



**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  
planejamento de qualidade de software proposto pelo  
guia PMBOK.**

*Rodrigo Trage*

**CASCABEL**

**2014**

**RODRIGO TRAGE**

**INTEGRANDO O FRAMEWORK I\* AO PROCESSO DE  
PLANEJAMENTO DE QUALIDADE DE SOFTWARE PROPOSTO  
PELO GUIA PMBOK.**

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

CASCVEL

2014

**RODRIGO TRAGE**

**INTEGRANDO O FRAMEWORK I\* AO PROCESSO DE PLANEJAMENTO DE  
QUALIDADE DE SOFTWARE PROPOSTO PELO GUIA PMBOK**

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. Dr. Ivonei Freitas da Silva

Colegiado de Ciência da Computação,  
UNIOESTE.

---

Prof. Me. Elder Elisandro Schemberger

UTFPR – Toledo.

---

Prof. Me. Sidgley Camargo de Andrade

UTFPR – Toledo.

Cascavel, 14 de Outubro de 2014.

## **EPÍGRAFE**

O silêncio está tão repleto de sabedoria e de espírito em potência como o mármore não talhado é rico em escultura. Aldous Huxley.

## **AGRADECIMENTOS**

Agradeço a meus pais Ademir Trage e Salete Carvalho Trage, por todo o apoio que me deram, pelos valores morais e educação que me ensinaram, concedendo-me a oportunidade da desalienação e lutar pelo ideal de obter o conhecimento da graduação. Agradeço também a todos meus amigos que estiveram presentes na busca deste ideal, mas em especial a Felipe Augusto Henn, Bruno Eduardo Soares, Bruno Borguesan, Mauricio Begnini, que sempre estiveram próximos tanto nos bons quanto nos maus momentos, fossem eles estudando ou festejando, sempre podia-se contar com eles e agradeço a Kathia Regina Cantelli que sempre esteve ao meu lado me dando forças para continuar, incentivando e apoiando nos momentos de desânimo. Por fim agradeço a todos os meus professores pelo conhecimento adquirido.

# LISTA DE FIGURAS

Figura 1: Princípio da abordagem .....	6
Figura 2: visão geral do planejamento da qualidade (PMBOK, 2008).....	13
Figura 3: Matriz L-Shaped (ROSE, 2005).....	16
Figura 4: Os principais componentes do framework i* (BRISCHKE, 2011).....	22
Figura 5: Associações entre atores (GRAU et al., 2008).....	24
Figura 6: Exemplo de diagrama SR (Henn, 2012).....	25
Figura 7: Ligações de contribuição (GRAU et al., 2008).....	26
Figura 8. Diagrama de atividades da proposta.....	31
Figura 9: Atores identificados para o SD do projeto.....	35
Figura 10: Representação de dependência do tipo objetivo.....	36
Figura 11: Representação de dependência do tipo objetivo-soft.....	36
Figura 12: Representação de dependência do tipo tarefa.....	37
Figura 13: Representação de dependência do tipo recurso.....	37
Figura 14: Modelo SD do projeto gerado no passo 1.....	38
Figura 15: Modelo SD do sistema para requisitos do produto.....	38
Figura 16: Modelo SD do sistema para requisitos de qualidade.....	39
Figura 17: Modelo SR do projeto.....	41
Figura 18: Modelo SR dos requisitos do software.....	41
Figura 19: Modelo SR dos requisitos de qualidade do sistema.....	42
Figura 20: Obtendo os stakeholders para priorização a partir do modelo SD.....	42
Figura 21a: Obtendo os requisitos de <i>software</i> para priorização a partir do modelo SD.....	44
Figura 21b: Obtendo os requisitos de qualidade para priorização a partir do modelo SD.....	44
Figura 22: Processo para a obtenção da matriz de priorização conjunta.....	48
Figura 23: Modelo SD do sistema.....	57
Figura 24: Modelo SD do projeto.....	58
Figura 25: Modelo SD dos requisitos de qualidade do sistema.....	59
Figura 26: Modelo SR dos requisitos de software sistema.....	62
Figura 27: Modelo SR do projeto.....	63
Figura 28: Modelo SR dos requisitos de qualidade do sistema.....	64

# LISTA DE TABELAS

Tabela 1: Exemplo de matriz de priorização de stakeholders.....	43
Tabela 2: Matriz L-shaped para priorização de requisitos de software.....	45
Tabela 2.1: Matriz L-shaped para priorização de requisitos de qualidade.....	45
Tabela 3: L-shaped para requisitos de qualidade na visão do Orientador 1.....	46
Tabela 3.1: L-shaped para requisitos de qualidade na visão do Orientador 2.....	46
Tabela 3.2: L-shaped para requisitos de qualidade na visão do Orientando.....	47
Tabela 4: Matriz de priorização conjunta para requisitos de qualidade.....	48
Tabela 4.1: Métrica de qualidade.....	49
Tabela 4.2: Exemplo de documento de conformidade criado com base no diagrama SR do projeto.....	51
Tabela 4.3: <i>Checklist</i> de qualidade.....	51
Tabela 5.1: Matriz de priorização de stakeholders.....	65
Tabela 5.2 Matriz de priorização de requisitos de software.....	67
Tabela 5.3 Matriz de priorização de requisitos de qualidade.....	68
Tabela 5.4 priorização dos requisitos de software na visão do cliente.....	71
Tabela 5.5 priorização dos requisitos de software na visão do cliente minoritário.....	72
Tabela 5.6 priorização dos requisitos de software na visão do usuário.....	73
Tabela 5.7 priorização dos requisitos da qualidade na visão do cliente.....	75
Tabela 5.8 priorização dos requisitos da qualidade na visão do cliente minoritário.....	76
Tabela 5.9 priorização dos requisitos da qualidade na visão do cliente usuário.....	77
Tabela 5.10 Matriz conjunta de priorização dos requisitos de software.....	78
Tabela 5.11 Matriz conjunta de priorização dos requisitos da qualidade.....	79
Tabela 5.12: Métricas de qualidade e critérios de aceitação.....	80
Tabela 5.13: Documento de conformidade do processo.....	82
Tabela 5.14: <i>Checklist</i> de qualidade.....	84
Tabela 6: Resultado das questões conforme as métricas.....	86

