equipe nos dias 10/07/12; 10/08/12; 10/09/12. Reunião com o cliente nos dias 01/08/12 e 01/09/12.
- **Declaração do escopo do projeto:** descrita anteriormente (ver seção 2.2.1.1).
- **Ativos de processos organizacionais:** descritos anteriormente (ver seção 2.2.1.1).

2.2.2.2 Ferramentas e técnicas para sequenciar as atividades

Tendo as entradas descritas as seguintes ferramentas ou técnicas poderão ser aplicadas:

- **Método do diagrama de precedência (MDP):** consiste em gerar um diagrama onde cada atividade será representada por um quadrado, que deverá ser ligada as atividades adjacentes de acordo com a relação lógica entre elas. Relação esta que pode ser de quatro tipos:
 - **Término para início (TI):** o início da atividade sucessora depende do termino da atividade predecessora.
 - **Término para término (TT):** o término da atividade sucessora depende do termino da atividade predecessora.
 - **Início para início (II):** o início da atividade sucessora depende do início da atividade predecessora.
 - **Início para término (IT):** o término da atividade sucessora depende do início da atividade predecessora.
- **Determinação de dependência:** existem três tipos de dependências que podem ocorrer no projeto e que irão afetar a sequência de realização das atividades. São elas:
 - **Dependências obrigatórias:** ou *hard logic*, são dependências contratuais ou naturais ao projeto que irão tornar obrigatória à execução das atividades em uma determinada ordem.
 - **Dependências arbitradas:** ou *soft logic*, são definidas a fim de gerar a melhor sequência para o desenvolvimento das atividades, geralmente são baseadas na experiência da equipe ou de um de seus integrantes.
 - **Dependências externas:** não são definidas pelo cliente ou pela equipe, mas sim por fatores externos que possam bloquear o desenvolvimento de uma parte do projeto.
- **Aplicação de antecipações ou espera:** pode ser interessante para o desenvolvimento do projeto que a execução de alguma atividade seja adiantada ou postergada.

Uma funcionalidade que usará um banco de dados pode ser desenvolvida antes de o banco ser concluído usando um banco de dados temporário, se a mesma funcionalidade

irá conter prioridades de acesso, talvez seja interessante adiar seu desenvolvimento até que a parte de acessos esteja completa.

- **Modelos de diagramas de rede de cronograma:** um sistema todo ou uma parte dele pode ser representado na forma de um diagrama a fim de facilitar a visualização de dependências entre atividades e facilitar seu sequenciamento.

2.2.2.3 Saídas do sequenciamento de atividades

Tendo aplicado os métodos anteriores, de acordo com o PMBOK [1] devem-se ter as seguintes saídas:

- **Diagramas de rede do cronograma do projeto:** o diagrama pode ser gerado manualmente ou com auxílio de ferramentas e deve conter a sequência de realização das atividades, podendo conter informações detalhadas sobre o processo.
- **Atualização dos documentos do projeto:** ao menos a lista de atividades, atributos das atividades e a lista de riscos devem ser atualizados.

2.2.3 Estimar os recursos das atividades

Tendo as atividades sequenciadas agora se devem estimar os recursos necessários à sua execução, recursos estes que podem ser de diversas naturezas: hardware, softwares ou ferramentas, pessoas e qualquer outra coisa que seja necessária.

2.2.3.1 Entradas para estimar os recursos das atividades

Para o processo de estimativa de recursos, segundo o PMBOK [1], as seguintes entradas são necessárias:

- **Lista de atividades:** obtida anteriormente (ver seção 2.2.1.3).
- **Atributos das atividades:** obtidos anteriormente (ver seção 2.2.1.3).

- **Calendário de recursos:** consiste de uma lista de todos os recursos que serão usados no decorrer do projeto, contendo informações como quantidade do recurso, quando e por quanto tempo ele estará disponível. Como dito anteriormente membros da equipe também são contados como recursos, e são classificados de acordo com sua experiência. **Exemplo:**
 - Testador: disponíveis 2 testadores durante todo o projeto, por 8 horas diariamente.
 - Analista de teste: disponível 1 durante todo o projeto, por horas diariamente.
 - Banco de dados para testes: disponível 1 durante toda a execução do projeto, por 8 horas diariamente de segunda-feira a quinta-feira, sendo disponibilizado apenas nas 4 primeiras horas do expediente de sexta-feira.
 - Analista de negócios: disponível 1 até o dia 25/08/12 durante 8 horas nas segundas-feiras.
- **Fatores ambientais da empresa:** uma lista de fatores que podem interferir de algum modo no desenvolvimento das atividades. **Exemplo:** os funcionários da empresa do cliente já possuem experiência com sistemas, o que irá reduzir o tempo necessário para treinamento.
- **Ativos de processos organizacionais:** em geral é composto por políticas e procedimentos da empresa acerca de locação e obtenção de recursos, além do histórico da empresa em projetos semelhantes. **Exemplo:** historicamente se sabe que a primeira etapa no desenvolvimento é a interface gráfica, por ser a parte visível ao cliente é o primeiro item que precisa de sua aprovação.

2.2.3.2 Ferramentas e técnicas para estimar os recursos das atividades

A partir das entradas passadas na etapa anterior segundo o PMBOK [1] as seguintes técnicas e ferramentas poderão ser usadas para as estimativas:

- **Opinião especializada:** caso não haja alguém na equipe com experiência e conhecimento na área, será necessária a ajuda de um consultor ou especialista no problema.

- **Análise das alternativas:** uma atividade geralmente pode ser executada de múltiplas maneiras, nesta etapa cabe à equipe analisar a melhor maneira de executá-la de forma a usar os recursos da melhor maneira possível.
- **Dados publicados para auxílio a estimativas:** são os dados e estatísticas da empresa sobre custo de recursos e atividades executadas
- **Estimativa *bottom-up*:** está técnica consiste em fragmentar atividades que tenham complexidade muito alta para que possam gerar uma estimativa realista, ou atividades relacionadas. A estimativa final é obtida por meio das estimativas para cada um dos passos necessários à conclusão da atividade.
- **Software de gerenciamento de projetos:** existem ferramentas que auxiliam no processo de estimativa de recursos, tendo como entrada as atividades e os recursos estas ferramentas são úteis, sobretudo para projetos de grande porte.

2.2.3.3 Saída da estimativa de recursos das atividades

As informações e documentos resultantes da estimativa de recursos, de acordo com o PMBOK [1] são:

- **Requisitos do recurso da atividade:** documento no qual são listados individualmente os recursos necessários para a execução de cada atividade, podendo ainda conter detalhes como restrições e limitações para o desenvolvimento.
- **Estrutura analítica dos recursos:** consiste em um documento contendo informações referentes aos recursos organizados conforme sua natureza. Um recurso do tipo funcionário conterá informações do tipo experiência e media de produtividade por exemplo.
- **Atualização dos documentos do projeto:** ao menos a lista de atividades, a lista de atributos de atividades e o calendário de recursos devem ser revisados ao final deste processo.

2.2.4 Estimar a duração das atividades

O processo da gerência de tempo até esta etapa trata do levantamento de informações acerca das atividades necessárias à conclusão do projeto e dos recursos necessários para a conclusão das atividades. Agora será realizada a estimativa com relação ao tempo que deverá ser reservado para cada atividade.

2.2.4.1 Entradas para estimar a duração das atividades

De acordo com o PMBOK [1] as seguintes informações e documentos se fazem necessários para o processo de estimar a duração das atividades:

- **Lista das atividades:** obtida anteriormente (ver seção 2.2.1.3).
- **Atributos das atividades:** obtidos anteriormente (ver seção 2.2.1.3).
- **Requisitos dos recursos da atividade:** obtidos anteriormente (ver seção 2.2.3.3).
- **Calendário dos recursos:** descrito anteriormente (ver seção 2.2.3.1).
- **Declaração do escopo do projeto:** descrito anteriormente (ver seção 2.2.2.1), mas aqui se deve levar em conta as restrições do projeto e suas premissas, tais como disponibilidade de informações nas etapas do projeto (calendário de recursos).
- **Fatores ambientais da empresa:** aqui são relevantes fatores como informações da empresa acerca de estimativas e produtividade dos funcionários. **Exemplo:** a empresa possui uma ferramenta para testes de interface.
- **Ativos de processos organizacionais:** consiste em informações da empresa acerca de projetos anteriores, regras para construção do cronograma e o calendário do projeto. **Exemplo:** Se já houve um projeto semelhante desenvolvido na empresa, a estimativa base adotada para as atividades pode ser obtida do mesmo.

2.2.4.2 Ferramentas e técnicas para estimar a duração das atividades

De acordo com o PMBOK [1] as seguintes técnicas e ferramentas poderão ser aplicadas a fim de estimar a duração das atividades:

- **Opinião especializada:** pode ser de alguém da equipe de projeto ou um consultar especializado na área do problema.
- **Estimativa análoga:** esta técnica consiste em fazer as estimativas do projeto baseado em algum projeto anterior de complexidade semelhante desenvolvido pela empresa. É uma técnica que depende de menos tempo para ser aplicada, porém não garante resultados muito precisos. Esta técnica é mais eficiente se aplicada em uma equipe com membros que trabalharam no projeto base, pois seria agregada a experiência na equipe de desenvolvimento.
- **Estimativa paramétrica:** técnica que consiste em usar dados relativos às atividades como parâmetros para definição das estimativas, dentre outros estes dados podem abranger complexidade, prazo, qualidade, esforço e necessidade de recursos. Dentre tais técnicas destacam-se COCOMO, pontos por caso de uso e pontos por função.
- **Estimativa de três pontos:** por essa técnica é realizado um calculo das estimativas de tempo para a atividade segundo três cenários prováveis:
 - **O melhor caso possível**
 - **O pior caso possível**
 - **O caso mais provável**

A partir destas três estimativas será efetuado um calculo considerando o tempo no melhor caso, mais o tempo no pior caso, mais quatro vezes o tempo no caso esperado, e então se divide o tempo obtido por seis. O resultado obtido será a estimativa de duração da atividade.

- **Análise das reservas:** será considerada a sobra de tempo no cronograma, a fim de adequá-la de acordo com a necessidade de mais ou menos horas para conclusão do projeto ou de uma de suas etapas.

2.2.4.3 Saídas da estimativa da duração das atividades

Como resultado da aplicação das técnicas e ferramentas descritas, segundo o PMBOK [1] as seguintes saídas são obtidas:

- **Estimativas da duração da atividade:** poderá ser descrita em horas ou dias de trabalho, podendo ainda conter uma margem de erro dada em dias, horas ou ainda na forma de probabilidade de ocorrência.
- **Atualização dos documentos do projeto:** ao menos as documentações de atributos das atividades e as premissas levantadas devem ser atualizadas.

2.2.5 Desenvolver o cronograma

Nesta etapa é gerado o cronograma do projeto com base em todas as informações acerca de atividades levantadas até agora, relevando o calendário de recursos, dependência entre tarefas, limitações, prioridades e etc.

O cronograma deve conter ainda uma margem de segurança estimada com relação as suas atividades e a probabilidade de atraso, para que haja certa tolerância a atrasos no processo, as etapas a seguir são definidas pelo PMBOK [1] para o desenvolvimento do cronograma.

2.2.5.1 Entradas para desenvolver o cronograma

As seguintes entradas são necessárias para gerar o cronograma:

- **Lista de atividades:** obtida anteriormente (ver seção 2.2.1.3).
- **Atributos das atividades:** obtidos anteriormente (ver seção 2.2.1.3).
- **Diagrama de rede do cronograma do projeto:** obtido anteriormente (ver seção 2.2.2.3).
- **Requisitos dos recursos da atividade:** obtidos anteriormente (ver seção 2.2.3.3).
- **Calendário dos recursos:** obtido anteriormente (ver seção 2.2.3.1).

- **Estimativas da duração da atividade:** obtidas anteriormente (ver seção 2.2.4.3).
- **Declaração de escopo do projeto:** descrito anteriormente (ver seção 2.2.2.1).
- **Fatores ambientais da empresa:** fatores como ferramentas usadas no desenvolvimento do cronograma. **Exemplo:** será feito o uso da ferramenta Microsoft Project para auxiliar no desenvolvimento do cronograma.
- **Ativos de processos organizacionais:** ativos que podem influenciar o desenvolvimento do cronograma. **Exemplo:** serão adicionados 10 dias como folga no prazo de desenvolvimento.

2.2.5.2 Ferramentas e técnica pra desenvolver o cronograma

A seguir as ferramentas e técnicas são propostas pelo PMBOK [1] para a criação do cronograma:

- **Análise da rede do cronograma:** o diagrama de rede do cronograma é analisado de acordo com alguma técnica a fim de gerar o cronograma do projeto. A seguir apresentarem-se algumas das técnicas que podem ser usadas nesta análise.
- **Método do caminho crítico:** busca um caminho no diagrama priorizando as atividades que não possuem flexibilidade na data de entrega ou possuem restrições.
- **Método da cadeia crítica:** semelhante ao método do caminho crítico, mas há a inserção de um marco, cuja duração é a variância de tempo na realização da atividade, ao fim de um caminho crítico. Para atividades dependentes do caminho crítico também são colocados marcos a fim de fazer o fluxo convergir para a cadeia crítica.
- **Nivelamento de recursos:** técnica aplicada a um diagrama já varrido pelo método do caminho crítico, com o intuito de fazer uma distribuição igualitária de recursos e garantir que recursos não estejam alocados a duas atividades diferentes simultaneamente.
- **Análise do cenário “E-se”:** esta técnica analisa os possíveis casos desencadeados por riscos ou por eventos que possam afetar o desenvolvimento do sistema. Este método pode ser usado para redefinir o cronograma do projeto a fim de minimizar os impactos causados pelo evento.
- **Antecipação e espera:** descrito anteriormente (ver seção 2.2.2.2).

- **Compressão do cronograma:** existem técnicas para encurtar o prazo de execução do projeto sem alterar seu escopo. A seguir listarem-se duas destas técnicas.
 - **Compressão:** consiste em alocar mais recursos conforme a demanda a fim de agilizar o termino da atividade.
 - **Paralelismo:** ao invés de executar as atividades em sequência, elas poderão ser executadas em paralelo, com exceção de atividades com dependências.
- **Ferramentas para o desenvolvimento do cronograma:** estas ferramentas geram o cronograma automaticamente, com base nas atividades, recursos, atributos de atividades, calendário de recursos e afins.

2.2.5.3 Saídas do desenvolvimento do cronograma

Segundo o PMBOK [1] os seguintes artigos são gerados como saída do desenvolvimento do cronograma:

- **Cronograma do projeto:** o cronograma pode ser representado de maneira tabular ou na forma de gráficos de marcos ou barras, ou ainda no formato de diagrama de rede.
- **Linha de base do cronograma:** documento contendo o cronograma base obtido por meio da análise do diagrama de rede.
- **Dados do cronograma:** contem as informações referentes ao cronograma, como a alocação dos recursos por períodos de tempo e caminhos alternativos de execução.
- **Atualização dos documentos de projeto:** ao menos a documentação referente aos requisitos dos recursos das atividades, os atributos das atividades, calendário e o levantamento de riscos devem ser revisados.

2.2.6 Controlar o cronograma

É a única etapa da gerência de tempo que não é realizada na fase de planejamento do projeto. É realizada durante a fase de execução do projeto, pois riscos ou eventos podem alterar a ordem com que as tarefas devem ser executadas.

2.2.6.1 Entradas para controlar o cronograma

De acordo com o PMBOK [1] os seguintes documentos se fazem necessários para o controle do cronograma:

- **Plano de gerenciamento do projeto:** define como o projeto será executado, monitorado e controlado. Este documento é obtido baseando-se na análise do termo de abertura do projeto, dos fatores ambientais da empresa, dos ativos organizacionais e de todos os processos de planejamento feitos até o momento. **Exemplo:** consiste de um documento contendo uma introdução ao projeto (nome, objetivos, justificativas, restrições, premissas, orçamento entre outros), e os planos de gerenciamento (gerenciamento de escopo, custo, tempo, qualidade riscos entre outros). Cada um destes itens contem diversos documentos, sendo que para cada plano de gerenciamento é feita uma análise dos mesmos. Abaixo listarem-se a constituição de alguns dos planos de gerenciamento do projeto:
 - Gerenciamento de qualidade: objetivos, processos e metodologias a serem utilizados, indicadores de desempenho, responsáveis pelas medições, métodos de análise das medições e ações decorrentes.
 - Gerenciamento de custos: orçamento do projeto, moeda, reservas de contingência, reservas gerenciais e técnicas para controle de custos.
- **Cronograma do projeto:** a ultima versão, contendo todas as modificações anteriores. Um exemplo de cronograma é apresentado na Figura 2.2:

Nome	Duração	Início	Término
<input checked="" type="checkbox"/> Definir casos de teste	2 dias	16/07/12 08:00	17/07/12 17:00
Definir condições de aprov	0,25 dias	16/07/12 08:00	16/07/12 10:00
<input checked="" type="checkbox"/> Gerar check-list de tes	0,75 dias	16/07/12 10:00	16/07/12 17:00
Mapear atividades	0,25 dias	16/07/12 10:00	16/07/12 13:00
Controlar atividades	0,5 dias	16/07/12 13:00	16/07/12 17:00
Revisar e formalizar docum	1 dia	17/07/12 08:00	17/07/12 17:00

Figura 2.2: Exemplo de cronograma.

- **Informações sobre o desempenho do trabalho:** como o desenvolvimento está progredindo. **Exemplo:** o prazo médio estimado para conclusão das atividades está com atraso médio de 2 dias.
- **Ativos de processos organizacionais:** os ativos referentes ao controle de cronograma. **Exemplo:** o cronograma é atualizado sempre nas terças-feiras e nas quintas-feiras durante o SCRUM diário.

2.2.6.2 Ferramentas e técnicas para controlar o cronograma

Segundo o PMBOK [1] o controle do cronograma pode ser feito com o uso das seguintes técnicas e ferramentas:

- **Análise de desempenho:** deve ser realizada a análise no cronograma a fim de monitorar o progresso do projeto considerando a importância das tarefas no contexto do projeto.
O impacto do atraso na conclusão de uma tarefa varia conforme a criticidade da mesma com relação a dependências de tarefas. A análise deve considerar dependências e prioridades entre tarefas.
A análise pode ser realizada por meio de técnicas como o gerenciamento de valor agregado (GVA) considerando a variação de prazos (VP), que consiste no orçamento agregado ao desenvolvimento de uma tarefa menos seu orçamento estimado. Outra medida de desempenho é o índice de desempenho de prazos (IDP), que é a razão entre o orçamento agregado ao desenvolvimento e o orçamento estimado.
- **Análise de variação:** são usadas as medidas de desempenho VP e IDP descritas anteriormente para avaliar a variação que houve com relação à linha base do projeto (ver seção 2.2.5.3) e a folga no cronograma.
- **Software de gerência de projetos:** sistemas que por meio de um cronograma auxiliam no desenvolvimento do projeto.
- **Nivelamento de recursos:** descrito anteriormente (ver seção 2.2.5.2).
- **Análise do cenário “E-se”:** descrito anteriormente (ver seção 2.2.5.2).

- **Ajuste de antecipações e esperas:** realizar o ajuste de acordo com o progresso do projeto (ver seção 2.2.5.2).
- **Compressão do cronograma:** descrito anteriormente (ver seção 2.2.5.2).
- **Ferramentas para desenvolvimento de cronograma:** usadas para redefinir o cronograma e avaliar o progresso do projeto.

2.2.6.3 Saídas do controle do cronograma

Por fim as saídas esperadas no processo de controle de cronograma, de acordo com o PMBOK [1] são:

- **Medição de desempenho do trabalho:** os valores de VP e IDP (ver seção 2.2.6.2) são adicionados à documentação da tarefa avaliada.
- **Atualizações de ativos de processos organizacionais:** ao menos as causas da diferença, as ações corretivas tomadas e os detalhes acerca dos atrasos devem ser documentados.
- **Solicitações de mudanças:** mudanças previstas para o cronograma podem acarretar em mudanças na linha de base do cronograma.
- **Atualização do plano de gerenciamento de projeto:** ao menos a linha de base do cronograma, o plano de gerenciamento do cronograma e a linha de base dos custos deverão ser atualizados.
- **Atualização dos documentos do projeto:** deverão ser atualizados o cronograma do projeto e os dados referentes a ele.

2.3 Considerações Finais

Neste Capítulo inicialmente foi feita uma breve introdução sobre a gerência de projetos e sua importância no âmbito do desenvolvimento de software. Na sequência explicamos em detalhes a área de gerência de tempo e todas as suas etapas, conforme proposto pelo PMBOK [1].

Explicou-se de forma detalhada as seis atividades que compõe o gerenciamento de tempo: definição e sequenciamento de atividades, estimativa de recursos e de tempo, e a criação e controle do cronograma. Para cada um dos seis passos são necessários documentos ou informações que servirão de entrada e na sequência as informações serão analisadas ou processadas por técnicas ou ferramentas para então gerar a documentação de saída.

A proposta abrange cinco das seis etapas propostas pelo PMBOK, sendo elas a identificação e sequenciamento das atividades, estimativa de recursos e tempo e a geração do cronograma.

Por meio desta análise constatamos que a tarefa de gerenciar tempo é de grande importância para a qualidade do processo, porém de alta complexidade. A gerência de tempo apoiada por uma técnica de modelagem organizacional poderia simplificar o processo.

Capítulo 3

O *Framework* i*

Neste Capítulo será apresentada a modelagem organizacional e sua aplicação na área de gerência de projetos de sistemas para abstração de estruturas organizacionais. Apresenta-se o *framework* i* (istar), que servirá de base para a proposta deste trabalho, será feito um detalhamento e exemplificação de seu uso a fim de fornecer um melhor entendimento da ferramenta.

Na seção 3.1 fala-se sobre modelagem organizacional e sua importância como técnica para compreensão de ambientes organizacionais, destacando sua aplicação no auxílio a gerência de projetos. Na seção 3.2 apresenta-se o *framework* de modelagem i*, explicam-se seus conceitos e sua aplicação sobre equipes de desenvolvimento de software. Na seção 3.3 são apresentadas as considerações finais deste Capítulo.

3.1 Modelagem Organizacional

De acordo com Vargas [8], a ideia de um projeto é a não repetição, tal ideia deve-se dentre outros fatores a equipe, ao ambiente organizacional e ao próprio gerente. O sucesso de um projeto está diretamente ligado a compreensão e a um gerenciamento eficiente de tais fatores.

gerência de projetos é de grande importância para obtenção de qualidade no processo de desenvolvimento de software, mas para que a mesma tenha maior eficácia quando aplicada no desenvolvimento comercial os fatores relativos ao âmbito organizacional devem ser relevados.

Um sistema é desenvolvido para um cliente com o propósito de atender suas necessidades, porém de acordo com Alencar, Castro [3], a maior dificuldade em se realizar uma engenharia de requisitos eficiente é a captura correta e completa dos requisitos organizacionais, funcionais e não funcionais de um sistema, a fim de obter um maior entendimento do problema e do ambiente no qual ele está inserido.

Segundo Alencar, Castro [3], a modelagem organizacional não ajuda a solucionar os problemas da organização, não valoriza o negócio e muitas vezes apenas automatiza processos manuais sem modificá-los. Mas ela é uma importante técnica por fornecer o entendimento e a compreensão do ambiente organizacional, contextualizando o ambiente em que os projetos serão desenvolvidos, e possibilitando a identificação de elementos que irão afeta-lo.

A modelagem organizacional pode tornar a gerência de projetos mais precisa, pois “ela é um passo importante para se chegar a um conhecimento estruturado do negócio, discutir mudanças, estruturar as regras do negócio e definir os atores (pessoas envolvidas no projeto) e recursos envolvidos em cada processo do negócio.”, Pádua, Cazarini e Inamasu [13].

De acordo com Grandó [14], a modelagem organizacional permite uma melhor identificação dos requisitos organizacionais que interferirão no sistema, além de auxiliar a tarefa de identificar alternativas para os vários processos da organização, o que facilita as atividades durante o desenvolvimento do sistema computacional, além de permitir que a análise seja mais bem integrada aos processos de desenvolvimento do sistema.

Sendo assim é de grande interesse que o uso de técnicas que permitam obter o conhecimento da estrutura organizacional possa ser aplicado também para equipes de desenvolvimento, a fim de permitir uma melhor gerência com base nos objetivos definidos para cada membro da equipe, disponibilidade de recursos, fatores ambientais da empresa entre outros.

O uso da modelagem organizacional objetiva, segundo Alencar, Castro [3]:

- Fornecer um objeto que possa ser compartilhado e reutilizado no fornecimento de informação e conhecimento.
- “Suportar tarefas da cadeia de fornecimento, pela habilitação de respostas a questionamentos, que não estão explicitamente representados no modelo”.

- Definir os objetos de modo que os mesmos possam ser facilmente interpretados e aplicados pelos usuários.
- Permitir uma visualização simplificada e de fácil compreensão do modelo.

Visando estes objetivos e a possibilidade de sua aplicação no contexto de gerência de projetos, o *framework i** foi escolhido como ferramenta de modelagem organizacional.

3.2 *Framework i**

Proposto por Yu [5] o *framework i** é uma ferramenta que permite representar relacionamentos e intencionalidades entre atores envolvidos no projeto, possibilitando uma visão geral do sistema e das pessoas envolvidas no seu desenvolvimento.

Dentre várias ferramentas para modelagem organizacional, tais como o modelo EKD (*enterprise knowledge development*) ou modelo proposto por Bukenko e Kirikova [15]. Neste trabalho será usado o *framework i** por diversas razões, dentre elas a facilidade de representar relações entre atores, à possibilidade de expressar requisitos funcionais e não funcionais facilitando inclusive sua identificação e a possibilidade de representar intencionalidades e motivações.

O *framework i** pode ser usado para uma melhor compreensão da organização, pois “a técnica *i** propicia esta compreensão por meio de mecanismos que permitem expressar tarefas, objetivos, objetivos-soft e recursos associados com intenções e necessidades de atores no ambiente organizacional” Santander [16]. Isto a torna uma técnica interessante do ponto de vista da engenharia de requisitos por possibilitar uma visão ampla do projeto a todos os envolvidos e no contexto da gerência de projetos por fornecer uma visão detalhada do projeto em termos de atores e objetivos.

O *framework i** possui dois modelos de diagramas com diferentes níveis de detalhamentos: o diagrama de dependências estratégicas e razões estratégicas. Os componentes principais de ambos os modelos são detalhados na Figura 3.1.

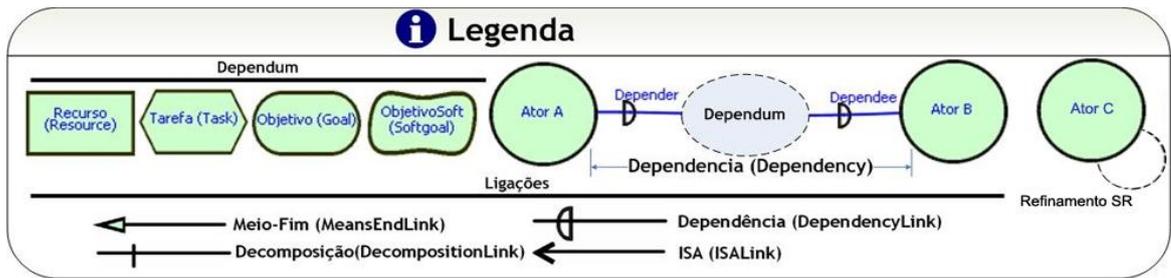


Figura 3.1: Legenda dos principais componentes do *framework i** (fonte: Brischke [17]).

3.2.1 Modelo de Dependências Estratégicas (SR)

Segundo Alencar, Castro [3], O modelo *Strategic Rationale* (SR) representa as intencionalidades, motivações e dependências entre os diversos atores integrantes do projeto, descritas por meio de nós e ligações, sendo uma relação constituída basicamente da seguinte forma, como representada na Figura 3.1:

- **Depender:** é representado pelo ator A, se relaciona com o ator B (*dependee*) a fim de satisfazer uma dependência.
- **Dependee:** ainda no contexto da Figura 3.1 o ator B será o *dependee*, pois ele será o responsável por realizar ou fornecer algo ao depender (ator A).
- **Dependum:** o *dependum*, fator de dependência entre dois atores, pode ser de quatro tipos conforme descrito na Figura 3.1:
 - Recurso: pode ser uma informação, parte de um sistema ou subsistema, um documento, entre outros.
 - Tarefa: atividade da qual o *dependee* precisa do *dependee* para conclusão.
 - Objetivo: tarefa ou atividade de maior importância para o projeto.
 - Objetivo-soft: meta ou intencionalidade objetivada pelo *dependee*, a qual o *dependee* deve satisfazer.

Todos os três elementos são conectados pelas ligações do tipo dependência, como descritas na Figura 3.1.

A fim de tornar mais claro os elementos descritos acima, demonstra-se sua aplicação por meio de um exemplo:

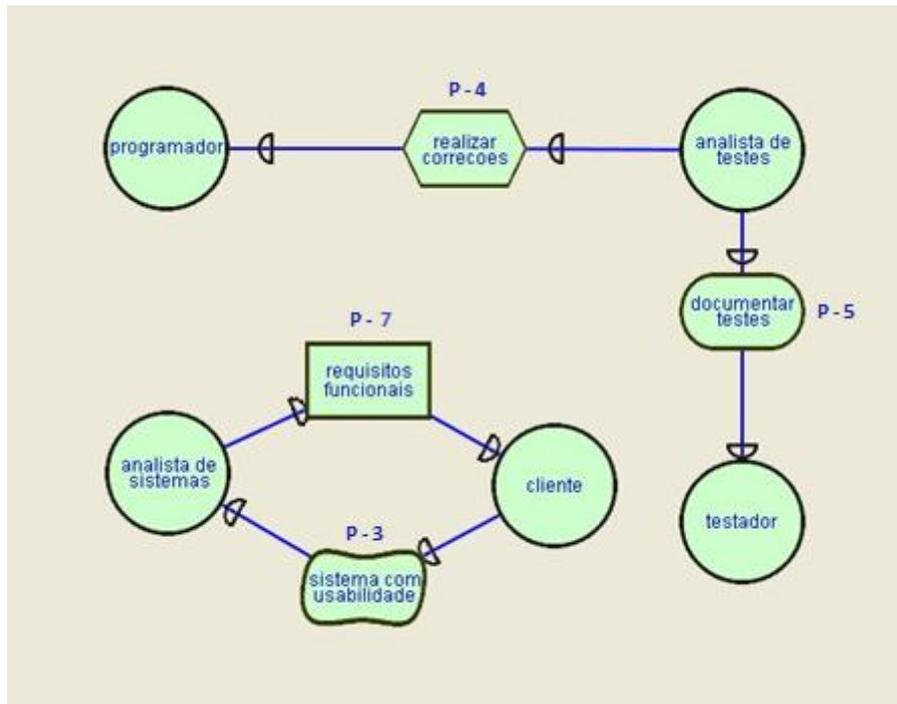


Figura 3.2: Exemplo de diagrama SD.

Conforme apresentado na Figura 3.2, identificam-se as seguintes relações:

- O analista de teste (*dependor*) depende do testador (*dependee*) para que o objetivo “documentar testes” (*dependum*) seja atingido.
- O analista de teste (*dependor*) depende do programador (*dependee*) para que a tarefa “realizar correções” (*dependum*) seja realizada.
- O analista de sistemas (*dependor*) depende do cliente (*dependee*) para obter o recurso "requisitos funcionais" (*dependum*).
- O cliente (*dependor*) depende do analista de sistemas (*dependee*) para que o seu desejo de um “sistema com usabilidade” (*dependum*) seja alcançado.

O modelo SR pode ainda possuir associações entre atores, sendo possíveis os seis tipos de relações representados na Figura 3.3:

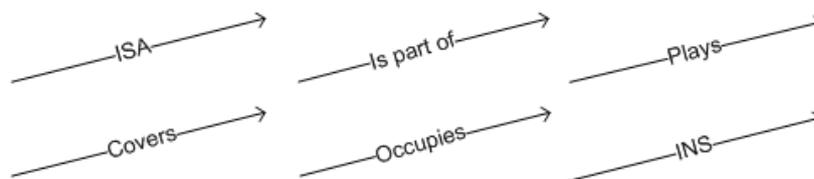


Figura 3.3: Possíveis associações entre atores (fonte: i*wiki [18]).

- **ISA:** quando um ator é uma classe especializada de outro ator. **Exemplo:** um gerente de desenvolvimento também é um programador.
- **Is part of:** descrevem partes de agentes, papéis ou posições. **Exemplo:** programar é papel de um programador (posição).
- **Play:** usado entre uma agente e um papel, para representar qual agente desempenhará o papel. **Exemplo:** um programador ocupa o papel de relator de erros.
- **Cover:** usado para indicar qual posição cobre um papel. **Exemplo:** a posição analista de sistema cobre o papel de delimitar escopo.
- **Occupies:** ligação entre agente e posição, no sentido de indicar qual agente ocupa qual posição. **Exemplo:** um agente desenvolvedor ocupa a posição gerência de projeto.
- **INS:** *instantiation* indica uma instância de uma entidade. **Exemplo:** um agente é uma instância de outro agente.

De acordo com Yu [5] estes elementos são suficientes para gerar uma representação de um modelo de dependências estratégicas (SD), sendo que todos os elementos aqui descritos também fazem parte do modelo de razões estratégicas (SR), o qual adiciona outros elementos.

3.2.2 Modelo de Razões Estratégicas (SR)

Em complemento ao modelo SD, o modelo SR visa detalhar tarefas, objetivos e atividades decompondo-as de modo que sejam identificadas as etapas necessárias para satisfação das dependências.

Como descrito acima, o modelo SR possui todos os elementos do modelo SD além de seus próprios elementos, sendo que os mesmos como proposto por Yu [5], serão detalhados a seguir:

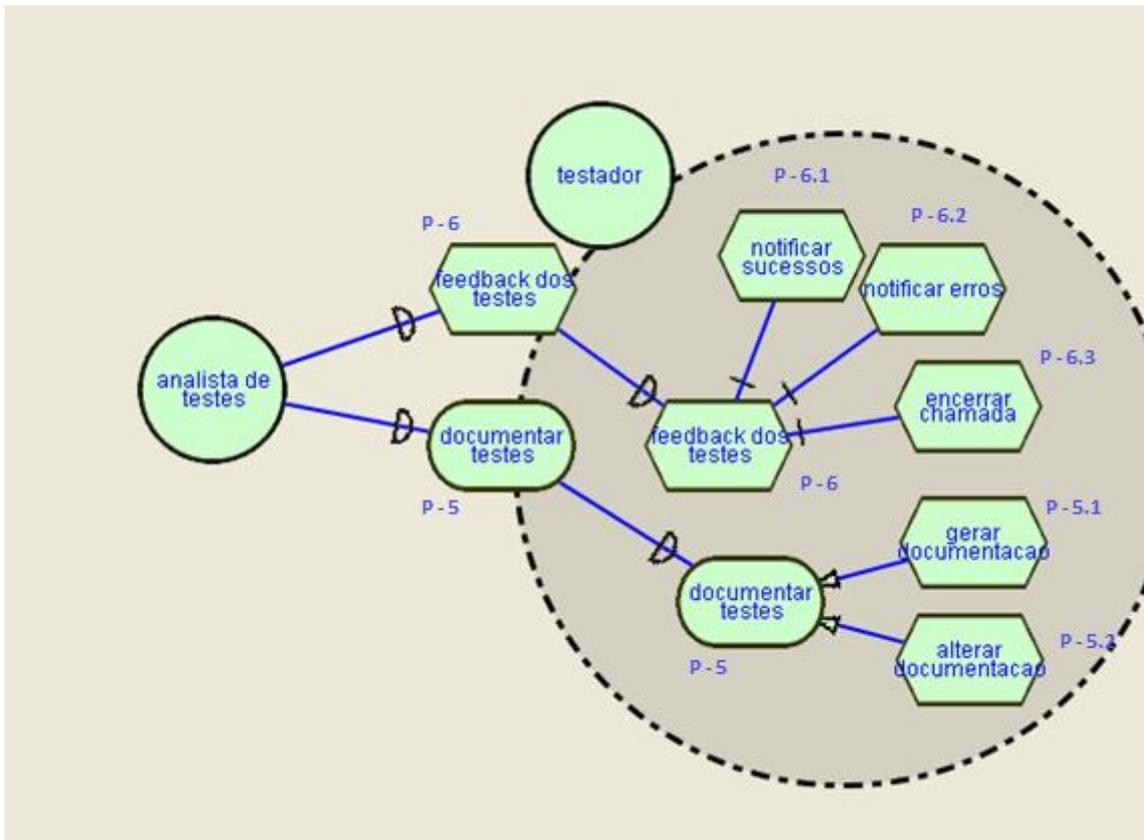


Figura 3.4: Exemplo de diagrama SR.

A área em cinza escuro delimitada por uma linha pontilhada na Figura 3.4 corresponde ao escopo do ator testador, ou seja, cabe ao testador realizar todas as atividades e satisfazer todas as dependências contidas no seu escopo. Pode-se observar na Figura 3.4 que na divisão e detalhamento das atividades surgem dois novos tipos de ligações. Segundo Yu [5], estas são:

- **Ligações de decomposição:** fazem o papel de ligação do tipo AND, pois para a conclusão de uma tarefa dividida em várias etapas, podendo estas etapas serem do tipo tarefa, objetivo, objetivo-soft ou recursos; a conclusão de todas as sub-tarefas é necessária para a conclusão da tarefa inicial. Ligações de decomposição podem dividir uma tarefa. **Exemplo:** na Figura 3.4, para se realizar a tarefa “feedback dos testes”, será necessário encerrar a chamada de teste, notificar os erros e notificar os sucessos.
- **Ligações meio-fim:** representam ligações do tipo OR, usada quando uma tarefa pode ser completada de formas distintas. **Exemplo:** o objetivo “documentar testes” na Figura

3.4 pode ser atingido alterando a documentação, caso ela já exista, ou criando a documentação no caso de uma nova funcionalidade estar sendo testada.