# **LISTA DE ABREVIATURAS E SIGLAS**

EAP Estrutura Analítica do Projeto

PMBOK Project Management Body of Knowledge

PMI Project Management Institute

SD Modelo de Dependências Estratégicas

SR Modelo de Razoes Estratégicas

# Sumário

Lista de Figuras .....	vi
Lista de TABELAS .....	vii
Lista de Abreviaturas e Siglas .....	viii
Resumo.....	xii
Capítulo 1 .....	1
1.1 CONTEXTO .....	1
1.2 Motivações.....	4
1.3 Objetivos.....	5
1.3.1 Objetivos Gerais.....	6
1.3.2 Objetivos Específicos.....	7
Capítulo 2.....	8
2.1 Qualidade de Software.....	8
2.1.1 Conceitos Básicos sobre qualidade de software.....	8
2.1.2 Modelos de avaliação de qualidade/maturidade.....	10
2.2 Gerenciamento de projetos e sua importância.....	12
2.2.1 Planejar Qaulidade.....	12
2.2.1.1 Entradas para planejar a qualidade.....	13
2.2.1.2 Ferramentas e técnicas para planejar a qualidade.....	14
2.2.1.3 Saídas do planejamento da qualidade.....	17
2.3 Considerações finais.....	17
Capítulo 3.....	19
O Framework i* .....	19
3.1 A Modelagem Organizacional e o Planejamento de Qualidade de Projetos de Software... 19	
3.2 Uma Visão Sobre o Framework i* .....	21
3.2.1 O Modelo de Dependências Estratégicas (SD).....	22
3.2.2 O Modelo de Razões Estratégicas (SR).....	24
3.3 Considerações Finais.....	27
Capítulo 4.....	28
Integrando o Framework i* ao Planejamento de Qualidade.....	28
4.1 Introdução.....	28

4.2 Proposta.....	30
4.3 Diretrizes.....	34
4.3.1 Passo 1: criar os modelos de dependências estratégicas.....	34
4.3.1.1 Passo 1 - Diretriz 1: identificação dos atores participantes do projeto e stakeholders.....	34
4.3.1.2 Passo 1 - Diretriz 2: identificação de dependências entre atores.....	36
4.3.2 Passo 2: criar os modelos de razões estratégicas.....	39
4.3.2.1 Passo 2– Diretriz 3: geração dos modelos SR com foco nos atores envolvidos no processo.....	40
4.3.3 Passo 3: criar a matriz de priorização dos <i>stakeholders</i> .....	42
4.3.4 Passo 4: criar a matriz de priorização de requisitos.....	43
4.3.5: Passo 5: aplicar a matriz de priorização de requisitos aos atores.....	45
4.3.6 Passo 6: gerar matriz conjunta de priorização de requisitos com a influência de cada ator.....	47
4.3.7 Passo 7: identificar as métricas de qualidade.....	49
4.3.8 Passo 8: validação dos artefatos.....	49
4.3.9 Passo 9: verificação da priorização dos requisitos de <i>software</i> e qualidade.....	50
4.3.10 Passo 10: gerar documento de conformidade.....	50
4.3.11 Passo 11: gerar checklist de qualidade.....	51
4.4 Considerações Finais.....	52
Capítulo 5.....	53
Avaliação da integração do <i>framework</i> i* ao planejamento da qualidade.....	53
5.1 Contexto.....	53
5.2 Utilizando o GQM para orientar a realização do estudo de caso.....	53
5.3 Aplicação da proposta.....	55
5.3.1 Passo 1: criar os modelos de dependências estratégicas.....	55
5.3.2 Passo 2: criar os modelos de razões estratégicas.....	60
5.3.3 criar a matriz de priorização dos <i>stakeholders</i> .....	63
5.4.4 criar a matriz de priorização de requisitos.....	64
5.4.5 aplicar a matriz de priorização de requisitos aos atores.....	68
5.4.6 gerar matriz conjunta de priorização de requisitos com a influência de cada ator.....	77
5.4.7 identificar as métricas de qualidade.....	78

5.4.8 validação dos artefatos.....	80
5.4.9 verificação da priorização dos requisitos.....	80
5.4.10 gerar documento de conformidade.....	80
5.4.11 gerar <i>checklist</i> de qualidade.....	82
5.4 Análise da abordagem.....	83
5.5 Discussão.....	84
5.6 Considerações finais.....	88
Capítulo 6.....	90
Considerações finais.....	90
6.1 Vantagens do uso da proposta.....	90
6.2 Contribuições futuras.....	91
6.2.1 Ferramentas.....	91
6.3 Considerações finais.....	91
Referências Bibliográficas.....	93

# RESUMO

O planejamento da qualidade é a primeira dentre três atividades pertencentes a gerência de qualidade, esta atividade possui saídas que servirão de entrada para as outras duas atividades, por isso atribui-se demasiada importância a esta atividade. E considerando que a mesma possui dilemas em sua abordagem padrão no que se refere a elaboração e utilização de matrizes de priorização, tendo em mente que essas matrizes podem influenciar no resultado das saídas métricas de qualidade e *checklist* de qualidade da atividade do planejamento da qualidade, observou-se a oportunidade de integrar a modelagem organizacional no intuito de apoiar o processo de elaboração destas matrizes, a avaliação dos elementos a serem priorizados e na obtenção do documento de conformidade, das métricas de qualidade e do *checklist* de qualidade. Então neste cenário, a modelagem organizacional pode ser de grande auxílio no desenvolvimento da proposta haja vista os benefícios tais como: A melhora no discernimento do ambiente organizacional e do contexto do *software* encomendado; Melhoria no processo de obtenção de dados junto ao usuário e na modelagem de metas estratégicas e possíveis recursos para satisfação das mesmas; e a possibilidade de representações de relacionamentos existentes entre os atores por meio de diagramas. Neste trabalho será utilizado o *framework* de modelagem organizacional *i\** com integração ao processo de planejamento da qualidade proposto pelo modelo PMBOK. Tem-se como intuito obter-se as saídas documento de conformidade, métricas de qualidade e *checklist* de qualidade de maneira mais fiel e rápida, a fim de reduzir o número de documentos necessários. Contudo como ambição chave deste trabalho, espera-se alcançar um melhor nível de entendimento do escopo do projeto no intuito de melhorar o processo de planejar a qualidade. Também há a perspectiva de que esta abordagem facilite a elaboração e aplicação das matrizes de priorização, uma vez que as partes interessadas no projeto poderão utilizar os diagramas gerados por meio do *framework i\**, como alicerces para o processo.