O modelo de razões estratégicas apresenta ainda ligações de contribuição, que são usadas de modo aditivo aos diagramas para expressar informações, podendo uma ligação de contribuição ser de um dos seguintes tipos apresentado na Figura 3.5.

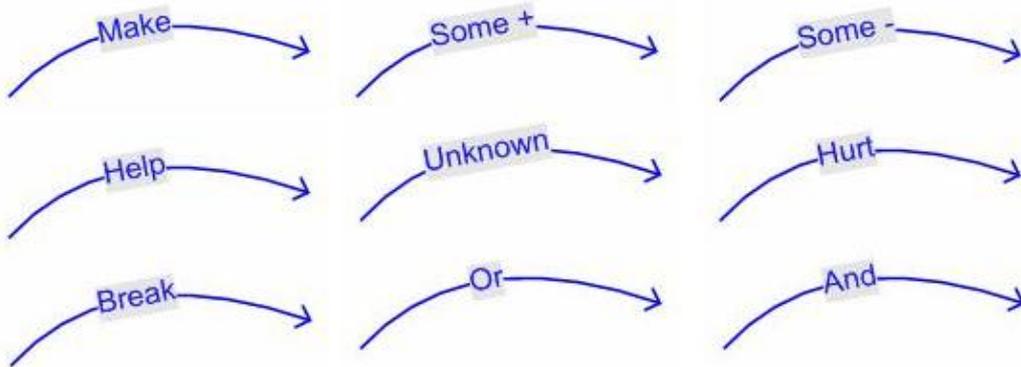


Figura 3.5: Possíveis ligações de contribuição (fonte: i*wiki [18]).

Links de contribuição podem ser de nove tipos. A seguir serão detalhados e exemplificados os links da Figura 3.5, sendo que o *dependum* envolvido com o link será representado entre aspas. Como descritos por Yu[5] os links de contribuição são:

- **Make:** contribuição forte o suficiente para satisfazer um objetivo. **Exemplo:** ao realizar a tarefa “implementar interface”, pode-se atingir como consequência o objetivo “entregar nova versão”.
- **Help:** ajuda na conclusão de uma tarefa ou objetivo, mas não ajuda o suficiente para realiza-lo prontamente. **Exemplo:** a tarefa “estimar custos” auxilia no objetivo “gerar orçamento”.
- **Some +:** contribui demais que o **help**, mas menos que o **make**. **Exemplo:** a tarefa “gerar estimativas” contribui mais que apenas definir custos, mas ainda não é o suficiente para cumprir o objetivo “gerar orçamento”.
- **Break:** contribui de forma negativa anulando automaticamente outro objetivo-soft. **Exemplo:** se a tarefa “executar testes” não for desenvolvida, ela irá anular o objetivo “documentar testes” já que não haverão resultados a serem documentados.

- **Hurt**: de maneira inversa ao **help**, irá contribuir de maneira negativa para a conclusão de um objetivo-soft. **Exemplo**: se “entregar versões” sofrer atraso, isso terá um impacto negativo no objetivo-soft “cliente satisfeito”.
- **Some -**: dificulta na conclusão de alguma tarefa de maneira inversa ao **some +**. **Exemplo**: “revisar documentação” é uma tarefa que contribui de maneira negativa para a “entrega da versão”, pois irá atrasá-la.
- **Unknow**: contribuição de peso desconhecido para a conclusão de um objetivo-soft. **Exemplo**: “testar funcionalidade” terá um impacto desconhecido sobre o objetivo “entrega da versão”.
- **Or**: indica que uma tarefa pode ser concluída se ela for satisfeita, ou se a tarefa indicada pela contribuição **or** for satisfeita. **Exemplo**: a tarefa “encerrar chamada” indica que a chamada para teste pode ser encerrada. Isso pode ocorrer se a funcionalidade for aprovada ou se ela voltar ao desenvolvimento (retornará para teste uma nova chamada).
- **And**: ao contrario do **or**, além da tarefa em si, a tarefa indicada pelo link de contribuição também precisa ser satisfeita. **Exemplo**: para “adicionar funcionalidade ao sistema”. Além de cumprir o objetivo “funcionalidade aprovada”, ainda é necessário o recurso “aprovação do cliente” antes que a mesma possa ser incluída no próximo *release* do sistema.

3.3 Considerações Finais

Neste Capítulo foi realizada uma introdução à modelagem organizacional, explicando sua importância na área de gerencia de projetos e o seu auxílio na engenharia de requisitos. Também foi feita uma introdução ao *framework* de modelagem organizacional i*.

Foi apresentado o modelo de dependências estratégicas (SD), um diagrama que permite representar dependências, intencionalidades e motivações entre os atores envolvidos no projeto.

Também foi apresentado o modelo de razões estratégicas (SR), o qual permite um maior detalhamento de tarefas e objetivos, permitindo dividi-los em elementos de menor complexidade, a fim de atingir o máximo grau possível de simplificação das tarefas.

O *framework* i* permite uma visão geral do projeto por meio dos diagramas, o que os torna artefatos interessantes para serem usados no gerenciamento do projeto.

No próximo Capítulo apresentamos uma proposta para simplificar e tornar menos onerosa a gerência de tempo fazendo-se uso da modelagem organizacional via *framework* i*.

Capítulo 4

Integrando o *framework* i* ao processo de gerência de tempo

Neste Capítulo será apresentada a proposta para integrar o *framework* de modelagem organizacional i* ao processo de gerência de tempo conforme descrito pelo PMBOK [1], visando criar uma abordagem que auxilie no processo de gerência de projetos, com foco na área de gerência de tempo.

Na seção 4.1 é feita uma introdução à proposta e ao contexto que a mesma visa auxiliar, de modo a destacar os principais pontos no qual a nova abordagem se difere da tradicional. Na seção 4.2 apresenta-se a proposta de como integrar a gerência de tempo proposta pelo PMBOK ao *framework* i*. Por fim realizam-se as considerações finais a respeito da proposta na seção 4.3.

4.1 Introdução

A gerência de projetos é uma atividade de grande importância no contexto de desenvolvimento de softwares por formalizar o processo, realizando uma análise detalhada e documentação de todo o mesmo, estabelecendo diretrizes e passos de modo a guiar o desenvolvimento do projeto a fim de obter um produto de qualidade.

Apesar de sua importância, muitos gerentes de projeto não aplicam o modelo formal proposto em decorrência da dificuldade do mesmo em gerar uma visão do projeto que considere fatores estratégicos relacionados aos atores participantes do mesmo.

Existem diversos fatores cuja consideração seria desejável para uma abordagem mais eficiente de gerenciamento, tais como as informações relativas a equipe de desenvolvimento e aos clientes e demais pessoas envolvidas no projeto, bem como explorar, mais especificamente as dependências entre os mesmos que afetam estrategicamente o planejamento e a execução do projeto. Estas dependências podem ser do tipo tarefa, recurso, objetivos e objetivo-soft entre outros, visando descrever de forma mais completa as categorias de dependências existentes.

Neste contexto é proposta uma abordagem que não apenas vise considerar os aspectos organizacionais e as dependências entre os atores, como também proporcione uma visão de todo o processo, por meio de modelos e diagramas.

Usar a modelagem organizacional como base para um processo de gerência pode auxiliar a equipe não só no controle do processo, mas também auxiliar no processo de engenharia de requisitos, além de fornecer uma visão ampliada da organização relevando características da mesma.

A proposta é apresentada considerando como base o processo proposto pelo PMBOK[1], o *framework* i* e o uso de técnicas para estimativas e análise, Visando a criação de um cronograma de projeto por este ser, em última instância o documento obtido a final do processo de gerência de tempo durante a etapa de planejamento.

Espera-se a partir desta abordagem, auxiliar no processo de definição e sequenciamento de atividades, estimativas de recursos e tempo para atividades e na criação do cronograma. Também se destacam outras vantagens advindas deste processo de integração, tais como a viabilidade de aplicação a projetos de qualquer porte, a possibilidade de considerar motivações e intencionalidades como fatores importantes no processo, além de permitir uma visão macro do sistema, a fim de possibilitar aos envolvidos no projeto um maior entendimento das relações organizacionais.

Com relação a todo o processo de gerência, os modelos gerados no *framework* i* não apenas irão auxiliar no processo de planejamento, como constituirão parte da documentação e das informações referentes ao projeto. Os mesmos ainda poderão auxiliar no processo de controle de cronograma, durante o a etapa de monitoramento e controle do projeto.

Na próxima seção será apresentada e exemplificada a proposta com o mesmo estudo de caso apresentado brevemente no Capítulo 2. Cabe salientar que este estudo de caso é

apresentado parcialmente apenas para fins de permitir um melhor entendimento da proposta. O estudo de caso completo será realizado no Capítulo 5.

4.2 As Diretrizes

Com o objetivo de definir passos para a aplicação da proposta, são apresentadas diretrizes baseadas no processo descrito no guia do PMBOK [1] para gerência de tempo em projetos, processo este realizado visando considerar as informações advindas da modelagem organizacional, por meio dos modelos SD e SR obtidos com o *framework* i*.

Dentre os motivos de centralizar o processo na modelagem organizacional destacam-se não só a possibilidade de considerar aspectos, intencionalidades e motivações entre os atores estratégicos envolvidos, mas também a capacidade do *framework* i* para representação de tais dados. Tal capacidade permite não só identificar os elementos do projeto de maneira bastante intuitiva, como posteriormente possibilita sequenciar e estimar custos sobre tal representação visual.

Como nem sempre se pode dispor da documentação completa do projeto, as diretrizes são propostas considerando procedimentos ou meios de realizar o gerenciamento de tempo em um projeto, em contextos onde exista ou não a documentação.

Cada diretriz está vinculada direta ou indiretamente a uma das etapas citadas no guia do PMBOK [1], podendo estar associada à execução de uma atividade ou obtenção de dados para tal, de modo que sua execução sequencial resulte no cronograma do projeto.

4.3 A Proposta

O objetivo da presente seção é apresentar a proposta de integração da gerência de tempo por meio da modelagem organizacional via *framework* i*. Apresenta-se a seguir um conjunto de passos a serem seguidos a fim de realizar a referida integração.

Para cada passo proposto, serão especificadas quais as entradas necessárias para sua execução (documentos, informações e artefatos do projeto), quais as diretrizes a serem seguidas e quais as saídas geradas.

Os passos serão exemplificados com base no projeto que será realizado no Capítulo 5. O sistema basicamente trata do cadastro e manutenção de cronotacógrafos e dos selos aplicados aos mesmos, selos estes que representam certificações de institutos tais como o INMETRO e o ISO.

O sistema deve possibilitar cadastrar cronotacógrafos nos chassi, e selos nos cronotacógrafos, possibilitando ainda a geração de relatórios das tarefas realizadas. Cabe ressaltar que este estudo de caso considera uma equipe de desenvolvimento de uma empresa de pequeno/médio porte.

A proposta será apresentada em passos, cada um podendo conter uma ou mais diretrizes. No passo 1 serão gerados os modelos SD por meio de diretrizes para identificação de atores e dependências, sendo um modelo voltado apenas ao desenvolvimento do sistema e outro ao projeto como um todo. No passo 2 serão gerados os modelos SR seguindo uma diretriz para mapear razões estratégicas com foco nos atores, novamente um para o sistema e um para o projeto.

No passo 3 serão listadas as atividades, para um levantamento de atividades mais eficaz, as mesmas serão identificadas por uma diretriz para identificação de atividades implícitas, na sequência outra diretriz é seguida para aplicação de um *template* a fim organização das informações, identificando em cada *template* a qual dos diagramas o mesmo pertence (S para sistema, P para projeto).

No passo 4 serão realizadas as estimativas dos recursos para as atividades, os quais estão associados unicamente ao projeto, não sendo neste passo portanto necessária a análise de atividades do sistema. No passo 5 serão sequenciadas com base na disponibilidade de recursos, limitações entre as mesmas e entre marcos e entregas do projeto. No passo 6 as estimativas de duração são obtidas via alguma técnica, sendo então apresentadas algumas sugeridas para tal.

Por fim no passo 7 é gerado o cronograma com base nos modelos, nas atividades e as informações contidas nos *templates*.

4.3.1 Passo 1: gerar os modelos de dependências estratégicas SD

Para a criação dos modelos de dependências estratégicas, será de grande auxílio a análise de alguns documentos relativos ao projeto, ou um estudo da organização e da equipe de desenvolvimento. Neste contexto, é desejável que as seguintes entradas estejam disponíveis:

- Declaração de escopo do projeto.
- Plano de gerenciamento de recursos humanos.
- Fatores ambientais da empresa.
- Ativos de processos organizacionais.

Caso não haja documentação referente ao projeto, o processo pode ser feito por meio da análise organizacional e do estudo etnográfico da mesma. Não é estritamente necessária a existência de documentos pra serem analisados, mas os mesmos facilitam muito o processo.

Descreverem-se na seguinte diretriz como proceder para geração dos modelos SD, do projeto e do sistema respectivamente, caso haja ou não documentação.

4.3.1.1 Passo 1 - Diretriz 1: identificar os atores participantes do projeto

Possuindo como base as informações contidas na documentação, será possível identificar quais são os atores que irão interagir com o projeto, sendo que estes não são limitados a equipe de desenvolvimento, pois incluem o cliente e podem, quando necessário, incluir outros atores externos (um especialista contratado para uma fase do projeto, por exemplo).

Os atores envolvidos no projeto podem ser identificados via observação do ambiente e analisando a declaração de escopo do projeto e o plano de gerenciamento de recursos humanos, bem como analisando a equipe envolvida no projeto. Se a ação de um ator externo for necessária ao projeto (um especialista contratado para fazer uma análise comercial, por exemplo), o mesmo deverá ser mapeado para o modelo.

Caso não haja documentação os atores podem ser identificados questionando o gerente de projeto acerca de quantos membros constituem a equipe de desenvolvimento e quais as atribuições dos mesmos.

Também é possível observar o ambiente de trabalho e questionar individualmente quais são as tarefas executadas por cada membro da equipe, obtendo assim informação suficiente para classificá-lo em alguma categoria (tal como testador, programador, analista de sistemas) e mapeá-lo como um ator para o modelo SD.

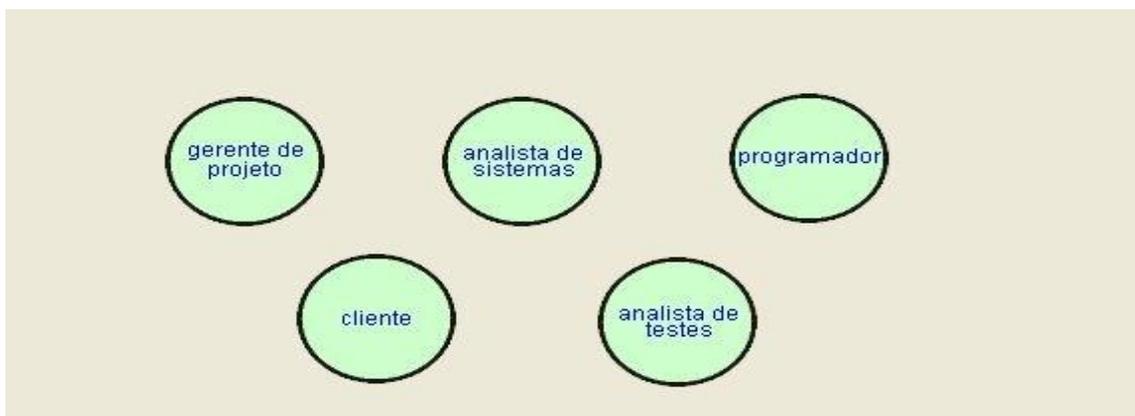


Figura 4.1: Representação de atores participantes de um projeto.

Para o presente estudo de caso abordado no Capítulo 5, serão utilizados os atores representados na Figura 4.1, outros atores identificados e o sistema (o sistema representa um ator nos modelos voltados ao produto). Tendo identificado os atores, o próximo passo será identificar os objetivos, motivações e intencionalidades entre os mesmos, e mapeá-los em termos de dependências entre os atores relacionados.

4.3.1.2 Passo 1 - Diretriz 2: identificar dependências entre os atores

As relações propostas pelo *framework* i* mapeiam uma relação entre dois atores em termos de dependências. Neste contexto, pode-se analisar a declaração de escopo do projeto a fim de identificar quais são as atividades necessárias para sua conclusão, estas atividades serão representadas como dependências (*dependum*).

Em seguida deve-se, com base nas atribuições dos atores, identificar qual ator dentre os atores identificados previamente, será o responsável pela sua execução (*dependee*), e qual ator depende da sua execução (*dependor*).

Cada *dependum* encontrado possui uma natureza diferente no relacionamento entre os atores, podendo ser uma tarefa, um objetivo, um objetivo-soft ou um recurso (ver Capítulo 3), sendo a mesma identificada por meio da análise da documentação anteriormente mencionada.

No caso de não haver documentação disponível as atividades podem ser identificadas com base em premissas a respeito do escopo do sistema e na análise dos requisitos do sistema efetuada pela equipe. A experiência da equipe também ajuda neste caso.

A seguir será demonstrado como se efetua o mapeamento de uma dependência do tipo tarefa no exemplo adotado, onde o gerente de projetos dependendo do analista de testes para execução da atividade “definir casos de teste”.

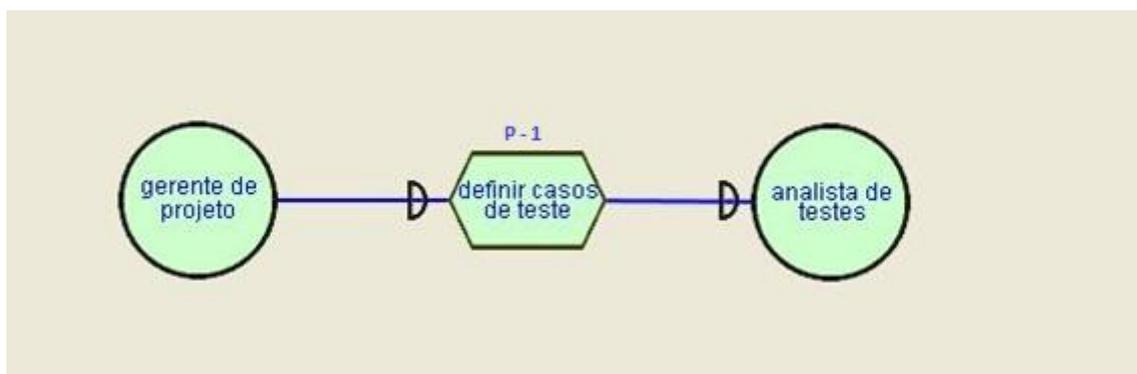


Figura 4.2: Representação de dependência do tipo tarefa entre dois atores.