**Palavras-chave:** Planejamento da qualidade, *framework i\**, modelagem organizacional, PMBOK

# Capítulo 1

## Introdução

### 1.1 CONTEXTO

A área de gestão de projetos de *software* fortaleceu-se ao longo dos anos devido a necessidade de garantir a qualidade e o sucesso dos projetos de *software*. Ao estabelecer uma ligação entre os conceitos clássicos da área de administração - planejar, coordenar, organizar e controlar - com elementos específicos da área de *software*, a área de gestão de projetos de *software* caracteriza-se como multidisciplinar e envolve aspectos tanto organizacionais quanto técnicos (ANDRADE; TAIT, 2012).

Por meio do gerenciamento de projetos de outras áreas já maduras, como administração, engenharias, entre outras, surgiu a gestão de projetos de *software*, cujas técnicas foram adaptadas em uma sequência de etapas que envolvem os aspectos e questões do processo de desenvolvimento de um projeto de *software*, desde o estabelecimento inicial dos conceitos até o artefato de programação (TEIXEIRA; CUKIERMAN, 2007).

Nesse contexto, a gestão de projetos difere das outras áreas por possuir características e fatores que tornam ainda mais difíceis as atividades do gerenciamento, tais como designação específica dos recursos humanos, mudanças e/ou avanços tecnológicos, habilidades multidisciplinares e principalmente dificuldades de mensurar o software (ENAMI; *et al.*, 2006). No intuito de diminuir os impactos desses fatores faz-se necessário optar por modelos e técnicas específicas para gerenciar projetos de software.

A gerência de projetos de *software* visa cumprir os requisitos estipulados para o sistema dentro do prazo definido levando em conta fatores como orçamento, manutenções e alterações no projeto com o objetivo de criar um produto documentado e com qualidade.

Um dos padrões para a gerência de projetos é definido pelo *Project Management Book of Knowledge* (PMBOK), sendo que esta base congrega o conjunto de conhecimentos necessários para gerenciar projetos de softwares e abrange nove áreas desta gestão, sendo elas: gerenciamento de integração do projeto, gerenciamento de escopo do projeto, gerenciamento de tempo do projeto, gerenciamento de custos 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 aquisições do projeto. Este trabalho terá foco no auxílio a obtenção de algumas saídas da área de gerenciamento de qualidade.

Para o PMBOK (PMI, 2008) o objetivo do gerenciamento da qualidade do projeto é determinar as políticas de qualidade, os objetivos e as responsabilidades, para que com esse conjunto de características possa ser atingido um grau de satisfação das necessidades para as quais foi empreendido.

Essa gestão é dividida em três fases principais:

- Planejar a qualidade: O processo de identificar os requisitos e/ou padrões de qualidade do projeto e do produto, bem como documentar de que modo o projeto demonstrará a conformidade.
- Realizar a garantia da qualidade: O processo de auditoria dos requisitos de qualidade e dos resultados das medições de controle de qualidade para garantir que sejam usados os padrões de qualidade e as definições operacionais apropriadas.
- Realizar o controle de qualidade: O processo de monitoramento e registro dos resultados da execução das atividades de qualidade para avaliar o desempenho e recomendar as mudanças necessárias.

O presente trabalho aborda a primeira fase da gerência da qualidade. Nesta fase objetiva-se gerar os seguintes artefatos:

1. Plano de gerenciamento da qualidade ou documento de conformidade: este plano descreve como a equipe de gerenciamento de projetos implementará a política de qualidade. Este plano pode ser formal ou informal. O estilo e os detalhes são determinados pelos requisitos do projeto (PMBOK, 2008).

2. Métricas da qualidade: são definições que descrevem como o processo de controle da qualidade irá medir um atributo do projeto ou do produto (PMBOK, 2008).
3. Listas de verificação da qualidade: é uma ferramenta estruturada para verificar se um conjunto de etapas necessárias foi executado.

Mais especificamente propõe-se apoiar a obtenção dos artefatos 1,2 e 3 por meio do uso de modelos organizacionais.

A modelagem organizacional permite uma compreensão mais profunda do domínio da aplicação, abandonando a visão simplesmente funcional do sistema (CHICHINELLI, 2002).

O framework de modelagem adotado é o *i\** (*istar*) que permite descrever aspectos de intencionalidade e motivações envolvendo atores em um ambiente organizacional (YU; LIU; LI, 2001). Esse *framework* permite modelar os requisitos funcionais e não funcionais do sistema, as tarefas, os objetivos, os recursos, as dependências e as pessoas envolvidas através de diagramas. Com isso pode-se obter uma visão organizacional através dos modelos de diagramas associados à proposta original do *i\** - Strategic Dependency (SD, descreve os relacionamentos de dependência entre os vários atores no contexto organizacional) e o modelo Strategic Rationale (SR, tem como objetivo representar as estratégias internas de cada ator).

Contudo, observando as saídas do planejamento de qualidade já descritas, percebe-se uma grande dificuldade para gerá-las. Isto ocorre porque muitas vezes o processo torna-se intuitivo e a equipe do projeto baseia-se no ‘acho que’ para tentar responder algumas questões relacionadas às visões do cliente sobre o produto desejado (ROSE, 2005). Uma ferramenta utilizada para auxiliar a atividade de planejamento da qualidade é a matriz de priorização de requisitos de software, qualidade e *stakeholders*, a qual fornece uma forma de classificar um conjunto variado de problemas (PMBOK, 2008), como os requisitos do produto e os envolvidos na elaboração dele. Essa matriz tem importância para as saídas de métricas de qualidade e listas de verificação de qualidade, pois a utilização da mesma indica a posterior existência ou ausência de alguns elementos nessas saídas, como por exemplo quando tem-se recursos suficientes para implementar apenas um requisito de qualidade, então o elemento que constará nas saídas de métricas de qualidade e *checklist* de qualidade será apenas o elemento que tiver maior índice de prioridade. A relevância da priorização exige um processo rigoroso e disciplinado, o que pode ser um desafio para a equipe de projeto no momento de sua elaboração e aplicação (ROSE, 2005). E a não

aplicação total ou coerente da priorização de requisitos pode levar a um planejamento de qualidade com estados inconsistentes (ROSE, 2005).

Considerando este contexto, notou-se que a utilização da modelagem organizacional pode auxiliar a equipe do projeto na elaboração das matrizes de priorização de requisitos de *software* e qualidade e de *stakeholders* e na aplicação destas considerando a visão dos *stakeholders* para a priorização dos requisitos e também visando apoiar-los com a modelagem organizacional para que estes possam melhor avaliar os elementos dessas matrizes. E desta forma, pode-se tentar elevar a qualidade do *software*, além de apoiar na construção das saídas já citadas da atividade de planejar a qualidade.