Uma tarefa do tipo objetivo é uma tarefa com substantiva relevância no contexto do projeto, tal como o objetivo de “entregar versões executáveis” (protótipos) do qual o gerente de projeto depende, e cuja responsabilidade de execução é do programador.

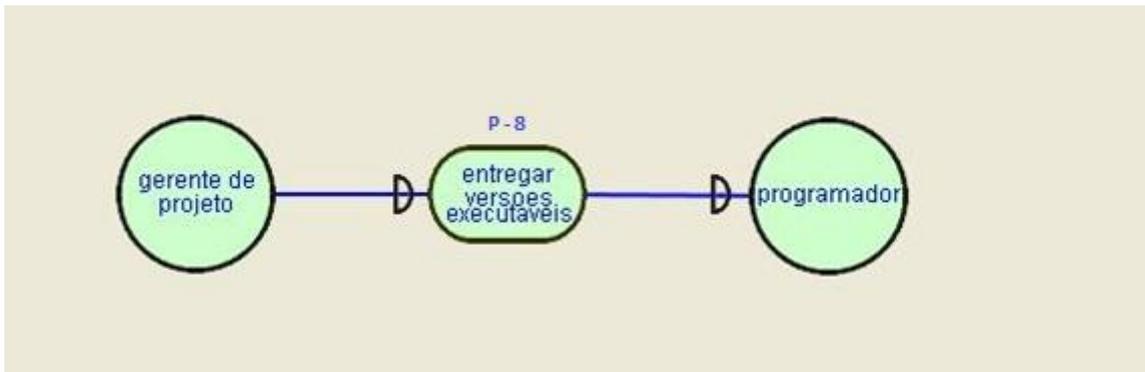


Figura 4.3: Representação de dependência do tipo objetivo entre dois atores.

Quando um recurso, independentemente de sua natureza, é necessário para a execução do projeto ele também será mapeado. No exemplo a seguir o gerente de projeto depende do analista de sistemas para obter os “requisitos funcionais”.

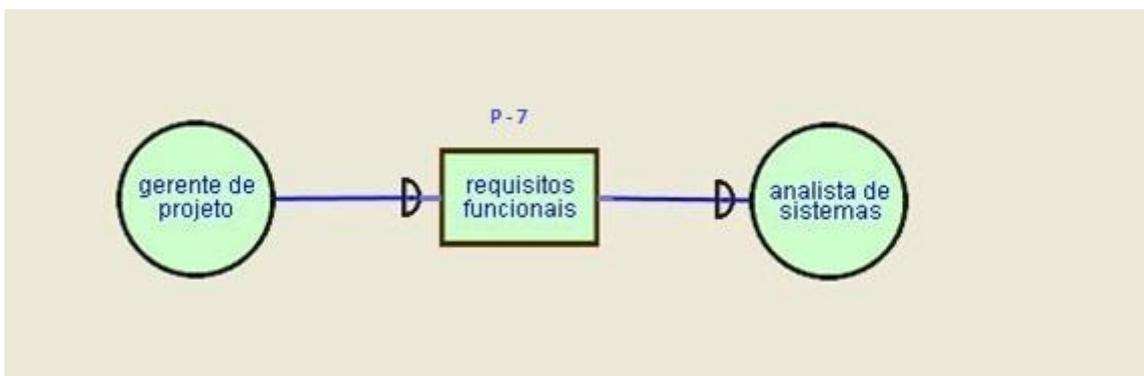


Figura 4.4: Representação de dependência do tipo recurso entre dois atores.

Intencionalidades, desejos e motivações por parte dos atores também representam dependências do tipo objetivo-soft (*softgoal*). O cliente depende do gerente de projeto para obter um “software com qualidade”, é um desejo do cliente, mas é de grande relevância para o projeto.

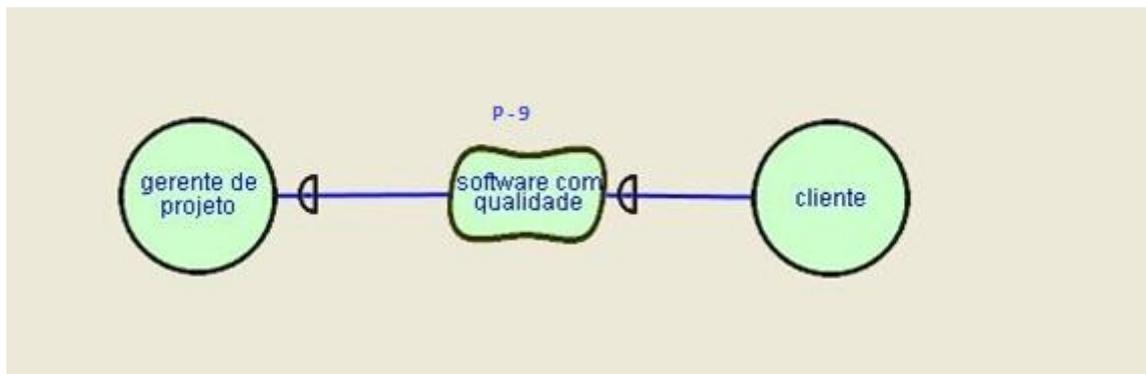


Figura 4.5: Representação de dependência do tipo objetivo-soft entre dois atores.

Também será necessária uma análise dos fatores ambientais e ativos de processos organizacionais da empresa, além de uma análise da relação entre os atores envolvidos no projeto, a fim de identificar possíveis dependências que estão implícitas ao desenvolvimento do sistema (vide Figura 4.5).

Tendo a lista de atores e as atividades com suas dependências especificadas, segue-se para a próxima diretriz.

4.3.1.3 Passo 1 - Diretriz 3: gerar os modelos de dependências estratégicas com foco nos atores envolvidos no processo

O modelo SD deve ser gerado com foco nos atores envolvidos no projeto, com o intuito de expressar dependências entre os mesmos, a Figura 4.6 apresenta o modelo SD voltado para a equipe e o ambiente organizacional.

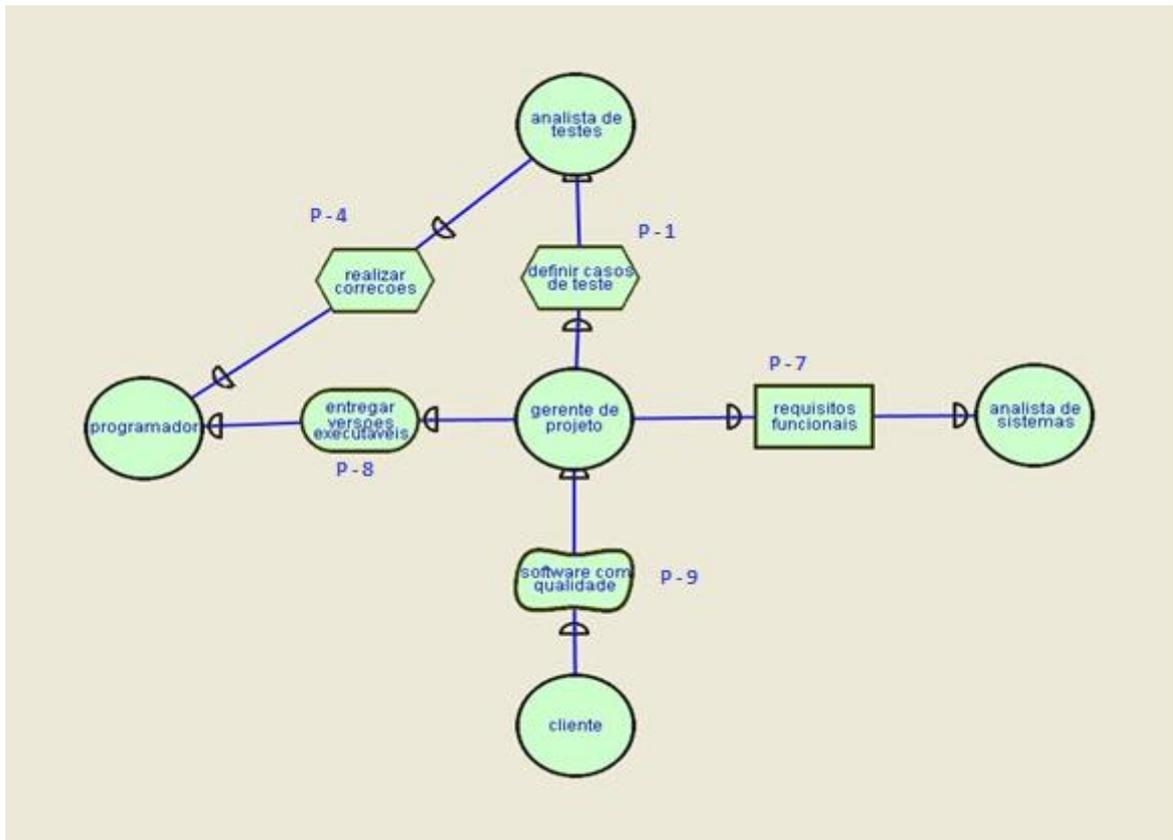


Figura 4.6: Modelo SD do projeto com foco nos atores envolvidos no projeto.

No exemplo descrito na Figura 4.6 observa-se que o cliente se relaciona com o gerente de projeto, desejando um software com qualidade. Para realizar tal desejo o gerente de projetos depende do analista de sistemas para obter requisitos funcionais, do analista de testes para definir casos de teste e do programador para entregar as versões executáveis do sistema. O analista de teste depende do programador para realizar as correções nas funcionalidades.

Quanto ao modelo SD do sistema, o mesmo será feito de modo análogo ao SD do projeto, porém todas as dependências são expressas entre dois atores: o cliente e o sistema. As dependências expressas neste modelo são do tipo objetivo pois representam os requisitos funcionais do sistema. Dependências do tipo recursos e softgoals são expressos apenas no modelo do projeto tendo em vista que o modelo do sistema visa apenas expressar atividades e objetivos voltados ao desenvolvimento do mesmo.

As dependências entre cliente e sistema podem ser identificadas tanto por meio do documento de requisitos do projeto quanto por comunicação direta com o cliente. A Figura 4.7 exibe o modelo SD do sistema.

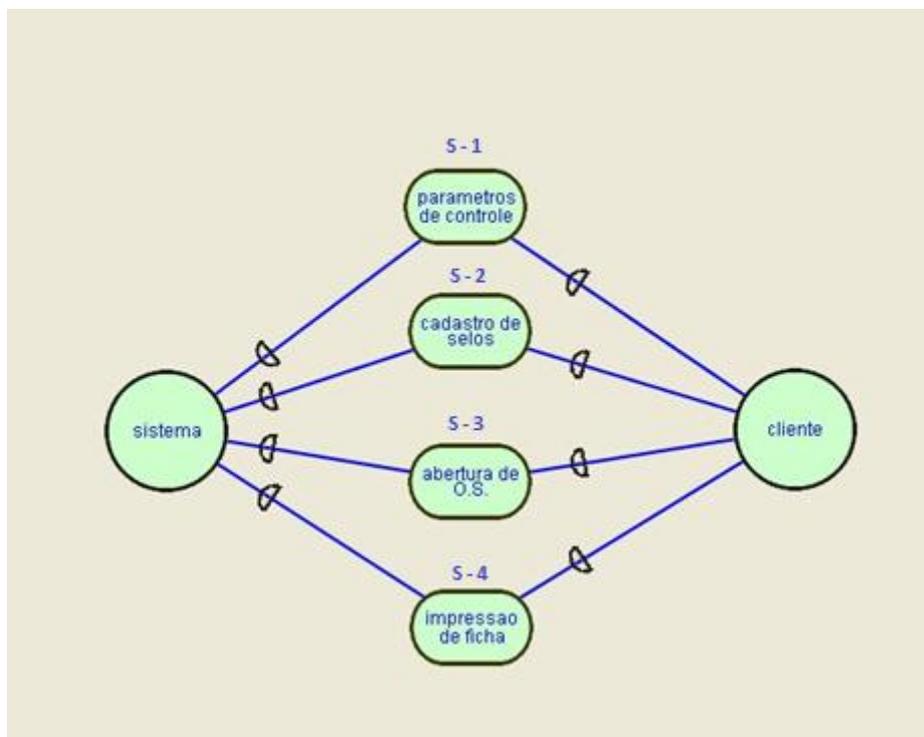


Figura 4.7: Modelo SD do sistema com foco nos atores envolvidos no projeto.

Como saída resultante do passo 1, tem-se o modelo de dependências estratégicas SD.

4.3.2 Passo 2: gerar os modelos de razões estratégicas SR

Partindo-se dos diagramas SD gerados no passo 1, cada ator que fizer o papel de *dependee* ao menos uma vez deverá ter detalhadas as atividades de sua responsabilidade, de modo que para cada dependência mapeada obtenha-se o maior detalhamento possível em termos de decomposição de atividades, a fim de obter maior precisão nas estimativas que serão realizadas.

Os seguintes fatores, artefatos ou documentos podem ser analisados se disponíveis, a fim de auxiliar no processo para uma correta construção dos modelos SR:

- Modelos de dependências estratégicas SD (foco deste trabalho).
- Declaração de escopo do projeto
- Plano de gerenciamento de recursos humanos.
- Fatores ambientais da empresa.
- Ativos de processos organizacionais.

Com relação ao modelo SR do projeto, as atividades podem ser identificadas na declaração de escopo do projeto, mas ainda é necessária a análise dos ativos de processos organizacionais e os fatores ambientais a fim de identificar possíveis fatores que influenciem na construção do diagrama SR. Neste ponto contamos ao menos com o diagrama SD cujos passos para obtenção foram previamente descritos.

4.3.2.1 Passo 2 - Diretriz 4: gerar os modelos de razões estratégicas com foco nos atores envolvidos no processo

Para a construção do modelo SR devem-se detalhar as razões estratégicas acerca das dependências do tipo tarefa ou objetivo, tomando como base o modelo SD e consultando os documentos e informações listados anteriormente, quando disponíveis.

Novamente as atividades podem ser obtidas da declaração de escopo do projeto. Com base no modelo SD, analisa-se cada atividade ou objetivo e realiza-se a decomposição do mesmo em atividades de menor complexidade, se possível.

Em última análise os elementos expressos no modelo SD representam os pacotes de atividade descritos formalmente em uma estrutura analítica de projeto (EAP), e por meio do SR podem-se descrever as sub-atividades das quais cada pacote é composto.

Para realizar o detalhamento pode-se ainda questionar cada *dependee*, levantando informações acerca de quais são os passos necessários para a conclusão da atividade que o mesmo deve satisfazer.

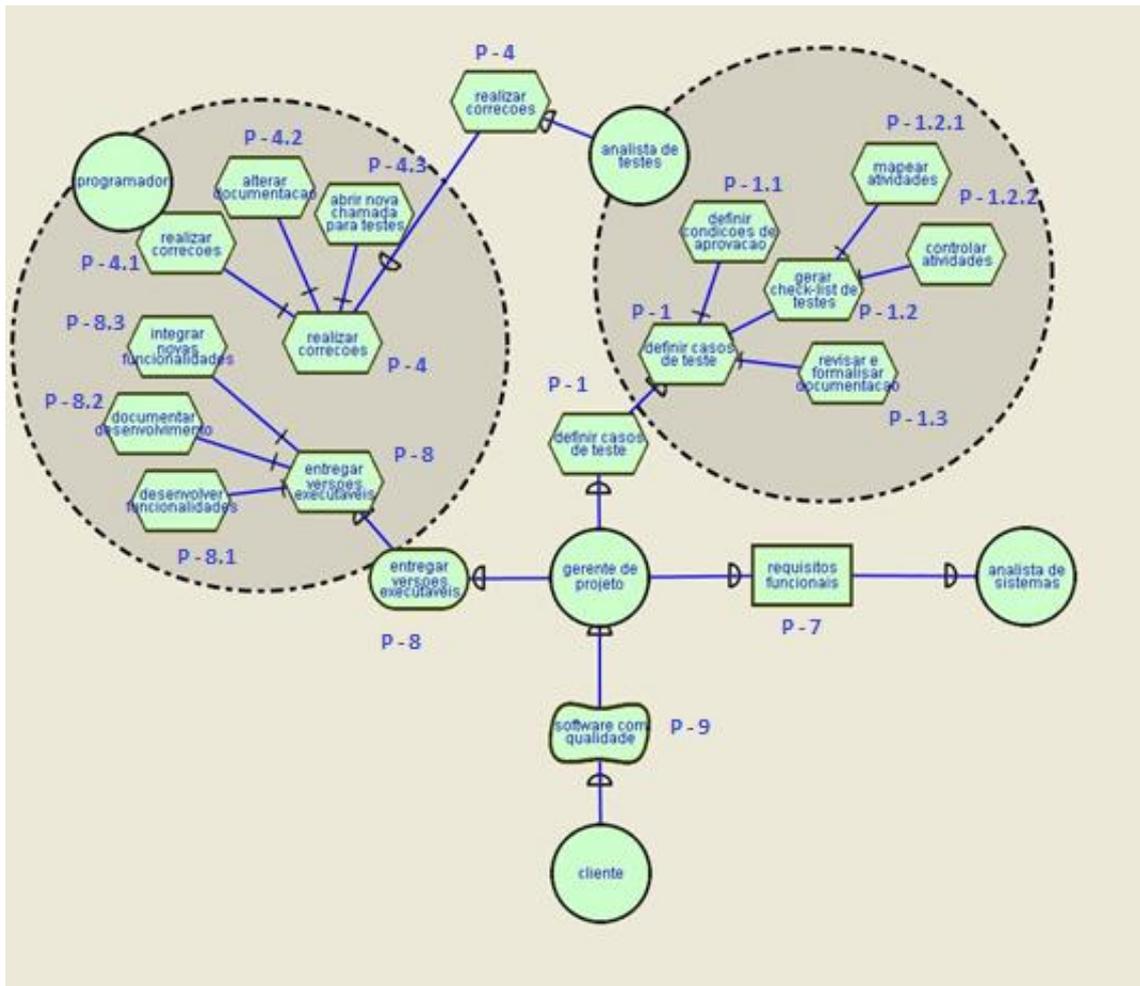


Figura 4.8: Modelo SR do projeto com foco nos atores envolvidos no projeto.

Como se pode ver na Figura 4.8, uma atividade pode ser composta de diversas atividades menores. Para exemplificar, observa-se a tarefa de definir casos de teste.

Para conclusão de tal atividade será necessário definir condições de aprovação para os testes, gerar a *check-list* de testes e revisar e formalizar a documentação. Porém, a atividade gerar *check-list* de testes é composta por duas outras atividades: mapear e controlar atividades. Sendo assim para definir um caso de teste 5 atividades de menor complexidade terão de ser executadas.