## 1.2 Motivações

Levando em conta a preocupação levantada por (ROSE, 2005), deve-se ter em mente que um projeto é repleto de muitos atores e que provavelmente as técnicas de *elicitação* identificarão uma quantidade de requisitos numericamente maior que a de atores. Entretanto, devido às restrições de tempo e orçamento, pode a ser difícil atender a todos os requisitos identificados para um sistema. Para (ALLEN *et al.*, 2008) a priorização de requisitos auxilia a definir quais devem ser implementados primeiro. Na priorização destes requisitos, pode-se associar o peso da influência que um *stakeholder* possui, a fim de uma priorização mais completa e eficaz. Sendo assim deve-se também priorizar os *stakeholders*. A priorização tanto dos requisitos quanto dos *stakeholders* é um passo importante para um planejamento de qualidade efetivo (ROSE, 2005). Considerando então essas priorizações, há ainda outra preocupação a ser considerada, como relatado no estudo de (CORDEIRO; GOMES, 2011), que como a priorização de requisitos é feita por meio da aplicação de uma matriz que visa identificar quais requisitos são mais importantes segundo a visão do *stakeholders* (leia-se clientes/usuários) essa priorização pode não ser coerente para a equipe do projeto.

A fim de tornar esta ideia mais clara, considere um cenário em que se pode notar esta priorização inconsciente de requisitos, como por exemplo: um requisito B que necessite totalmente de um outro requisito A e neste cenário B tem maior priorização e tem sua prática executada de maneira exemplar, porém, A é de certa forma negligenciado. Seguindo este exemplo, de nada adiantaria B ser totalmente implementado e ter seu plano seguido à risca se A falhar. Sendo assim A deveria ter

tido maior atenção, uma vez que neste cenário a prática bem sucedida de A levaria a um estado de qualidade parcial melhor do que no caso em que B tenha sido realizado de maneira exímia.

O ocasionar destas situações diminuirá a qualidade, pois para que as tarefas possam ser cumpridas dentro do prazo final, seria necessário um replanejamento de recursos, orçamento, tempo e equipe. Mas muitas vezes o replanejamento destes não é possível ou muito custoso. Ou seja no âmbito de planejamento da qualidade, essas tarefas negligenciadas influenciam diretamente na qualidade.

Analisando estes quesitos, notou-se a oportunidade de agregar a modelagem organizacional a este contexto, onde a utilização desta pode contribuir para a elaboração das matrizes de priorização tanto dos requisitos de *software* e de qualidade quanto dos *stakeholders*. Além dos diagramas provenientes dessa modelagem poderem ser utilizados para o melhor entendimento dos *stakeholders*, apoiando-os no momento de avaliação dos requisitos, a fim de obter uma priorização mais confiável o seu uso pode tornar mais clara a identificação dos atores e os requisitos pertinentes a eles, sendo a análise destes, importante para a priorização dos *stakeholders*.

Assim, espera-se uma melhoria na qualidade do produto e que esta seja alcançada com menor esforço e retrabalho e que a utilização dos modelos organizacionais possa auxiliar de maneira mais eficiente no planejamento da qualidade, tanto na utilização da técnica de priorização, quanto na obtenção do documento de conformidade, das métricas de qualidade e dos *checklists* de qualidade.

Optou-se pela utilização do *framework* i\* para a modelagem organizacional, devido a trabalhos recentes na área que mostraram uma integração eficiente deste *framework* com outras áreas de conhecimento do (PMBOK,2008), sendo a utilização deste na gerência de riscos e na gerência de tempo, (VARELA, 2011; HENN, 2012). Alia-se a estes trabalhos prévios a motivação de que este *framework* vem sendo utilizado em outras áreas e seu poder de representação de intencionalidades é relatado em pesquisas mais recentes.

## 1.3 Objetivos

Na comunidade de engenharia de software nota-se uma emergente e constante preocupação no âmbito de desenvolvimento de software de alta qualidade. Para atingir esse objetivo, várias abordagens têm sido propostas (VASCONCELOS; *et al.*, 2011), e este trabalho apresenta como proposta uma nova abordagem para auxiliar o processo clássico de planejamento de qualidade,

integrando uma técnica de modelagem organizacional ao processo de planejamento da qualidade descrito pelo PMBOK (PMI, 2008).

A partir dos diagramas gerados pelo framework i\*, as tarefas de gerar documento de conformidade, obter métricas de qualidade e criar listas de verificação de qualidade (*checklists* de qualidade) quanto utilizá-las, tornam-se mais simples e diretas, uma vez que são apoiadas por representações visuais que mostram uma visão mais simplista do processo/sistema do que um documento que contém as mesmas informações, só que porém descritas textualmente.

Por isso considera-se que é possível um melhor entendimento das etapas do projeto por meio do diagrama SR do projeto. Também considera-se que estes modelos são de grande ajuda tanto para a elaboração de matrizes de priorização de elementos - a qual é uma ferramenta que pode ser utilizada a fim de contribuir na identificação dos elementos que estarão presentes nas métricas de qualidade e no *checklist* de qualidade do planejamento da qualidade – quanto para um melhor discernimento dos elementos que estão sendo priorizados. Isso mostra-se importante pois esse entendimento pode modificar a avaliação, sendo mais confiável e conseqüentemente isso agrega qualidade ao processo. Na Figura 1 pode ser visto um resumo da proposta.



Figura 1: Princípio da abordagem

### 1.3.1 Objetivos Gerais

O objetivo geral deste trabalho é auxiliar a priorização de requisitos e *stakeholders* com a modelagem organizacional, para melhorar o planejamento da qualidade.

### 1.3.2 Objetivos Específicos

Estudar como é feita a priorização.

Estudar qual técnica é apropriada e como ela funciona.

Estudar como é possível apoiar a tarefa de elaboração das matrizes de priorização com a utilização do *framework* i\*.

Analisar a utilização dos modelos do i\* para apoiar os clientes em suas avaliações.

Obtenção da melhoria da qualidade do *software* através do auxílio do *framework* i\* no planejamento da qualidade, mais especificamente no auxílio da obtenção do documento de conformidade, das métricas de qualidade e dos *checklists* de qualidade.