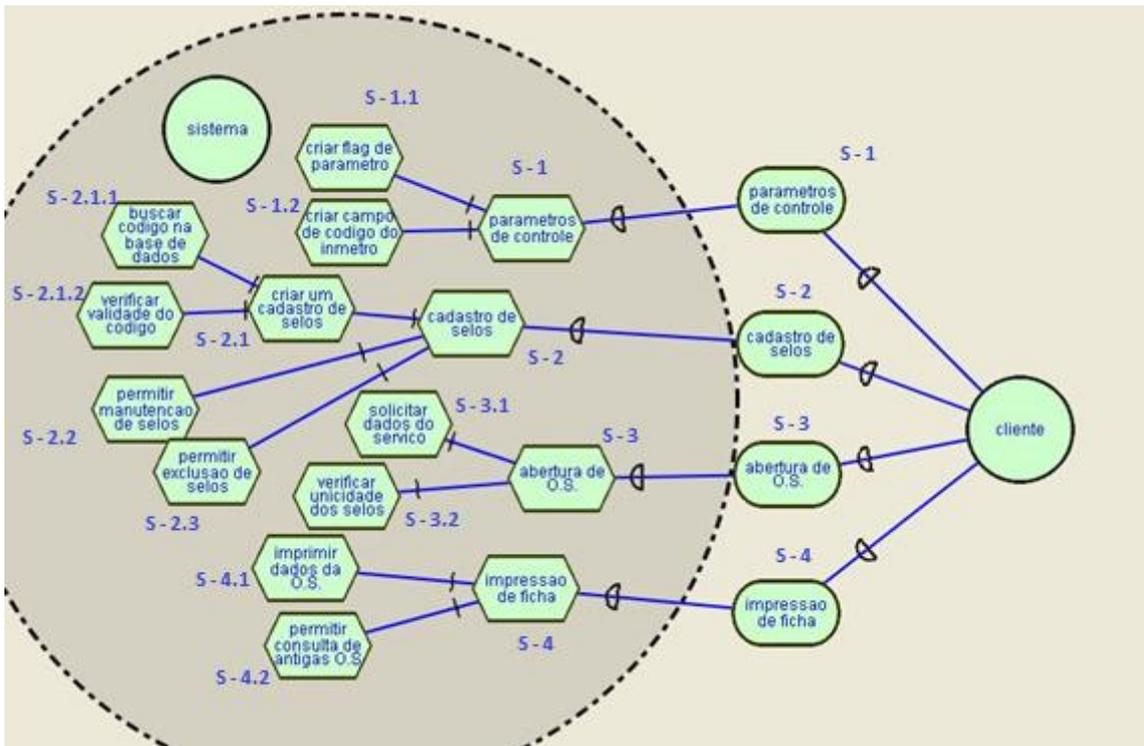


Figura 4.9: Modelo SR do sistema com foco nos atores envolvidos no projeto.

Na Figura 4.9 podem-se observar as atividades necessárias para realização dos objetivos expressos na Figura 4.7 (SD do sistema). Para fins de exemplificação detalha-se a seguir o objetivo cadastro de selo.

O Objetivo será mapeado para uma tarefa de responsabilidade do sistema, que deve permitir efetuar o cadastro de uma série de selos verificando se os mesmos já não constam na base de dados. Também deve ser possível alterar e excluir selos.

Assim como os modelos SD, os modelos SR agora também fazem parte dos documentos do projeto e podem ser usado nos próximos passos.

Nesta etapa, as atividades e os atores já foram identificados e modelados em termos de dependências nos modelos SD e SR, porém ainda é possível obter um maior detalhamento observando os recursos expressos no modelo SR do projeto, processo descrito na Diretriz 5.

4.3.2.2 Passo 2 - Diretriz 5: detalhar razões estratégicas de recursos

Deve-se realizar uma análise de cada recurso presente no modelo SR do projeto, a fim de verificar quais dos mesmos podem ser expressos em termos de tarefas, possibilitando assim estimativas mais precisas para obtenção ou realização dos mesmos. Novamente para auxiliar em tal tarefa pode-se questionar o responsável por fornecer o recurso, a fim de descobrir quais as atividades necessárias para a obtenção do mesmo, ou buscar no calendário de recursos, se disponível.

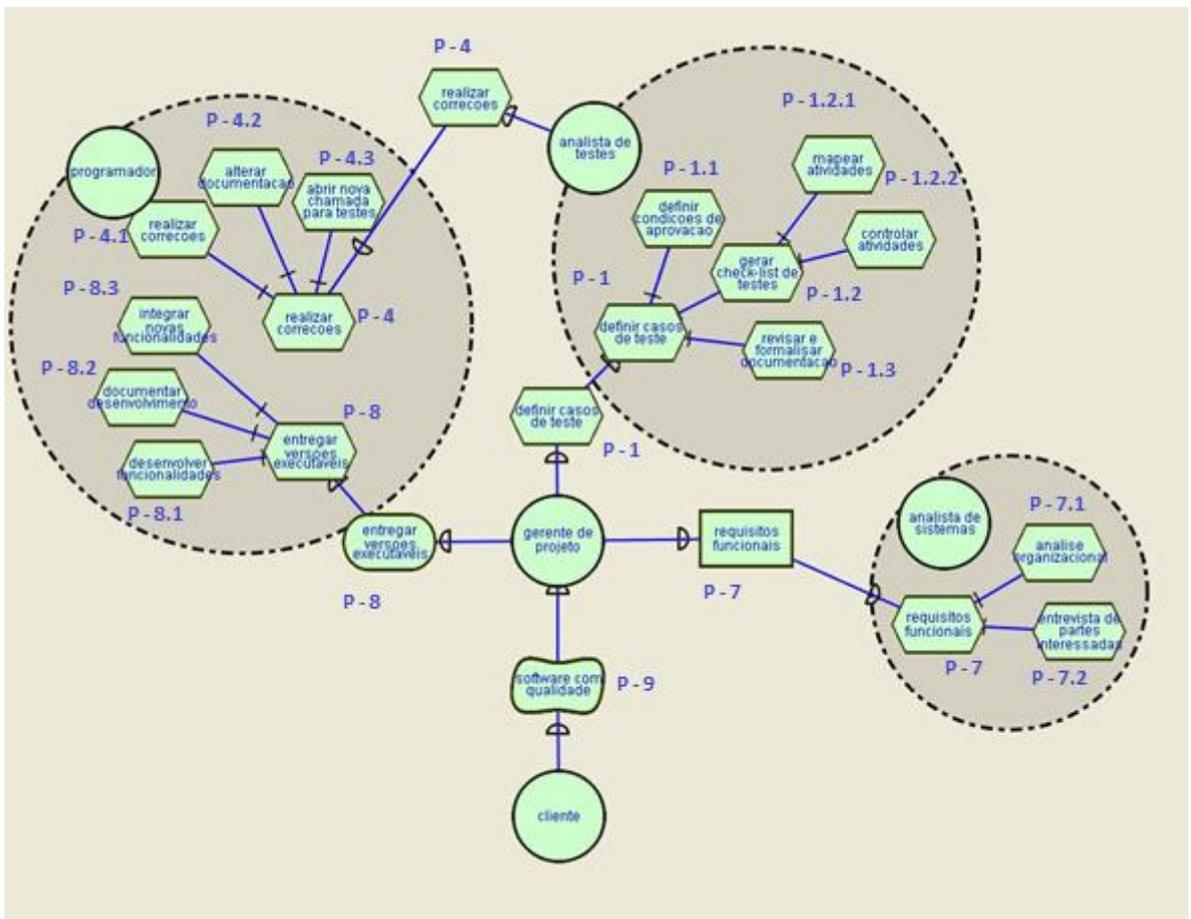


Figura 4.10: Detalhamento de recursos.

É possível observar comparando as Figuras 4.8 e 4.10 que o recurso “requisitos funcionais” pôde ser decomposto em atividades de complexidade menor, pois para a obtenção

dos mesmos serão dispendidos recursos, que agora com um maior nível de detalhamento, poderão ser estimados de maneira mais precisa.

Com os modelos SD e SR prontos, já é possível identificar as atividades necessárias para a conclusão do projeto, porém para que o processo de gerência de tempo tenha sequência, devem-se organizar todas as atividades de modo que, para cada atividade estejam disponíveis suas informações inerentes tais: identificação, descrição, *depender*, *dependee*, recursos necessário para realização, atividades limitantes e limitadas, atividade antecessora e predecessora e estimativa de tempo.

4.3.3 Passo 3: listar as atividades

Para realizar este passo, os seguintes artefatos serão utilizados:

- Modelos de dependências estratégicas SD
- Modelos de razões estratégicas SR.

As informações obtidas até o presente momento já permitem conhecer as atividades referentes ao projeto. Para organizar tais informações, propomos o *template* exibido na Tabela 4.1. Os elementos que o constituem serão descritos na Tabela 4.1, porém sua exemplificação será feita ao longo deste passo e dos passos seguintes, a partir da Diretriz 6, conforme as informações necessárias forem sendo obtidas.

Identificação da atividade	Identificação - Número da atividade - Nome da atividade.
Descrição da atividade	A descrição da atividade, podendo citar dentre outras informações, a atividade pai, qual o pacote de atividades ao qual integra.
<i>Depender</i>	Ator dependente da realização da atividade.
<i>Dependee</i>	Ator responsável pela realização da atividade.
Atividade antecessora	Atividade que deve ser realizada imediatamente antes da atividade atual.
Atividade predecessora	Atividade que deve ser realizada imediatamente após a

	atividade atual.
Atividades limitantes	Atividades cuja conclusão é necessária antes de iniciar-se a execução da atividade atual.
Atividades limitadas	Atividades cuja execução depende da conclusão da atividade atual.
Estimativa de recursos	Recursos humanos, computacionais, financeiros ou de qualquer outra natureza necessários para a conclusão da atividade.
Estimativa de duração	Estimativa de duração da atividade

Tabela 4.1: *Template* para listar atividades.

Dentre outras razões, o *template* da Tabela 4.1 é proposto como um modo de permitir conectar informações entre atividades e sub-atividades, além de prover uma melhor organização das informações disponíveis além dos modelos SD e SR. Contudo o uso da Tabela 4.1 tem por principal objetivo prover informações para obtenção do cronograma em conjunto com os modelos SD e SR no passo 7.

4.3.3.1 Passo 3 - Diretriz 6: Aplicar o *template* para todas as atividades

Neste ponto já estão identificadas as atividades, agora as mesmas serão listadas com base nos diagramas. Para cada atividade ou sub-atividade identificada nos modelos SD e SR, deve-se gerar um *template*. A Tabela 4.2 apresenta um exemplo de *template* de atividade gerado.

Identificação da atividade	P - 1.2 - gerar <i>check-list</i> de testes
Descrição da atividade	Consiste em identificar o que já foi feito, o que está sendo feito e o que será feito no teste de uma funcionalidade.
<i>Depender</i>	Gerente de projeto.
<i>Dependee</i>	Analista de testes.

Atividade antecessora	P - 1.1 - definir condições de aprovação.
Atividade predecessora	P - 1.2.1 - mapear atividades.
Atividades limitantes	P - 3.1 - desenvolver funcionalidades.
Atividades limitadas	P - 4 - realizar correções.
Estimativa de recursos	-
Estimativa de duração	-

Tabela 4.2: *Template* da atividade gerar *check-list* de testes, após listagem de atividades.

O exemplo aplicado na Tabela 4.2 diz respeito à tarefa de gerar *check-list* de testes, a qual faz parte do projeto (P) e é a segunda sub-tarefa para definir casos de teste (atividade 1 do projeto), e cuja descrição consta no segundo campo da tabela.

Antes de gerar um *check-list*, as condições de aprovação devem ser especificadas, quanto a sua atividade sucessora, a revisão e formalização da documentação deverá ser executadas após os testes serem realizados. Para conclusão da atividade em questão será necessário mapear e controlar as atividades identificadas para conclusão do teste.

Ainda no contexto do exemplo, um *check-list* só poderá ser criado caso haja um teste a ser realizado, logo seus *dependor* e *dependee* serão os mesmos, e um teste só será realizado se já houver uma funcionalidade a ser testada. De modo semelhante, uma correção só pode ser realizada se uma funcionalidade foi reprovada em um teste.

Assim como no nível do projeto podem existir dependências entre atividades (tal como um teste necessita de uma funcionalidade já desenvolvida), podem também tais dependências estarem presentes no nível do sistema, como por exemplo, a aplicação de um selo em um cronotacógrafo somente será possível se existirem selos válidos cadastrados.

Tendo os modelos SD e SR, e as atividades listadas, será feita a estimativa de recursos para cada atividade no passo 4, agregando informações nos *templates*.

4.3.4 Passo 4: estimar recursos para atividades

Neste passo as seguintes informações, artefatos e documentos são desejáveis:

- Modelo de dependências estratégicas SD do projeto
- Modelo de razões estratégicas SR do projeto
- *Templates* das atividades
- Calendário de recursos (se disponível)
- Fatores ambientais da empresa (se disponíveis)
- Ativos de processos organizacionais (se disponíveis).

Neste passo não serão analisados os modelos SD e SR do sistema, visto que recursos necessários ao seu desenvolvimento são tratados no nível de projeto, de modo que somente será necessária a obtenção de uma estimativa de duração para cada atividade.

Com relação à aplicação de técnicas de estimativa sobre modelagem organizacional aliadas aos *templates*, serão enfatizadas as técnicas ou métodos que possibilitem o uso das informações obtidas até agora no processo.

No processo descrito segundo o PMBOK [1], técnicas de estimativa *bottom-up* não são eficientes, pois as atividades já se encontram com alto grau de fragmentação para fins de facilitar o processo de estimativas. Softwares que auxiliem no processo também não possuem relevância com relação a esta proposta.

Para realizar a estimativa de recursos para as atividades, deve-se analisar o calendário de recursos (se houver um) para verificar quais são os recursos disponíveis, e quando estarão disponíveis para uso no projeto.

Nesta etapa serão verificados os fatores ambientais e os ativos organizacionais, a fim de descobrir, dentre outras coisas, se já houve um projeto semelhante desenvolvido na empresa, ou se algum funcionário já possui experiência com problemas semelhantes.

Em um contexto em que haja a documentação sugerida, os recursos já se encontram listados e com informações relativas ao mesmos disponíveis ao gerente de projeto, sendo que o processo de estimativas pode ser realizado com base no histórico de projetos da empresa, na experiência da equipe, ou ainda com base em uma técnica, regra ou software padrão aplicado na empresa.

Caso não haja documentação, um levantamento sobre os recursos disponíveis pode ser feito por meio de uma análise da empresa, e o questionamento dos funcionários pode ajudar no sentido de detectar fatores ou recursos externos necessários ao projeto.

Destaca-se que até o presente momento as estimativas não incluem o fator tempo, abrangendo dentre outros fatores: equipe, espaço físico e hardwares disponíveis, softwares necessários, contratação de especialistas, banco de dados do sistema e quaisquer outros tipos de recursos que sejam necessários. A alocação da equipe como recurso nesta etapa não considera seu uso em horas, já que a estimativa de duração obtida no Passo 6 expressa a demanda de uso de tal recurso.

Como o uso de recursos por uma atividade está diretamente vinculada a múltiplos fatores que interferem diretamente no processo de estimativa, tais como suas dependências, sua complexidade e a etapa do projeto na qual a mesma se encontra, e sabendo que algumas destas informações estão presentes no *template*, podem-se usar os mesmos em conjunto com os diagramas como base para realizar as estimativas.

Dentre os fatores que podem ser considerados como relevantes para estimativas, com relação a uma funcionalidade, estão: número de entidades, telas, relatórios, casos de teste e documentação, quais foram criados e quais somente modificados, além de sua complexidade. Podem-se ainda considerar o treinamento e a implantação como fatores.

Sugere-se que juntamente aos fatores anteriormente listados, sejam consideradas as atividades limitantes e as limitadas, de modo que como resultado deste passo obtenha-se para cada atividade o seu *template* modificado, contendo agora informações a respeito dos recursos necessários a sua conclusão.

Com os modelos, os *templates*, e informações sobre os recursos, as estimativas podem agora ser feitas pelo gerente de projetos, o qual deve dispor da equipe para obtenção de quaisquer informações necessárias no processo.

4.3.4.1 Passo 4 - Diretriz 7: Realizar estimativa de recursos necessários

Analisa-se cada atividade e com relação aos recursos identificados previamente, definem-se as estimativas conforme pode ser observado na Tabela 4.3. A decisão fica a encargo do gerente de projeto, mas tal atividade pode contar com o auxílio do ator envolvido no processo. No caso apresentado na Tabela 4.3 a adesão de uma ferramenta de testes automatizada não apenas diminuiria o tempo de execução dos testes, como também a realização da atividade 1.2.2 - Controlar atividades seria feita automaticamente, resultando em um relatório gerado por uma ferramenta de testes automatizados.

Identificação da atividade	P - 1.2 - gerar <i>check-list</i> de testes
Descrição da atividade	Consiste em identificar o que já foi feito, o que está sendo feito e o que será feito no teste de uma funcionalidade.
<i>Depender</i>	Gerente de projeto.
<i>Dependee</i>	Analista de testes.
Atividade antecessora	P - 1.1 - definir condições de aprovação.
Atividade predecessora	P - 1.2.1 - mapear atividades.
Atividades limitantes	P - 3.1 - desenvolver funcionalidades.
Atividades limitadas	P - 4 - realizar correções.
Estimativa de recursos	<ul style="list-style-type: none"> • 1 analista de testes • 1 computador com acesso ao banco de dados de testes • Ferramenta de testes automatizados disponível
Estimativa de duração	-

Tabela 4.3 *Template* da atividade gerar *check-list* de testes, após estimativa de recursos.

4.3.5 Passo 5: sequenciar atividades

Para realizar este passo, os seguintes artefatos serão utilizados:

- Modelos de dependências estratégicas SD
- Modelos de razões estratégicas SR
- *Templates* das atividades
- Lista de marcos de controle (se disponível).
- Declaração de escopo do projeto. (se disponível).

Tendo as atividades listadas e com os recursos necessários devidamente estimados, será realizado o sequenciamento das atividades de modo que não haja conflito quanto a dependências entre atividades ou recursos, além de considerar marcos de controle e entregas previstas na declaração de escopo do projeto.

4.3.5.1 Passo 5 - Diretriz 8: sequenciar as atividades

As atividades serão sequenciadas com base nos diagramas, na lista de marcos de controle (se disponível) e na declaração de escopo do projeto (se disponível).

Inicialmente devem-se observar os marcos de controle e a declaração de escopo do projeto, a fim de identificar qual a sequência de atividades necessária para realizar as entregas, ou seja, identificar quais os módulos do sistema e qual sua sequência de desenvolvimento, com relação a suas dependências. Tais módulos ou pacotes de atividades são representados nos modelos SD. Dependências entre atividades localizadas em diferentes pacotes de atividades podem ocorrer.

Por meio do modelo SD que evidencia os pacotes de atividades, e o SR que expressa limitações entre atividades considerando sua hierarquia, deve-se fazer uma análise para cada atividade, a fim de identificar quais as limitações por elas geradas, quais são suas limitantes, e suas atividades antecessoras e predecessoras. Também há um campo para identificação da atividade, sendo que um pacote representado no modelo SD recebe um número de identificação, e suas sub-atividades representadas no modelo SR recebem identificações indexadas pelo mesmo número de atividade, porém com suas próprias subdivisões.

Com a documentação recomendada as atividades podem ser visualizadas nos modelos e classificadas com os *templates*, cujas informações adicionais podem ser obtidas via declaração de escopo e lista de marcos.

Na ausência de documentação, estas informações podem ser obtidas com o encarregado de sua realização, no contexto da Tabela 4.3 será o *dependee* analista de testes. Por ser o ator responsável pela geração das *check-lists* de teste, o analista conhece as atividades antecessoras e predecessoras, quais as atividades das quais sua conclusão depende e quais sofrem dependência, além de naturalmente saber quem é o *dependor*.

Com as atividades identificadas e sequenciadas, o próximo passo é definir custos em termos de tempo. No Passo 6 serão apresentadas diretrizes para auxiliar no processo de estimativas de duração das atividades.

4.3.6 Passo 6: estimar a duração das atividades

Os documentos que irão auxiliar nesta etapa do processo encontram-se listados abaixo:

- Modelos de dependências estratégicas SD.
- Modelos de razões estratégicas SR.
- *Templates* das atividades.
- Declaração de escopo do projeto (se disponível).
- Fatores ambientais da empresa (se disponíveis).
- Ativos de processos organizacionais (se disponíveis).

Em posse dos recursos necessários para conclusão das atividades listados no *template*, a etapa de estimativa de duração pode ser iniciada. Caso esteja disponível dentre os documentos do projeto a declaração de escopo, a mesma será analisada a fim de identificar restrições e premissas do projeto que possam de algum modo interferir no processo de estimativa, tal como a disponibilidade de um módulo somente após um prazo previamente definido.

Nesta etapa já se encontram identificados no *template* as informações acerca das atividades e os recursos e informações acerca dos mesmos quanto à disponibilidade quando necessário (quando a atividade for referente ao projeto). Fatores ambientais e ativos organizacionais, se

presentes também devem ser analisados pelo gerente de projeto, pois podem interferir no processo.

A obtenção das estimativas de duração de uma atividade pode ser obtida por meio da soma das estimativas das sub-tarefas necessárias a sua conclusão, sendo portanto interessante a análise *bottom-up* neste passo.

Para conclusão desta etapa é possível fazer uso de várias técnicas, dentre as mesmas listam-se aqui as estimativas análoga, COCOMOII e PERT, além da estimativa análoga e da opinião especializada.

Na sequência será realizada uma breve descrição das técnicas, atentando-se para as diferenças na natureza das mesmas, que são baseadas em parâmetros, média de expectativas, opinião ou julgamento de um especialista, e históricos tanto analisados de maneira tradicional quanto por meio de técnicas vide COCOMO II.

Algumas técnicas aqui apresentadas tem como base a experiência, histórico ou estatísticas organizacionais, caracterizando uma técnica baseada no julgamento do gerente de projeto. Outras técnicas apresentam uma relação entre tamanho, esforço, prazo e qualidade, que são parâmetros da atividade, sendo possível traçar uma estimativa a partir destas informações. Estas são técnicas paramétricas.

Encontram-se listadas abaixo algumas técnicas sugeridas para traçar estimativas de duração:

- **Pontos por casos de uso**

Esta técnica gera estimativas com base em uma análise de uma série de parâmetros, cada um com um peso associado, de modo que a estima será dada pela equação $UCP = UUCP * TCF * EF$, onde UCP representa os casos de uso ajustados, que poderá ser convertido para uma estimativa em pessoas/horas.

Para obtenção dos casos de uso ajustados, os fatores necessários são:

- UUCP - pontos por caso de uso não ajustados: os atores participantes do projeto deverão ser contados e classificados de acordo com seu conhecimento/produktividade

com um valor de 1 a 3. Em seguida o mesmo deve ser feito para os casos de uso com relação a sua complexidade assumindo valores de 5, 10 ou 15 dependendo de sua complexidade. Os pontos por caso de uso não ajustados serão resultado da soma destas duas estimativas, sobre todas as atividades.

- TCF – fatores de complexidade técnica: são atribuídos pesos aos fatores técnicos do sistema, tais como portabilidade, usabilidade online, características especiais de segurança dentre outros, sendo que cada fator receberá um valor de 0 a 5, de acordo com sua relevância para o projeto. Cada um dos fatores identificados terá ainda seu próprio grau de complexidade associado, que será usado na sua ponderação.
- EF – fatores ambientais: de modo semelhante aos fatores de complexidade técnica, os fatores ambientais serão listados e terão um peso atribuído, que será considerado juntamente com os graus de peso dos fatores em si.

Esta é uma técnica paramétrica interessante para ser aplicada em projetos em que não há disponibilidade de histórico organizacional. Dentre outras razões para o uso de estimativas por casos de uso destacam-se ainda a sua independência de plataforma, experiência da equipe e tamanho do projeto. Um exemplo de aplicação desta técnica será exibido na Diretriz 8.

Maiores informações sobre pontos por casos de uso podem ser encontradas em Karner [20].

- **PERT – estimativa de duração de três pontos**

De acordo com Willians [21], Esta técnica consiste em traçar uma estimativa pessimista, uma realista e uma otimista para duração da atividade e então aplica-las na equação **PERT = (O+4R+P)/6**, onde PERT recebe a média das estimativas otimista, pessimista e quatro vezes a realista em horas de trabalho.

A precisão das estimativas está vinculada com a experiência do gerente de projeto, o do ator por ele consultado, pois as únicas informações usadas como entrada são suas expectativas. As estimativas podem ser realizadas juntamente com a equipe, de modo que cada um faça suas estimativas e obtenha-se assim uma média.

- **COCOMO II**

Técnica paramétrica que gera estimativas por meio da análise de linhas de código que são passadas como parâmetros de entrada ou dos pontos por função dos projetos já arquivados, o que torna necessária a existência de um histórico organizacional. O COCOMO II pode ser aplicado em três níveis sequenciais de complexidade, segundo Aguiar [22].

O Nível mais básico considera basicamente linhas de código ou casos de uso e desenvolvedores para traçar estimativas de pessoas/horas. No nível médio podem se abranger ainda características da equipe, do projeto, de hardware dentre outros com seus pesos associados. No nível detalhado os fatores listados no nível médio são estimados para todas as etapas do processo.

O COCOMO II é uma técnica interessante para ser usada em organizações que disponham de uma base de dados para obtenção das estimativas, sendo que a precisão das estimativas está ligada ao tamanho do banco.

- **Opinião especializada**

Uma técnica baseada em experiência que pode ser usada principalmente quando o projeto está inserido em um contexto do qual a equipe não possui conhecimento é o uso de opinião especializada.

O gerente de projetos pode julgar necessária a assessoria de um consultor ou especialista, sendo que a estimativa de duração de uma atividade de sua alçada pode ser feita em conjunto com o consultor ou especialista.

- **Estimativa análoga**

Nesta outra técnica usam-se dados estatísticos da empresa referentes a projetos semelhantes, a fim de se buscar uma estimativa coerente para a atividade atualmente em análise. Estimativas análogas podem ser feitas tanto na estimativa de recursos quanto na estimativa de duração.

As estimativas obtidas podem ser adaptadas ao projeto atual pelo gerente de projetos, caso haja discrepância perceptível entre os fatores identificados, de modo que as estimativas sejam feitas com base em um projeto feito no passado, mas que estejam voltadas para o problema atual. Dentre os fatores que podem apresentar discrepância estão: complexidade, que pode ser maior no projeto atual, um maior uso de recursos humanos no projeto analisado ou um prazo de conclusão diferente.

Esta técnica embora nem sempre forneça uma estimativa muito precisa devido ao grau de semelhança entre o projeto atual e o histórico variar, porém é uma técnica de simples aplicação.

Caso o projeto pesquisado foi desenvolvido recentemente, de modo que os atores que o integraram ainda atuem na organização, há a possibilidade de o gerente de projeto contar com a experiência destes indivíduos.

Agora se deve definir uma técnica para realização das estimativas. A decisão por uma técnica está vinculada a vários fatores, e caberá ao gerente de projetos. A seguir nas Diretrizes 8 e 9 são exibidos exemplos da realização de estimativas para o projeto como um todo e para o sistema respectivamente.

4.3.6.1 Passo 6 - Diretriz 9: Realizar estimativa de duração das atividades do sistema

Nesta diretriz será aplicada a estimativa para atividades voltadas apenas para a entrega do produto, ou seja, que considerem apenas o esforço referente a parte de desenvolvimento dos requisitos funcionais do sistema. A partir do modelo SD do sistema faz-se a estimativa para a funcionalidade, e por meio do modelo SR do sistema as sub-tarefas para sua realização podem ter suas próprias estimativas definidas pontualmente, ou para o sistema como um todo.

Atenta-se para o fato de que esta diretriz diz respeito apenas às estimativas voltadas para a entrega do produto, caso não haja intenção de obter tais estimativas, esta diretriz se torna opcional visto que as estimativas para todo o projeto são obtidas na diretriz 9.

Dentre as técnicas usadas para estimativas, será aqui ilustrado o uso de uma técnica baseada em análise e decisão, seja pelo gerente do projeto ou pelo responsável por ele designado e outra técnica baseada em parâmetros, os quais em última instância também são definidos pela equipe, mas com base em questões bem definidas ao invés de um palpite. Segue-se um exemplo de aplicação das técnicas PERT e pontos por casos de uso.

- **PERT**

Tomando como exemplo a atividade S-2 – cadastro de selos, três estimativas são deduzidas, uma pessimista, uma realista e uma otimista. Estimando-se 8 horas de um ponto de vista, 9 horas para um realista e 11 horas para uma estimativa pessimista, obtém-se a estimativa dos três pontos usando-se a equação **PERT = (O+4R+P)/6**.

$PERT = (14 + (4 * 15) + 17) / 6$. O que resulta em uma estimativa de aproximadamente 15,16 horas.

- **Pontos por casos de uso**

A técnica será usada para gerar a estimativa da mesma atividade, de modo que possa ser feito um comparativo entre ambas. A fim de facilitar o processo de obtenção dos casos de uso por meio dos requisitos funcionais, será utilizada a ferramenta open source JGOOSE, a qual foi desenvolvida por Vicente [19] e melhorada por Brischke [17].

Na Figura 4.11 apresentam-se os casos de uso gerados pela ferramenta JGOOSE, a qual gera casos de uso e diagramas para os mesmos a partir dos modelos SD e SR, permitindo que sejam selecionadas quais diretrizes serão aplicadas na análise.

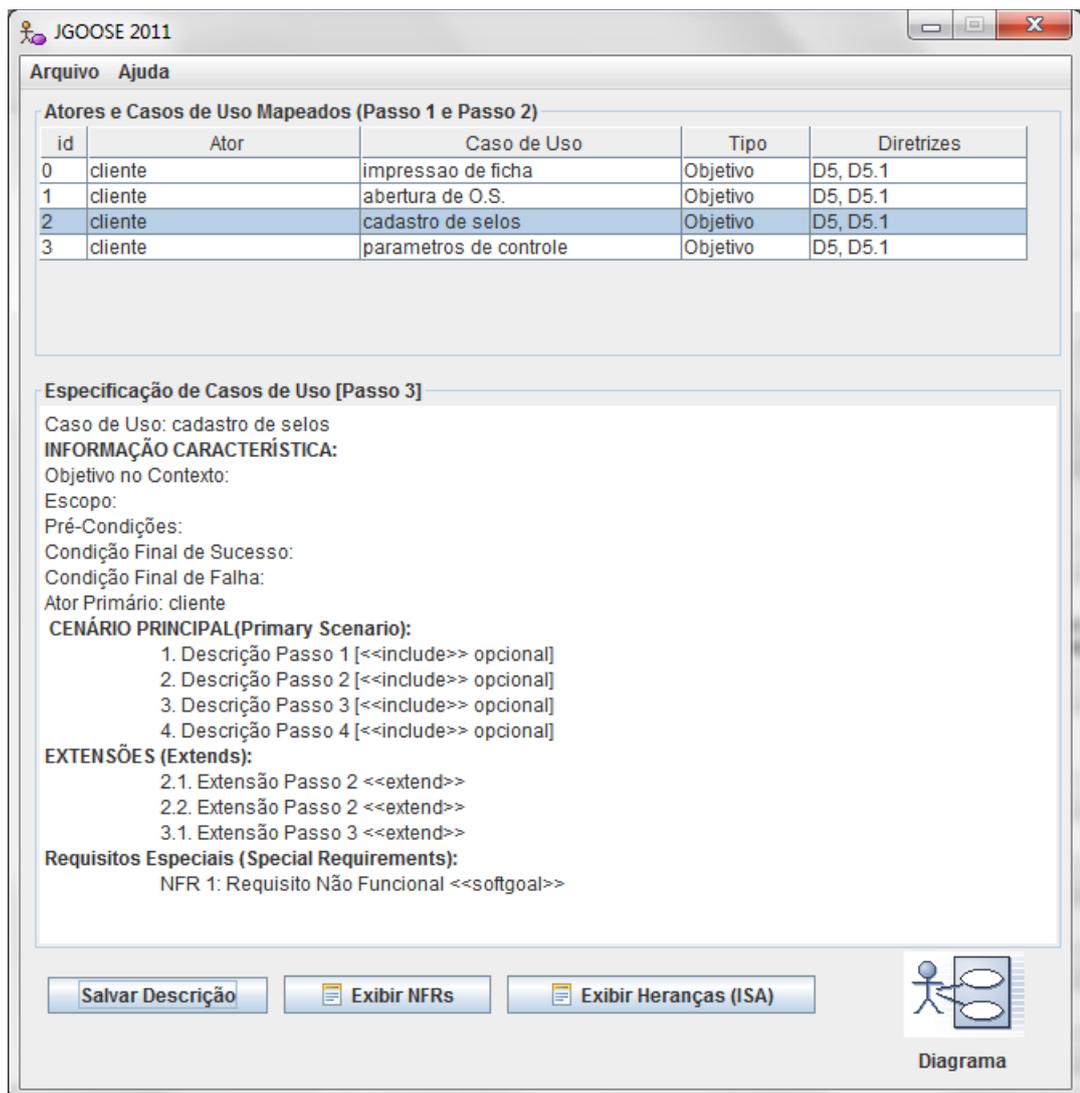


Figura 4.11 Casos de uso obtidos com base na Figura 4.7

Após a obtenção dos casos de uso, eles podem ser salvos em formato de texto e ter acrescidas informações obtidas por meio da documentação ou da equipe, além de obter juntamente a equipe informações referentes a complexidade da tarefa a ser desenvolvida.

Para a aplicação da técnica será usada uma tabela proposta por Karner [20] que considera complexidade dos atores, dos casos de uso, e dos fatores técnicos e ambientais do sistema. As informações necessárias ao seu preenchimento podem ser obtidos com a equipe.

Com base nos quatro casos de uso mapeados, foi feita em conjunto com a equipe obtenção dos parâmetros necessários para gerar a estimativa em horas para o desenvolvimento do sistema, a qual é de aproximadamente 63 horas.

A técnica de pontos por caso de uso avalia o sistema como um todo, com base na análise dos seus parâmetros, em contrapartida a PERT possibilita estimativas pontuais em cada atividade.

4.3.6.2 Passo 6 - Diretriz 10: Realizar estimativa de duração das atividades do projeto

De modo análogo a diretriz aplicada anteriormente, as estimativas para atividades voltadas para a conclusão do projeto como um todo, tanto a parte gerencial quanto a parte de desenvolvimento, podem ser obtidas por meio das mesmas técnicas demonstradas na Diretriz 8, mas desta vez com base nos modelos SD e SR do projeto.

Após obter a estimativa de duração, o *template* da atividade será atualizado, o que aqui será feito para fins demonstrativos usando os valores obtidos com a técnica PERT, por se tratar de uma técnica de estimativa pontual.

O resultado pode ser visto na tabela 4.4, destacando-se que a estimativa PERT obtida junto à equipe (0,0833 horas, ou 5 minutos) foi multiplicada pelo número de casos de uso presente no SD do sistema (9 casos de uso) e arredondada, devido aos mesmos representarem o total de funcionalidades a serem desenvolvidas e, por conseguinte, o número de vezes que será necessário realizar a atividade de gerar *check-list* de testes.

Identificação da atividade	P - 1.2 - gerar <i>check-list</i> de testes
Descrição da atividade	Consiste em identificar o que já foi feito, o que está sendo feito e o que será feito no teste de uma funcionalidade.
<i>Depender</i>	Gerente de projeto.
<i>Dependee</i>	Analista de testes.
Atividade antecessora	P - 1.1 - definir condições de aprovação.
Atividade predecessora	P - 1.2.1 - mapear atividades.
Atividades limitantes	P - 3.1 - desenvolver funcionalidades.

Atividades limitadas	P - 4 - realizar correções.
Estimativa de recursos	<ul style="list-style-type: none"> • 1 analista de testes • 1 computador com acesso ao banco de dados de testes • Ferramenta de testes automatizados disponível
Estimativa de duração (via PERT)	1 hora

Tabela 4.4 *Template* da atividade gerar *check-list* de testes, após estimativa de duração.

Com todas as atividades mapeadas, sequenciadas, tempo de conclusão e recursos estimados, pode-se gerar o cronograma do projeto voltado ao produto ou ao processo todo, de acordo com as atividades e os diagramas usados para tal.

4.3.7 Passo 7: gerar os cronogramas do projeto e do sistema

Nesta etapa a seguinte documentação pode ser utilizada para auxiliar na construção do cronograma.

- Modelos de dependências estratégicas SD.
- Modelos de razões estratégicas SR.
- *Templates* das atividades.
- Declaração de escopo do projeto (se disponível).
- Fatores ambientais da empresa (se disponíveis).
- Ativos de processos organizacionais (se disponíveis).

O objetivo da etapa de definição do cronograma é obter uma data de início e uma de conclusão estimadas para cada atividade, de modo que elas estejam na ordem necessária ao desenvolvimento do projeto.

Existem várias técnicas propostas pelo PMBOK para fazer esta definição de prazos por meio do diagrama de rede do projeto, todavia como o objetivo presente é proporcionar

ênfoque sobre a modelagem organizacional, os diagramas usados serão os modelos SD e SR obtidos.

4.3.7.1 Passo 7 - Diretriz 11: Gerar os cronogramas

Caso esteja disponível, a declaração de escopo do projeto pode ser analisada, pois a mesma pode conter uma lista de marcos definidos para o projeto, ou datas importantes para entrega de versões. Ademais informações a respeito das atividades, tais como recursos necessários e sua disponibilidade já constam nos seus *templates*.