# Capítulo 2

## Planejamento de Qualidade

Neste capítulo apresenta-se a gerência de qualidade como proposta pelo PMBOK (PMI, 2008), e mais especificamente daremos ênfase à fase de planejamento de qualidade, a qual é o foco da proposta deste trabalho. A seção 2.1 traz uma introdução à sobre alguns conceitos de qualidade. Na seção 2.2 será dado destaque para o planejamento de qualidade de *software* que é uma das três fases da gerência de qualidade conforme proposto pelo PMBOK (PMI, 2008).

## 2.1 Qualidade de Software

### 2.1.1 Conceitos Básicos sobre qualidade de software

Desenvolvimento de software de alta qualidade tem sido uma preocupação constante da comunidade de Engenharia de Software (VASCONCELOS; *et al.*, 2012).

O termo qualidade segundo ITC/Imetro (2003, p.113), é definido como:

[...] o grau em que um conjunto de características que preenchem as exigências, ou em outras palavras, pode ser considerada como a adequação para o uso ou conformidade, observadas as exigências do usuário, já que é ele quem determina se produto ou serviço tem qualidade desejada. (ITC/IMETRO, 2003, p.113).

A tarefa de obter qualidade em processos e produtos de engenharia de software não é uma tarefa trivial. Há muitos fatores que dificultam alcançar os objetivos de qualidade. Porém, não há nada mais decepcionante do que produzir um software que não atenda às necessidades dos clientes. Muitos dos recursos são gastos porém em muitos casos ocorre uma enorme frustração por parte dos clientes, em vista do software final encomendado (LEITE, 2001).

Qualidade de um produto depende da qualidade do processo. Muitas empresas apoiam-se nas especificações dos modelos ISO/IEC 15504 e CMMI para melhorar o padrão de qualidade do software de suas empresas (HWANG, 2010).

Quando se trata de qualidade, deve-se ter em mente que não basta apenas garantir a qualidade do produto, mas também é desejável que se garanta a qualidade do processo.

Na visão orientada ao produto, destacam-se três aspectos que dominaram a pesquisa e a prática de desenvolvimento de software (LEITE, 2001):

- A ênfase na qualidade das representações;
- A ideia de que a qualidade está diretamente relacionada com os testes do produto final; e
- O processo de desenvolvimento era centrado em fases tipificadas por produtos bem definidos.

É muito difícil escrever especificações de software completas. Portanto, embora um produto de software possa estar coerente com a sua especificação, os usuários podem não considerá-lo como um produto de alta qualidade por ele não corresponder às suas expectativas (SOMMERVILE, 2003).

Como relata Joseph M. Juran (JURAN; GODFREY, 1999), pioneiro na qualidade, ter qualidade significa que as características do produto tenham de satisfazer as necessidades dos clientes e, assim, proporcionar a satisfação do cliente. Qualidade também significa: a não existência de deficiências, onde essas deficiências são erros que exigem retrabalho ou resultam em falhas depois que um produto tenha sido entregue a um cliente (JURAN; GODFREY, 1999). Essas falhas podem resultar em reclamações, insatisfação do cliente, ou consequências que levam o produto a um estado inseguro sendo demasiado grave para o usuário. A melhoria da qualidade relacionada com deficiências geralmente custa menos (JURAN; GODFREY, 1999).

Por outro lado, os clientes são importantes por muitas razões, eles compram produtos repetidamente; dizem aos seus amigos para comprar esses produtos; eles definem as necessidades de novos produtos; indicam o interesse em, ou a falta de interesse, ou até mesmo a oposição a estes produtos e indicam produtos potenciais. Talvez o mais importante de tudo, eles se queixam e proporcionam informações valiosas e conhecimento em forma de *feedback* para melhorar estes produtos (ROSE, 2005).

Uma vez que projetos tem mais do que um cliente é de suma importância que se considere estes usuários. Ainda segundo (ROSE, 2005) a tendência é ver a pessoa ou organização que paga as contas como o único cliente ou o único cliente de importância. Porém, uma visão mais experiente reconhece a existência de um número de clientes que geralmente caem em três categorias, sendo elas: clientes externos (são aqueles que estão fora da organização ou da equipe do projeto), clientes internos (são considerados como próximo passo na cadeia de processos) e clientes ocultos (são partes interessadas - pessoas ou organizações que não participam diretamente do projeto).

Estas constatações sugerem que clientes desempenham quatro papéis que são importantes para um planejamento de qualidade do produto eficiente e para o desenvolvimento do projeto:

1. Fornecer dados sobre as necessidades e os requisitos.
2. Definir os padrões de requisitos.
3. Avaliar o produto.
4. Prover *feedback*.

### **2.1.2 Modelos de avaliação de qualidade/maturidade**

Há vários modelos para mensurar qualidade, entre eles os mais encontrados na literatura são a norma ISSO/IEC 9126 e o CMMI:

A norma ISO/IEC 9126 voltada para o processo de desenvolvimento de software tem como principal objetivo estabelecer uma estrutura que seja comum entre os processos de ciclo de vida, visando auxiliar as organizações a compreenderem todos os componentes presentes, a fim de conseguirem executar projetos de forma mais eficaz.

O Modelo de Maturidade em Capacitação (CMMI-SCAMPI) que, segundo (VASCONCELOS *et al.*, 2012) é um método para avaliar objetivamente o processo de desenvolvimento de uma organização de acordo com os requisitos das respectivas áreas de processos do CMMI. Trata-se da consolidação de evidências, (por exemplo: apresentações, documentos e entrevistas), relacionadas com a execução do processo em projetos reais. As evidências são usadas, por uma equipe de avaliação, para apoiar a atribuição de notas para as práticas, objetivos e finalmente, para áreas de processos avaliadas. A caracterização de um nível de implementação de uma prática pode se enquadrar em um de quatro diferentes tipos:

- Totalmente implementado: evidências estão presentes e são julgados suficientes para demonstrar a implementação de uma prática e pontos fracos não foram encontrados.
- Largamente implementado: evidências estão presentes e são julgados suficientes para demonstrar a implementação de uma prática, mas alguma fraqueza é encontrada.
- Parcialmente implementada: embora algumas informações sugerem que aspectos da prática são implementados, alguns ou todos os dados necessários estão ausentes ou julgados inadequados e alguma fraqueza é encontrada. Ou se os dados fornecidos à equipe de avaliação demonstram conflitos, isto é, alguns dados indicam que a prática é aplicada e outros que não o é.