Dentre as técnicas propostas para definir o cronograma, algumas encontram-se listadas na sequência, voltadas ao uso dos modelos e dos *templates*

- Método do caminho crítico: consiste em concentrar as estimativas inicialmente nas atividades críticas para o funcionamento do sistema, então estimar durações de início e término adiantados e atrasados de modo que tais intervalos possam ser adaptados com relação às folgas do projeto e as próprias atividades, considerando limitações, marcos e restrições previamente conhecidas, antecipações, esperas e quaisquer limitações relacionadas a prazos que a equipe ou a documentação ajude a identificar.
- Método da cadeia crítica: consiste em modificar o cronograma obtido pelo caminho crítico e adicionar *buffers* de duração no projeto e entre as atividades, de modo que os prazos para as atividades comecem e terminem mais tarde, e durante o desenvolvimento do projeto, tal folga seja controlada com base nos *buffers*.

O método aqui sugerido para definir o cronograma, consiste em analisar as atividades com uma abordagem *bottom-up*: para cada pacote de atividades identificado no modelo SD, deve-se fazer a análise das suas sub-tarefas expressas no respectivo modelo SR em ordem inversa. Os *templates* podem ser colocados em um mural do projeto na forma de uma árvore *bottom-up*, de modo que as atividades que não possuam limitação estejam no topo, e as atividades limitadas estejam nos níveis abaixo conectadas com as suas atividades limitantes.

A análise para definição do cronograma pode seguir alguma das técnicas anteriormente sugeridas ou outras ainda, apesar de existirem ferramentas de apoio a construção de cronogramas, prioriza-se aqui a análise da documentação e o auxílio da equipe de desenvolvimento. Um cronograma gerado a partir do modelo SR do sistema exibido na Figura 4.10, em conjunto aos *templates* das atividades nele presentes pode ser visto na Figura 4.12. Note que não estão inseridos marcos de controle no projeto.

Nome	Início	Término	Predecessoras	Nome do Recurso
1 - requisitos funcionais	17/10/12 08:00	18/10/12 14:00		
1.1 - analise organizacional	17/10/12 08:00	17/10/12 17:00		
1.2 - entrevista de partes	18/10/12 08:00	18/10/12 14:00		
2 - entregar versões executáveis	18/10/12 14:00	31/10/12 11:00	1	declaração de escopo
2.2 - desenvolver funcionalidades	18/10/12 14:00	30/10/12 09:00		
2.2 - documentar desenvolvimento	30/10/12 09:00	30/10/12 15:00		funcionalidades
2.3 - integrar novas funcionalidades	30/10/12 15:00	31/10/12 11:00		funcionalidades
3 - definir casos de teste	31/10/12 11:00	31/10/12 15:00	2	funcionalidades
3.1 - definir condições de execução	31/10/12 11:00	31/10/12 13:00		
3.2 - Gerar check-list de testes	31/10/12 11:00	31/10/12 14:00		
3.2.1 - mapear atividades	31/10/12 11:00	31/10/12 13:00		
3.2.2 - controlar atividades	31/10/12 11:00	31/10/12 14:00		
3.3 - revisar e formalizar documentação	31/10/12 11:00	31/10/12 15:00		
4 - realizar correções	31/10/12 15:00	05/11/12 16:30	3	chamada de correção;funcionalidades
4.1 - realizar correções	31/10/12 15:00	02/11/12 17:00		funcionalidades
4.2 - alterar documentação	02/11/12 15:00	02/11/12 17:00		documentação
4.3 - abrir nova chamada de correção	05/11/12 15:00	05/11/12 16:30		

Figura 4.12 Cronograma para o modelo SR apresentado na Figura 4.10.

Como se pode observar na Figura 4.12, as atividades contêm limitações entre si, de modo que a atividade de entrega de versões executáveis depende dos requisitos, os casos de teste só podem ser efetuados com novas funcionalidades desenvolvidas e as correções dependem dos testes. No campo de recursos têm-se os recursos necessários para execução das atividades de modo mais pontual.

De modo análogo é possível gerar o cronograma para obtenção de estimativas para o sistema por meio de seus respectivos diagramas, o que será feito no estudo de caso do Capítulo 5.

4.4 Considerações Finais

Neste capítulo foram apresentados os passos e as diretrizes que os constituem, de modo que o processo de gerência de tempo em projetos por meio da modelagem organizacional possa ser guiado e realizado com ou sem documentação.

Os passos e as diretrizes foram pensados não somente para permitir considerar aspectos organizacionais presentes no ambiente do projeto como também para guiá-lo quando não houver disponível a documentação sugerida no processo pelo PMBOK.

Neste trabalho é defendido que o *framework i** permite modelar e representar intencionalidades e relações entre membros integrantes de um projeto na forma de tarefas, recursos e objetivos, o que a torna uma alternativa interessante para servir de base ao processo de gerência de tempo por meio da modelagem organizacional.

O mapeamento realizado por meio do *framework i** permite identificar de maneira rápida e precisa os atores e as atividades a eles associadas nos modelos, o que possibilita uma visão macro do processo garantindo ao gerente e sua equipe um maior entendimento do projeto.

As atividades e tarefas expressas nos modelos SD expressam pacotes de atividade, as quais são detalhadas nos modelos SR correspondentes, de modo que mesmo sem uma estrutura analítica do projeto, é possível mapear as atividades e sequenciá-las, podendo estas tarefas ser realizadas com o auxílio de cada membro da equipe, com relação à área pela qual o mesmo é responsável.

Por meio do estudo do processo de estimativas de duração das atividades percebeu-se a possibilidade de levantar estimativas voltadas apenas para a entrega do produto ou do processo como um todo. Os Modelos gerados do sistema e do projeto possibilitam a definição de estimativas separadamente, além de possibilitarem considerar dependências entre ambos os modelos.

Os *templates* apresentados visam reunir as informações presentes em vários documentos do projeto ou obtidas nas análises em conjunto com a equipe de modo que as informações referentes as atividades estejam reunidas, poupando trabalho de reanálise de documentos.

Seguindo este processo, a atividade de gerência de tempo em projetos tende a ser menos onerosa, seja em decorrência do menor número de documentos e artefatos necessários, seja por centralizar todo o processo sobre os modelos organizacionais.

Por fim a abordagem proposta visa a obtenção do cronograma, o qual é o documento final obtido por meio deste processo.

Capítulo 5

Estudo de caso

Neste Capítulo será aplicado um estudo de caso usando a abordagem proposta sobre um projeto comercial desenvolvido, juntamente com os seus documentos e artefatos que estiverem disponíveis.

Como os passos e as diretrizes já foram explicadas, neste capítulo apenas será dado enfoque para o método usado em cada passo.

Inicialmente será apresentado o problema a ser analisado. Na sequência os passos propostos na nova abordagem serão aplicados e exemplificados, no final do presente capítulo serão apresentadas as considerações finais acerca do estudo de caso.

5.1 Contexto

O projeto a ser desenvolvido deve ser capaz de realizar cadastramento, manutenção ou remoção de cronotacógrafos de veículos de transporte de cargas ou passageiros, cadastro e gerenciamento de selos que serão aplicados aos cronotacógrafos, além de relatórios.

O projeto diz respeito a um módulo que será desenvolvido para uma empresa que já possui um sistema de gerenciamento de frotas de veículos de transporte de cargas ou passageiros.

Há um grande número de documentos disponíveis para análise, o que auxilia no processo por tornar a busca por tais informações mais rápida e precisa. Quando algum documento for útil em algum momento ao longo da aplicação da proposta, o mesmo será referenciado.

As funcionalidades requeridas para o sistema são: definir parâmetro de controle de selagem de cronotacógrafo, cadastro de selos, abertura de ordem de serviço para lançar dados dos selos, impressão de ficha de instalação e manutenção de cronotacógrafo, gerar relatórios de posto de ensaio e manutenção de cronotacógrafos, relatório de selos aplicados, pesquisa de cronotacógrafos aplicados e manutenção de cronotacógrafos em chassis.

A seguir será feita a gerencia de tempo para este projeto, seguindo os passos propostos.

5.2 Aplicação da proposta

5.2.1 Passo 1: gerar os modelos de dependências estratégicas SD

A partir do plano de abertura de projeto foi possível identificar os atores elencados para desenvolvimento do projeto, e por meio de entrevistas com cada um, as atividades necessárias ao projeto de responsabilidade de cada ator puderam ser mapeadas. Na Figura 5.1 Apresenta-se o modelo SD do projeto com foco nos atores envolvidos.

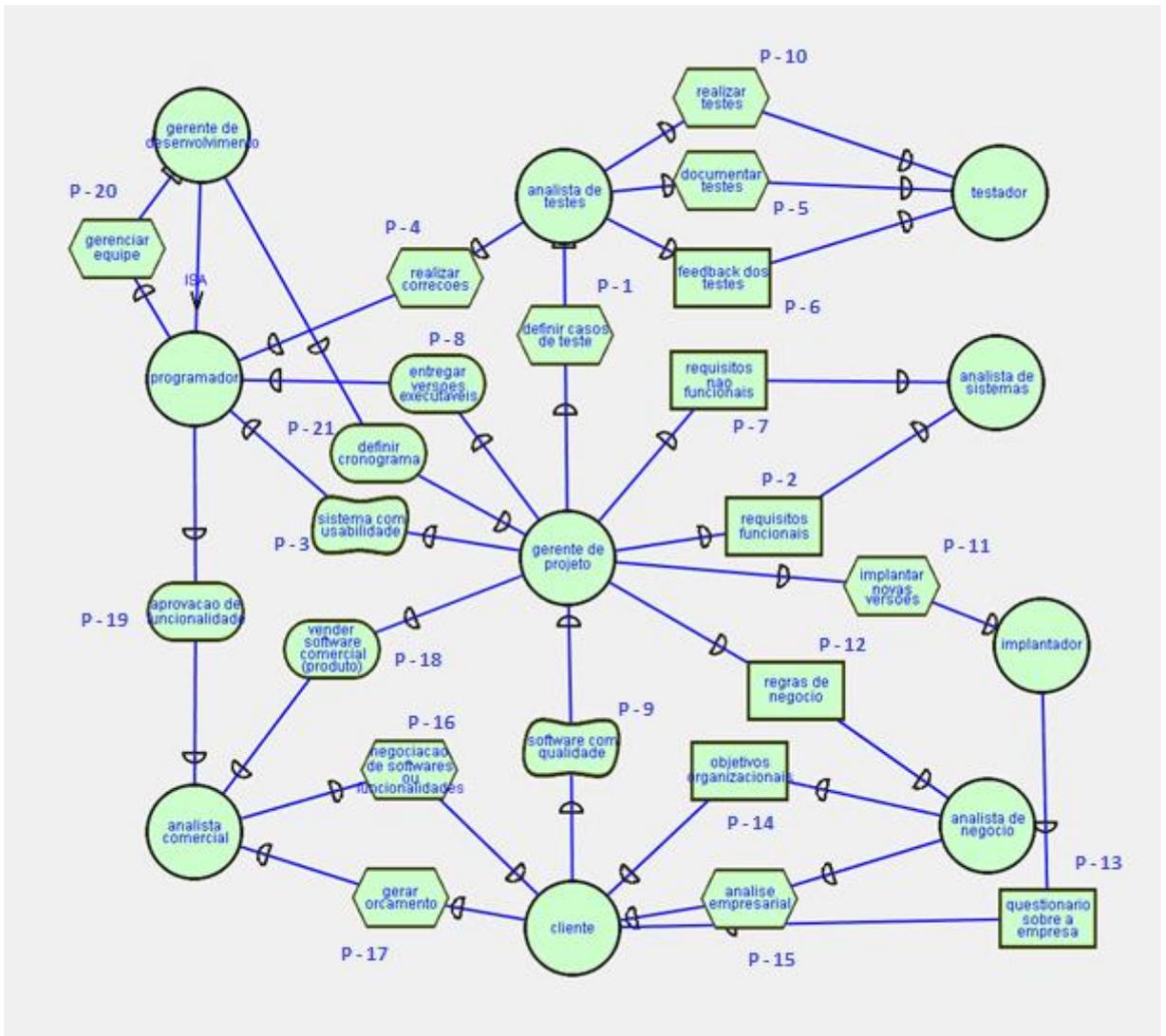


Figura 5.1 Modelo SD do projeto.

A partir do documento de requisitos do projeto foi possível identificar os requisitos que serão mapeados para objetivos no modelo do sistema. Este modelo pode ser visto na Figura 5.2.

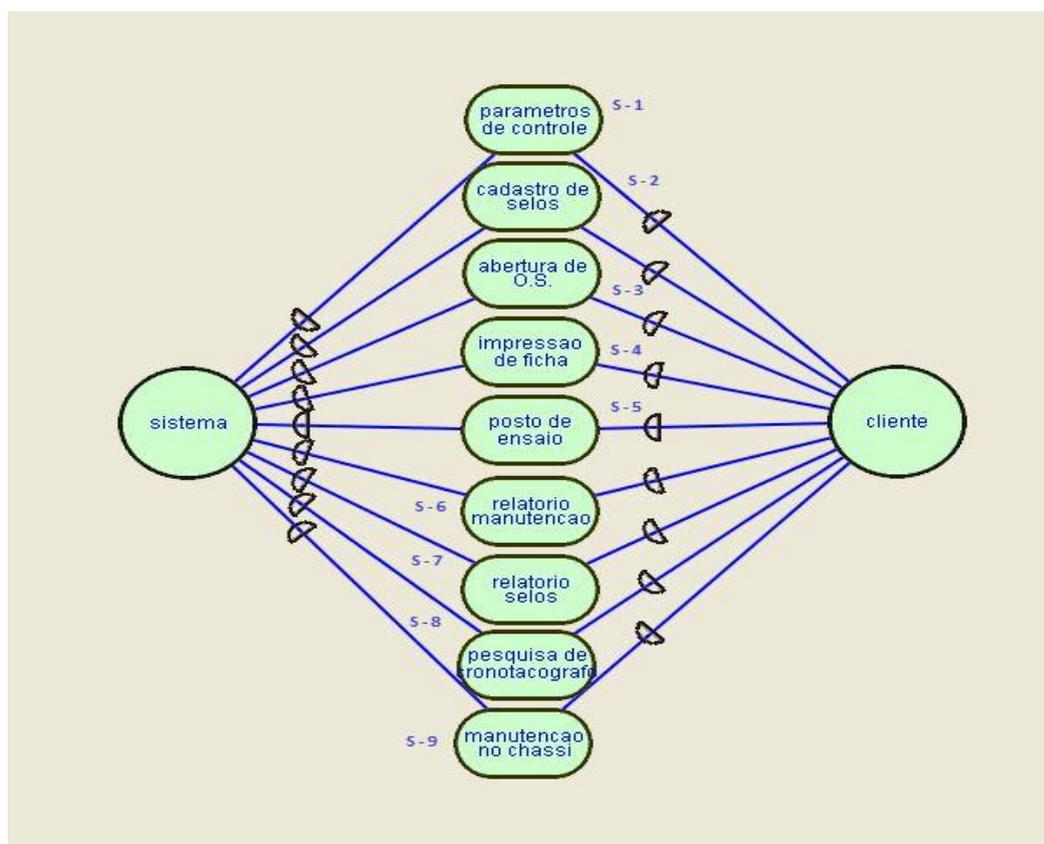


Figura 5.2 Modelo SD do sistema

5.2.2 Passo 2: gerar os modelos de razões estratégicas SR

Partindo dos modelos SD obtidos no passo 1, os atores novamente foram entrevistados para que fosse possível obter um maior nível de detalhamento acerca da conclusão das atividades, sendo que no modelo do projeto novamente cada ator descreveu as atividades de sua alçada, enquanto para o modelo do sistema todos puderam ajudar.

Na Figura 5.3 é exibido o modelo SR do projeto.

O modelo SR do sistema apresenta-se na Figura 5.4

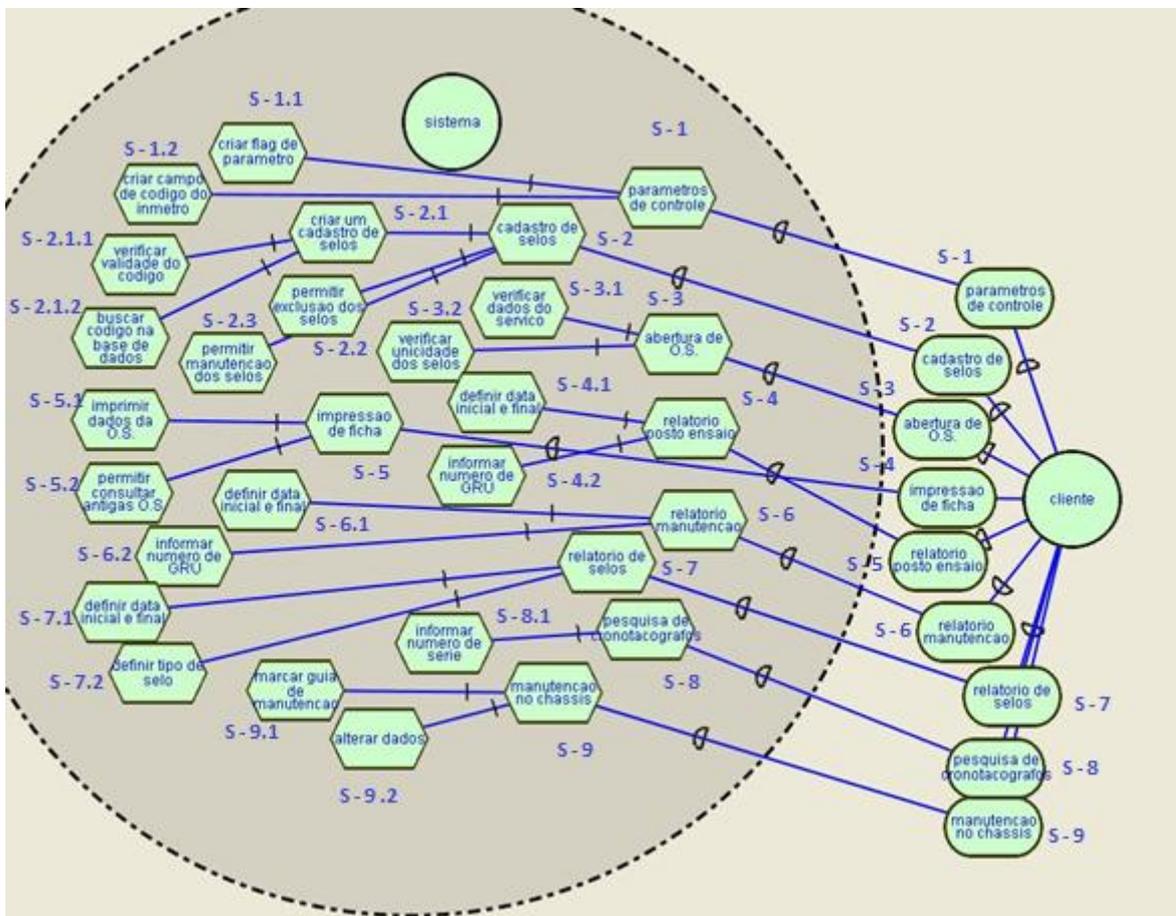


Figura 5.4 Modelo SR do sistema.

5.2.3 Passo 3: listar as atividades

Neste passo, com base nos diagramas são gerados os *templates* para cada atividade, sendo que através da equipe podem ser obtidas informações quando a restrições entre atividades (mais restrições podem ser identificadas no Passo 4).

Entre atividades do projeto e atividades do sistema são gerados noventa e oito *templates* o que torna inviável a exibição de todos aqui, sendo apenas exibido um exemplo. Na Tabela 5.1 pode-se observar a atividade do sistema referente à escolha do tipo de selo, na funcionalidade de relatório de selos, atividade esta limitada pela existência do cadastro de selos (deve ser desenvolvida após).

Identificação da atividade	S - 7.2 – definir tipo de selo
Descrição da atividade	O usuário deve escolher qual o tipo de selo a ser exibido: aplicado, em estoque ou inutilizável.
<i>Depender</i>	Cliente.
<i>Dependee</i>	Sistema.
Atividade antecessora	S - 7.1 - definir data inicial e final
Atividade predecessora	-
Atividades limitantes	S - 2 – cadastro de selos.
Atividades limitadas	-
Estimativa de recursos	-
Estimativa de duração	-

Tabela 5.1: *template* da atividade S – 7.2 – definir tipo de selo.

5.2.4 Passo 4: Estimar recursos para as atividades

Em posse do calendário de recursos e novamente com o auxílio da equipe, os recursos foram alocados para as atividades conforme necessário. Este passo é aconselhável de ser executado em conjunto com a equipe, pois cada ator sabe os recursos que demandarão suas atividades, de modo que este processo exige julgamento e experiência, que preferencialmente devem vir de quem realizará a atividade.

Novamente destaca-se será apresentado um *template* para fins de exemplificação. Na tabela 5.2 encontra-se um *template* exemplificando a aplicação da estimativa de recursos para a atividade 5.2 – alterar documentação, uma sub-atividade da tarefa 5 – documentar testes.

Identificação da atividade	P - 5.2 – alterar documentação
Descrição da atividade	A documentação deve ser alterada de acordo com o que foi modificado na funcionalidade

<i>Depender</i>	Analista de testes.
<i>Dependee</i>	Testador.
Atividade antecessora	P - 5 – documentar testes.
Atividade predecessora	-
Atividades limitantes	-
Atividades limitadas	-
Estimativa de recursos	<ul style="list-style-type: none"> • 1 testador • Documentação da atividade
Estimativa de duração	-

Tabela 5.2: *template* da atividade P – 5.2 – alterar documentação.

5.2.5 Passo 5: sequenciar as atividades

Tendo as atividades já listadas em seus *templates* e com os recursos disponíveis devidamente distribuídos entre as atividades, será realizado o sequenciamento de execução das atividades considerando as restrições impostas para cada atividade.

A sequência com que as atividades serão desenvolvidas serão escalonadas entre as fases do projeto, podendo elas estarem delimitadas por prazos ou marcos, sendo possível com o uso dos *templates* separá-las através de seus índices.

5.2.6 Passo 6: Estimar duração para as atividades

As estimativas de duração, como dito anteriormente, podem ser feitas via técnicas baseadas em julgamento ou parâmetros. No passo 6 será gerado o cronograma com base na estimativa de tempo obtida por pontos por casos de uso do sistema, mas para fins de demonstração no *template* uma atividade terá sua estimativa feita pontualmente via PERT.

Foram obtidas estimativas pontuais para o projeto com as planilhas de estimativas disponíveis, sendo exibida na Tabela 5.3 a atividade de cadastro de selos após a estimativa.

Identificação da atividade	S - 2 – cadastro de selos
Descrição da atividade	O sistema deve permitir cadastrar um ou uma série de selos, sendo necessária a verificação de sua validade. Também deve ser possível alterar e excluir selos.
<i>Depender</i>	Cliente.
<i>Dependee</i>	Sistema.
Atividade antecessora	-
Atividade predecessora	Criar um cadastro de selos
Atividades limitantes	-
Atividades limitadas	S – 7 - relatório de selos
Estimativa de recursos	-
Estimativa de duração (via PERT)	15 horas

Tabela 5.3: *template* da atividade S – 2 – cadastro de selos.

5.2.7 Passo 7: gerar o cronograma do projeto

Com base na estimativa de 63 horas (obtida na seção 4.3.5, via pontos por casos de uso) e considerando o diagrama do sistema a ser desenvolvido, foi gerado o cronograma exibido na Figura 5.5.

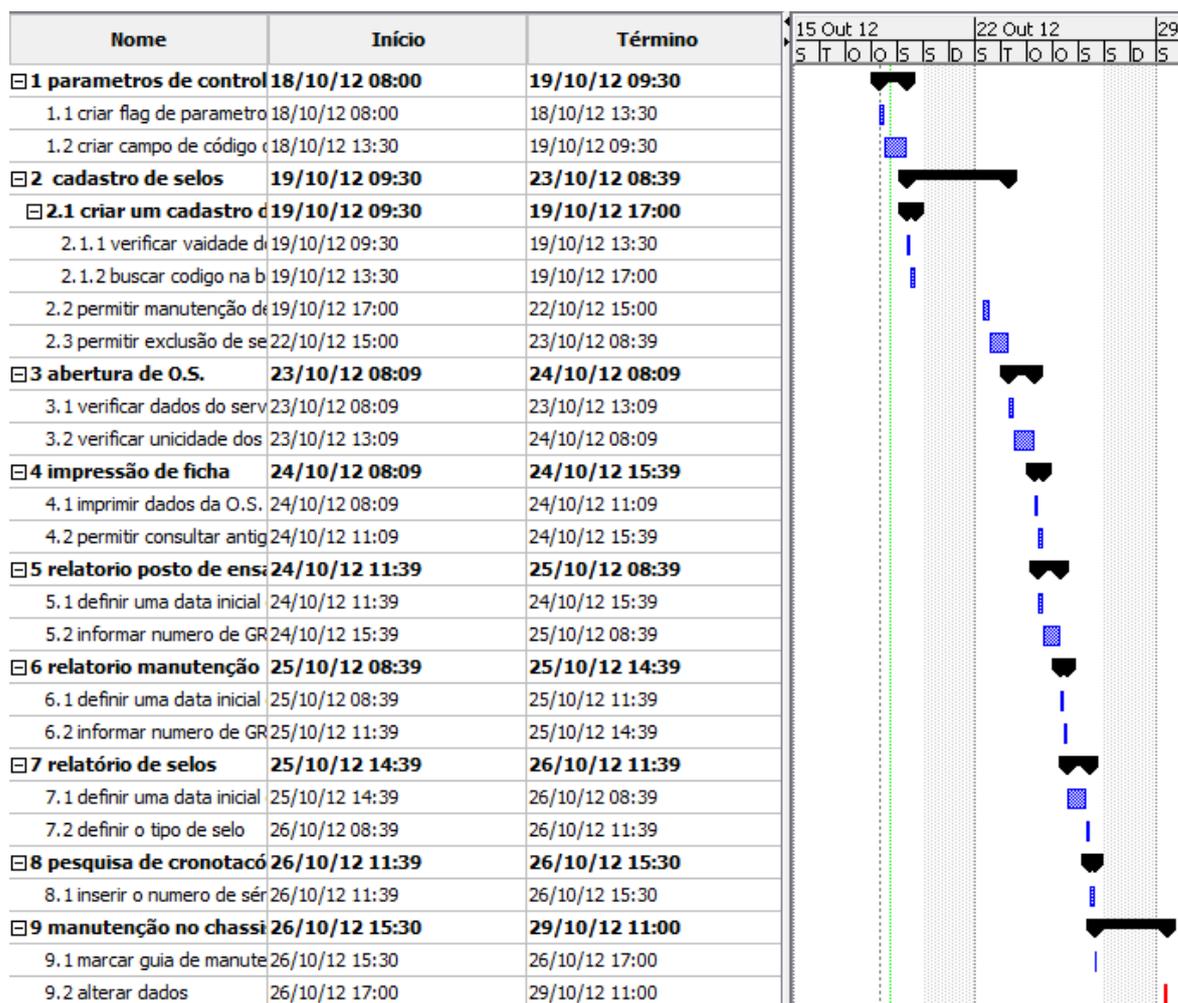


Figura 5.5 Cronograma do sistema a ser desenvolvido.

5.3 Considerações Finais

Neste capítulo a proposta foi aplicada a um estudo de caso de um projeto desenvolvido para o âmbito comercial, o qual contava inclusive com artefatos como plano de projeto e planilhas de estimativas, o que reduziu os esforços necessários para obtenção das informações referentes ao projeto.

As informações referentes aos atores e as atividades necessárias para concluir o sistema foram obtidas no plano de projeto e documento de requisitos respectivamente, sendo necessária a consulta a toda a equipe apenas para obter as atividades necessárias ao projeto.

As atividades foram listadas e sequenciadas fazendo-se a análise dos marcos de controle previstos no projeto, além de cada *depende* ter auxiliado no processo de análise de suas atividades.

Os processos de estimativa foram feitos também com o auxílio de documentação, sendo disponibilizado o calendário de recursos para auxiliar na estimativa de recursos, e uma planilha de estimativas de tempo pontual para as atividades que foram de auxílio nas estimativas de duração.

Na obtenção do cronograma foi usada a estimativa obtida com a técnica de pontos por caso de uso para obtenção do tempo de desenvolvimento das atividades, que de acordo com a sequência previamente definida foram inseridas no cronograma, ajustando-se as datas de acordo com as estimativas pontuais.

Capítulo 6

Considerações finais

Neste Capítulo será feita uma breve análise da proposta, com relação a abordagem tradicional de gerência, a fim de compará-las e listar as principais vantagens e diferenças no uso da modelagem organizacional.

Na seção 6.1 serão apresentados e defendidos pontos positivos no uso da proposta para gerência de tempo. Na seção 6.2 serão abordados tópicos cujos estudos no futuro são de grande interesse para validação e aprimoramento da proposta. Por fim na seção 6.3 serão realizadas as considerações desse capítulo.

6.1 Vantagens do uso da proposta

O processo de gerência de tempo considerando características organizacionais apresenta vantagens sobre a abordagem tradicional, tanto por permitir identificar tais características, quando por oferecer um maior número de possibilidades em termos de documentação para o gerente de projeto realizar sua análise.

Listam-se na sequencia algumas vantagens observadas através do uso da proposta.

6.1.1 Validação de requisitos

Os diagramas juntamente com os *templates* gerados permitem representar os requisitos do software em termos de dependências, cuja satisfação permite conseqüentemente satisfazer os requisitos, ou seja, validá-los.

Os requisitos podem ser satisfeitos em iterações ou sprints, e sua validação pode ser feita através da realização das atividades referentes aos mesmos.

6.1.2 Base de conhecimento

Os diagramas e *templates* gerados, principalmente para funcionalidades genéricas e presentes em muitos sistemas poderão com o tempo gerar uma base de dados para futuros projetos, permitindo não só estimativas mais realistas em termos de recursos e tempo individualmente quanto para módulos do projeto.

Tal base de dados é de grande utilidade em casos de estimativas análogas, visto que seu uso se torna vantajoso na medida em que cresce a base de dados da empresa.

6.1.3 Rastreabilidade

Por meio dos *templates* torna-se possível mapear o desenvolvimento das atividades, além de permitir monitorar o desenvolvimento dos requisitos, que pode ser mensurado através das fases do projeto ou marcos especificados.

6.2 Contribuições futuras

Dentre as várias possibilidades de expansão da proposta, algumas são aqui listadas para futuro desenvolvimento.

6.2.1 Experimento

A fim de verificar a eficiência da proposta, pode ser realizado um experimento que permita mensurar. Com relação a abordagem tradicional, ainda não foi realizada nenhuma comparação que permita verificar o ganho de tempo no processo.

A consideração de aspectos organizacionais possui outras vantagens além do possível ganho de tempo (dentre as vantagens, algumas estão listadas na seção 6.1). Um experimento em condições controladas permitiria verificar tais vantagens.

6.2.2 Ferramenta

Uma contribuição de grande impacto para a proposta seria o desenvolvimento de um software que permitisse realizar todo o processo, gerando os *templates* a partir dos diagramas, simplificando o processo.

Uma ferramenta possibilitaria ainda a integração com outras áreas da gerência de projetos, tais como gerência de riscos ou de mudanças.

6.3 Considerações finais

Com a proposta de gerência de tempo via modelagem organizacional espera-se que o processo seja simplificado e realizado de modo tão ou mais ágil que o método tradicional proposto pelo PMBOK[1]. Outras contribuições alcançadas dizem respeito a validação de requisitos, gerenciamento de dependências, rastreabilidade dentre outras.

A automatização do método proposto certamente reduziria seu tempo de aplicação, além de possibilitar alterações seja pelos diagramas, seja pelos *templates*.

A estrutura da proposta, organizada no formato de passos e diretrizes é bem objetiva e direta, de modo facilitar a aplicação da mesma. A relação entre passos e diretrizes pode ser observada na Figura 6.1

Passo 1..... diretriz 1
diretriz 2
diretriz 3

passo 2..... diretriz 4
diretriz 5

passo 3..... diretriz 6

passo 4..... diretriz 7

passo 5..... diretriz 8

passo 6..... diretriz 9
diretriz 10

passo 7..... diretriz 11

Figura 6.1 Relação entre diretrizes e passos.

Glossário

Cronotacógrafo

Instrumento que registra velocidade, tempo e distância percorrida por um veículo em seu deslocamento. Os veículos de carga com peso bruto acima de 4.536 quilogramas e os veículos de passageiros com mais de 10 lugares são obrigados pelo Código de Trânsito Brasileiro a possuir cronotacógrafo.

Framework

Abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.

Referências Bibliográficas

[1] PMBOK. Um Guia do Conhecimento em Gerenciamento de Projetos. 4. ed. Global Standard, Pennsylvania . EUA, 2008.

[2] PMI. *Project management institute*. Disponível em: <http://www.pmi.org/>. Último acesso em: 10 de Julho de 2012.

[3] ALENCAR, Fernanda Maria Ribeiro De; CASTRO Jaelson Freire Brelaz de. Mapeando a modelagem organizacional em especificações precisas. Tese (Doutorado em Ciência da Computação) Universidade Federal de Pernambuco, Recife, 1999.

[4] PÁDUA, Silvia Inês Dallavalle; CAZARINI, Edson Walmir; Modelagem Organizacional para capturar os requisitos organizacionais. USP- EESCEscola de Engenharia de São Carlos. Anais, SIMPEPVIII, 2002.

[5] E. Yu, L. Liu, and Y. Li. *Modelling Strategic Actor Relationship to Support Intellectual Property Management*. 20th International Conference on Conceptual Modelling (ER-2001), 2001.

[6] PATAH, L. A. Alinhamento estratégico de estrutura organizacional de projetos: uma análise de múltiplos casos. Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo, São Paulo, 2004 . Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3136/tde-26062004-161221/> Último acesso em 27/09/2012.

[7] SOMMERVILLE, Ian. Engenharia de Software. 6ed. São Paulo : Addison Wesley, 2003.

[8] VARGAS, Ricardo V. Gerenciamento de projetos: Estabelecendo diferenciais competitivos. 5.ed. Rio de Janeiro : Brasport, 2003, p1. Disponível em

<http://pt.scribd.com/doc/6935874/GERENCIAMENTO-DE-PROJETOS-RICARDO-VIANA-VARGAS>. Último acesso em 27/09/2012

[9] SEI. *Software engineering institute*. Disponível em: <http://www.sei.cmu.edu/>. Último acesso em: 10 de Julho de 2012.

[10] ISO, International Standart Organization; ISO/IEC 9126 – Software engineering – Product Quality, 2001.

[11] RUSSO, R. F. S. M. Tendência empreendedora do gerente de projeto: importância para o sucesso dos projetos. Dissertação de Mestrado, Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, São Paulo. 2007. Disponível em: <http://www.teses.usp.br/teses/disponiveis/12/12139/tde-17102007-214841/>. Último acesso em: 27/09/2012.

[12] PRESSMAN, Roger S. Engenharia de Software. Tradução José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.

[13] PÁDUA, Sílvia Inês Dallavalle de; CAZARINI, Edson Walmir; INAMASU, Ricardo Yassushi. Modelagem organizacional: captura dos requisitos organizacionais no desenvolvimento de software de informação. *Revista gestão e produção*. São Carlos, v.11, n.2, p.197-209, ago. 2004.

[14] GRANDO, F. L. Avaliando técnicas de modelagem organizacional no contexto de desenvolvimento de software computacionais: uma abordagem empírica – Colegiado de Ciências da Computação – Universidade Estadual do Oeste do Paraná, Cascavel – PR, Novembro 2010.

[15] BUBENKO JR., J.A.; KIRIKOVA, M. Enterprise modelling: improving the quality of requirements specification. In: IRIS-17 INFORMATION SYSTEMS RESEARCH SEMINAR IN SCANDINAVA, Oulu, Proceedings, s.n.t. 1994.

[16] SANTANDER, V. F. A. Integrando a Modelagem Organizacional com Modelagem Funcional. Tese (Tese Doutorado), Engenharia de Software – Centro de Informática – Universidade Federal de Pernambuco, Recife, PE, 2002, p 7.

[17] BRISCHKE, M. Melhorando a Ferramenta JGOOSE – Colegiado de Ciências da Computação – Universidade Estadual do Oeste do Paraná, Cascavel – PR, Novembro 2011.

[18] GRAU, G; Yu, E; Horkoff, J; Abdulhadi, S.: *i* Guide*. Disponível em: <http://istar.rwth-aachen.de/tiki-index.php?page=i%2A+Guide>, 2008. Último acesso em: 10 de Julho de 2012.

[19] VICENTE, A. ABE. JGOOSE: Uma ferramenta de Engenharia de Requisitos para a Integração da Modelagem Organizacional i* com a Modelagem Funcional de Casos de Uso UML. Trabalho (Trabalho de conclusão de graduação) –Unioeste –Universidade Estadual do Oeste do Paraná, Cascavel–PR. Dezembro de 2006.

[20] KARNER, G. Use Case Points: resource estimation for Objectory projects. Objective Systems SF AB (copyright owned by Rational/IBM), 1993. Disponível em: <http://projetos.unioeste.br/moodle/institucional/file.php/130/UseCasePoints.zip>, Último acesso em: Outubro de 2012.

[21] WILLIAMS, T.M. What are PERT estimates?, *Journal of the Operational Research Society*, 46 (12), 1995, pp.

[22] AGUIAR, M: Estimando os Projetos com COCOMO II. *TIMétricas*, 2004. Disponível em http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ES_I_2010_2/COCOMO_II.pdf Último acesso em: Novembro 2012.