- Não implementado: alguns ou todos os dados necessários estão ausentes ou são considerados inadequados, onde os dados fornecidos não apoiam a execução da prática, e alguma fraqueza é encontrada.

Com base nas classes definidas para a prática, cada objetivo específico ou genérico é classificado como:

- Satisfeito, se e somente se todas as práticas associadas são classificadas tanto como em grande parte implementado ou totalmente implementado e a agregação dos pontos fracos das práticas não tem um impacto negativo significativo sobre a realização do objetivo.
- Insatisfeito, se pelo menos uma das práticas associadas tem uma classe diferente de grande parte ou totalmente implementado.

O nível de capacidade de uma área de processo é definido a partir dos tipos definidos para os objetivos. Por exemplo, para cumprir com o nível 1, um processo deve satisfazer o objetivo genérico associado a este nível. Essa meta tem apenas uma prática genérica, que exige que todos os objetivos específicos associados com a área de processos devem ser satisfeitos. Se um dos objetivos específicos não é satisfeito, então a área de processo é considerada como tendo a capacidade de nível 0. Para níveis de capacidade superiores a 1, outras metas genéricas associadas com o específico e os níveis anteriores devem ser satisfeitas, o que impõem requisitos adicionais.

## 2.2 Gerenciamento de projetos e sua importância

O gerenciamento de projetos é a aplicação de conhecimentos, habilidades e técnicas para a execução de projetos de maneira eficiente. Essa estratégia para organizações permite que elas consolidem os resultados obtidos nos projetos com os objetivos do negócio, melhorando a qualidade de seus produtos e a competitividade do mercado (INSTITUTE, 2013).

Dessa forma, faz-se necessário a utilização de técnicas de gerência de projetos para que se possa realizar projetos que sejam coerentes com o planejamento das estimativas de recursos efetuadas bem como para que possam ir de encontro com as expectativas dos usuários a fim de alcançar certo nível de qualidade.

Um projeto é definido como esforço exercido temporariamente em prol da geração de um produto, serviço ou resultado exclusivo. Este é composto por inúmeras tarefas que são definidas durante o tempo de vida do projeto (PMI, 2013). Para a elaboração de um projeto coerente, dispõe-se de vários modelos de gerência de projeto, destacando-se os propostos por: *International Standards Organization* (ISO, 2001), *Software Engineering Institute* (SEI, 2013) e *Project Management Institute* (PMI, 2013) – sendo que para os fins propostos será utilizado o modelo proposto por *Project Management Institute* (PMI, 2013).

Desenvolver um bom projeto torna-se uma tarefa menos árdua seguindo-se os modelos de gerência de projeto, quando executado de forma precisa, sem ser negligenciado pelos gerentes de projeto e/ou demais envolvidos no projeto tende-se a obter tanto um processo de qualidade quanto de produto final com qualidade. Portanto, para obter um projeto bem sucedido faz-se necessário analisar corretamente os dados obtidos através de uma *elicitação*, atentando para alguns parâmetros, tais como o escopo do sistema, os recursos e atividades necessárias para sua conclusão, o tempo e custo estimado, entre outros (PRESSMAN, 1995).

O PMBOK apresenta um bom conjunto de conhecimentos e entre estes encontra-se uma área de gerência que é mais crítica, esta é a área de gerência de qualidade mais especificamente o planejamento da qualidade, sendo que a má execução deste pode levar a invalidar um produto.

### 2.2.1 Planejar Qualidade

O planejamento de qualidade é o processo de identificação dos requisitos e/ou dos padrões de qualidade acerca do projeto e do produto, também compreende a determinação de como satisfazê-

las e realiza uma documentação de como o projeto demonstrará a conformidade. Ela é uma das atividades da gerência de qualidade segundo o PMBOK e é uma atividade/prática importante nos modelos de maturidade e capacitação CMMI bem como na ISO.

Na Figura 2 podem ser vistas as etapas do planejamento da qualidade, as setas na figura apontam as três saídas que este trabalho propõe a auxiliar.

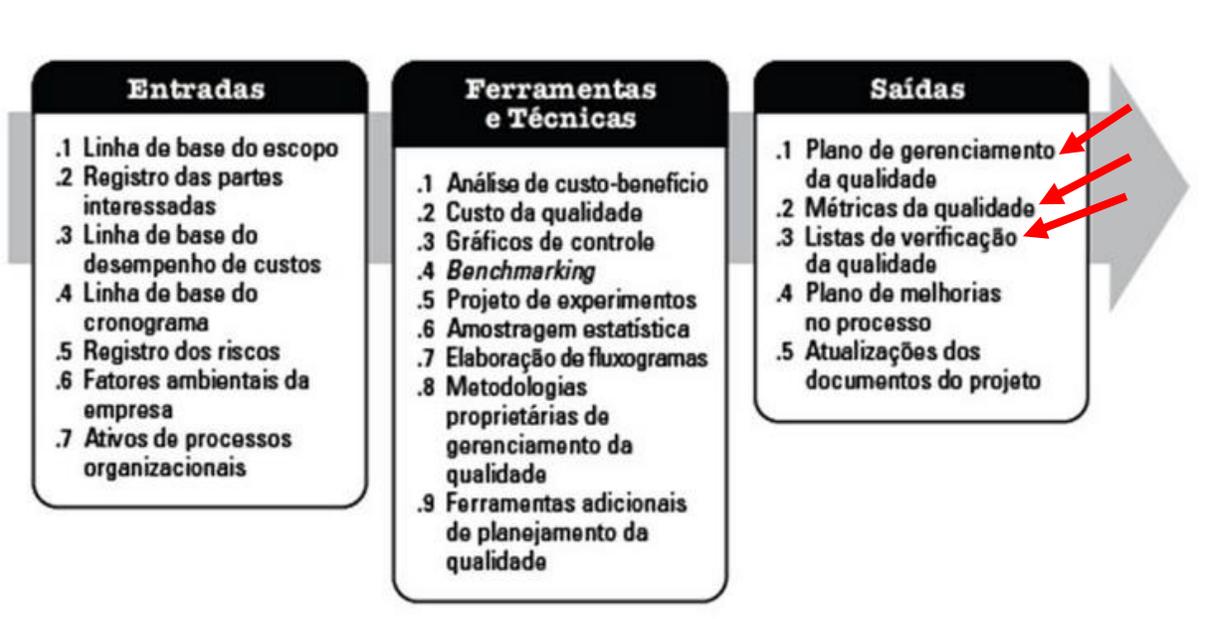


Figura 2: visão geral do planejamento da qualidade (PMBOK, 2008).

Serão descritas a seguir cada uma das atividades, entradas, ferramentas, técnicas e saídas que são propostas pelo (PMBOK 2008).

### 2.2.1.1 Entradas para planejar a qualidade

Na abordagem clássica para definir as atividades necessárias para a conclusão do projeto, fazem-se necessários os seguintes documentos (PMBOK, 2008):

- **Linha de base do escopo:** a linha base possui sete componentes, os quais são descritos a seguir:
  - **Declaração do escopo do projeto:** Este documento contém a descrição do projeto, principais entregas do projeto e os critérios de aceitação. A descrição do escopo do produto contém questões técnicas e preocupações que podem afetar o planejamento da qualidade. Os custos do projeto e da qualidade podem aumentar ou diminuir

dependendo dos critérios de aceitação definidos. Se todos os critérios de aceitação forem cumpridos isto implicará que as necessidades do cliente foram atendidas.

- **Estrutura analítica do projeto (EAP):** Identifica as entregas, os pacotes de trabalho e as contas de controle usadas para mensurar o desempenho do projeto. É gerada a partir do documento de requisitos, do documento de escopo do projeto e pode ser feito pelos ativos de processos organizacionais, como arquivos de projetos anteriores.
- **Dicionário da EAP:** Define as informações técnicas, como informações sobre a organização da EAP (hierarquia, descrição e critérios de aprovação).
- **Registro das partes interessadas:** Serve para identificar e registrar os ‘*stakeholders*’ ou partes que tem um interesse específico ou impacto na qualidade.
- **Linha de base do desempenho de custos:** Documenta a fase de tempo aceita, utilizada para mensurar o desempenho dos custos.
- **Linha de base do cronograma:** Documenta as medidas de prazos aceitas.
- **Registro dos riscos:** Documento que contém informações sobre as possíveis ameaças e oportunidades que podem interferir nos requisitos de qualidade.
- **Fatores ambientais da empresa:** Define fatores internos e externos que interferem de maneira direta ou indireta no processo de planejar a qualidade tais como cultura, a estrutura da empresa, normas, padrões e diretrizes da empresa ou do governo que possuam relação com a empresa.
- **Ativos de processos organizacionais:** Estes ativos podem ser encontrados na forma de informações, normas políticas, processos, procedimentos e planejamentos das atividades. A equipe de gerenciamento do projeto precisa garantir que todas as partes interessadas estarão totalmente cientes da política usada para o projeto.

### 2.2.1.2 Ferramentas e técnicas para planejar a qualidade

A seguir serão apresentadas algumas das técnicas e ferramentas para processar os dados obtidos nas entradas como a estrutura analítica do projeto, seu dicionário e os ativos organizacionais:

- **Análise do custo-benefício:** Compara-se o custo da etapa e o benefício que se espera desta etapa, para cada atividade de qualidade. Alguns benefícios como, menos retrabalho, maior nível de produtividade, aumento de satisfação das partes interessadas, entre outros.

- **Custo da qualidade (CDQ):** Os custos da qualidade são os custos totais em decorrência dos investimentos em prevenção de não conformidade com os requisitos.
- **Gráficos de controle:** Usado para identificar se um processo é estável ou se tem um desempenho previsível, a análise é feita com base em limitantes definidos pelo gerente de projeto, ao ultrapassar estes limites de especificação pode haver penalidades associadas. Estes gráficos podem ser utilizados para monitorar diversos tipos de variáveis de saída.
- **Benchmarking:** utilizado para comparar práticas de projetos reais ou planejados com demais projetos, a fim de identificar melhores práticas, considerar melhorias e fornece uma base para o mensurar o desempenho, contribuindo na obtenção melhores práticas que conduzam ao um desempenho superior.
- **Projetos de experimentos:** é um modelo estático utilizado para identificar agravantes de variáveis específicas de um produto ou processo em desenvolvimento. Também desenvolve um papel importante na otimização de produtos ou processos.
- **Matriz de priorização:** a matriz de priorização é uma matriz que pode levar em conta diversos elementos como, *stakeholders*, requisitos funcionais, requisitos não funcionais, entre outros, a fim de a partir da avaliação destes elementos identificar quais tem maior relevância. O PMBOK sugere a utilização de matriz de priorização mas não diz qual, então optou-se por utilizar a L-Shaped já que essa matriz é indicada por (ROSE, 2005), além disso ela é similar a abordagem Análise importância-Desempenho (IPA) utilizada por (CORDEIRO; FREITAS, 2011). Lembrando que a matriz de priorização é importante para a obtenção concisa das saídas já citadas no capítulo anterior e que esta carece de suporte para a sua elaboração, por isso propõe-se a utilização do i\* para auxiliar em sua elaboração e também para apoiar em seu preenchimento.

Essa matriz consiste em uma comparar os elementos de um-para-um. Para a construção da matriz, deve-se introduzir os nomes dos elementos a serem priorizados ao longo dos eixos verticais e horizontais. Um exemplo disto pode ser visto na Figura 3, onde as letras A,B,C,D,E e F representam os elementos a serem priorizados.

	A	B	C	D	E	F	Total da linha	Valor Relativo
A		5	1	10	1/5	1/5	16.4	0.21
B	1/5		1/5	1	1	5	7.4	0.09
C	1	5		1/5	1/10	5	11.3	0.14
D	1/10	1	5		1/5	1	7.3	0.09
E	5	1	10	5		1/10	21.1	0.26
F	5	1/5	1/5	1	10		16.4	0.21
<b>Total</b>							<b>79.9</b>	

Legenda :

10 Muito mais importante

5 Mais importante

1 Igual importância

1/5 Menos importante

1/10 Muito menos importante

Figura 3: Matriz L-Shaped (ROSE, 2005)

O próximo passo consiste em comparar os elementos uns aos outros sendo feito de um-para-um para determinar a sua importância com relação aos demais. Os valores que podem ser assumidos por essas comparações, variam entre 5 possibilidades. Sendo administrado o valor 10 para muito mais importante, ou seja, se considerarmos que A é muito mais importante que B então o valor da posição (A,B) deve assumir 10, no caso em que ele for considerado apenas mais importante, deve-se introduzir o valor 5. Se os elementos que estão sendo comparados tiverem igual importância, assumisse o valor 1. Os valores 1/5 e 1/10 são os opostos respectivos para 5 e 10, significando menos importante e muito menos importante.

Uma observação relevante a ser feita, é que a metade inferior da matriz é o inverso da metade superior, ou seja, se o elemento A em relação a B é mais importante (5) o inverso desta avaliação, ou seja B em relação a A seria menos importante (1/5). Essa observação pode ser útil para agilizar o processo de preenchimento.

Após concluída as comparações, as pontuações de cada linha devem ser somadas e o resultado de cada linha também deve ser somado, para determinar um total geral. Uma vez obtido o total geral, o último passo é dividir o total de cada linha pelo total geral. O resultado mostra as porcentagens totais de cada linha. Estes valores indicam a prioridade, por exemplo, analisando a Figura 3, podemos notar que ao final da atividade o elemento que foi considerado prioritário é o elemento E seguido de A, F, C, B, D.

Ainda neste contexto pode-se citar algumas outras ferramentas e técnicas que auxiliam no processamento desses dados, como: fluxogramas, a metodologia de desdobramento da função de qualidade (QFD), *brainstorming* e diagramas de afinidade (PMBOK, 2008).

### 2.2.1.3 Saídas do planejamento de qualidade

Como resultado obtido através das aplicações das ferramentas e técnicas descritas na seção anterior, deve-se obter as seguintes saídas, segundo (PMBOK, 2008):

- **Plano de gerenciamento da qualidade:** descreve como a equipe de gerenciamento de projetos implantará a política de qualidade da organização executora, este plano de gerenciamento inclui o controle da qualidade, a garantia da qualidade e abordagens para a melhoria contínua de processos. Este plano deve ser revisado na parte inicial do projeto a fim de evitar informações imprecisas.
- **Métricas da qualidade:** Define como um atributo do projeto ou do produto será medido e avaliado, podendo-se adicionar tolerância às variações aceitáveis.
- **Listas de verificação de qualidade:** Trata-se de uma ferramenta utilizada para verificar se as etapas necessárias para a conclusão de uma tarefa foram executadas.
- **Plano de melhorias no processo:** é um plano auxiliar ao plano de gerenciamento do projeto, utilizado para identificar atividades que tem seu valor aumentado e considera atividades dentro das áreas de: limite de processo, configuração do processo, métricas do processo e metas para melhoria do desempenho.
- **Atualizações nos documentos do projeto:** Os documentos do projeto têm a possibilidade de serem atualizados.

As saídas que serão apoiadas por este trabalho são: O documento de conformidade que também é conhecido como plano de gerenciamento de qualidade, métricas de qualidade e listas de verificação de qualidade, também conhecidas como *checklists* de qualidade.

## 2.3 Considerações finais

A partir da revisão bibliográfica notou-se que o guia de boas práticas PMBOK pode ter suas práticas ligadas ao planejamento de qualidade melhoradas. Ele permite a utilização de novas técnicas e conceitos a fim de obter melhores resultados na atividade de planejamento da qualidade. Inspirado por isto, este trabalho procurará apresentar uma abordagem para complementar a

abordagem clássica do planejamento de qualidade proposto pelo (PMBOK, 2008), a fim de obter saídas mais confiáveis e também apresentar uma forma de obter as três saídas que serão apoiadas por esta abordagem através de novos tipos de documentos – como os diagramas provenientes da modelagem organizacional i\* - que são mais rápidos e fáceis de obter/analisar.

# Capítulo 3

## O Framework i\*

Neste capítulo será apresentado o conceito de modelagem organizacional com foco na apresentação de algumas características que favorecem o seu uso na melhoria do processo de planejamento de qualidade de projetos de software. Particularmente, neste trabalho utilizaremos o framework de modelagem organizacional i\* (istar), o qual será brevemente descrito. Assim, nas subseções seguintes abordar-se-á quesitos acerca da importância da modelagem organizacional como uma técnica que permite facilitar a compreensão de ambientes organizacionais. Após estas considerações serem feitas, será apresentado o *framework i\** para modelagem organizacional, onde será explicado seus conceitos e sua forma original e também a sua aplicação e por fim são apresentadas algumas considerações acerca deste capítulo.

### 3.1 A Modelagem Organizacional e o Planejamento de Qualidade de Projetos de Software

O sucesso de um projeto está ligado a compreensão imprescindível do contexto do projeto por todos os seus participantes. Esta compreensão não é uma tarefa simples visto que cada projeto difere dos demais por diversos fatores, tendo sempre a difícil tarefa de *elicitação* e a compreensão dos requisitos de cada sistema (VARGAS, 2003).

A maior dificuldade em desenvolver uma engenharia de requisitos eficiente deve-se à *elicitação* correta dos requisitos organizacionais, funcionais e não funcionais, no intuito de compreender melhor o problema e o ambiente do mesmo (ALENCAR; CASTRO, 1999).

A fim de melhorar o planejamento da qualidade podemos observar com o que a Engenharia de Requisitos se preocupa, como, *elicitação*, priorização, rastreabilidade, documentação. Essas preocupações da Engenharia de Requisitos são quesitos pertinentes para a obtenção da qualidade do produto. No entanto, as ferramentas e técnicas indicadas pelo PMBOK para o planejamento de qualidade não dão suporte direto para estes quesitos. Então torna-se interessante a utilização de uma técnica que cubra essa deficiência.

Segundo (GRANDO 2010, p. 3), “A modelagem organizacional permite não só melhor entender os requisitos organizacionais que interferirão nos sistemas, mas também identificar alternativas para os vários processos da organização, facilitando os esforços durante o desenvolvimento do sistema computacional e permitindo que a análise seja mais bem integrada aos processos de desenvolvimento do sistema”. Esta ideia também é defendida por (ALENCAR, 1999), que a modelagem organizacional é importante por fornecer o entendimento e a compreensão do ambiente organizacional, bem como possibilitar uma análise que identifique elementos que irão afetar o projeto.

Sendo assim, a modelagem organizacional pode ser considerada como peça fundamental para que o software desenvolvido seja condizente com o que o cliente requisitou. Na modelagem organizacional são diagramados os requisitos tanto funcionais quanto alguns não funcionais e os processos de negócios da empresa a fim de dar suporte a equipe de desenvolvimento para que eles possam compreender melhor o problema, assim como conhecer a dinâmica do trabalho da equipe, o seu papel, as suas restrições, entre outros.

Portanto, tem-se demasiado interesse que o uso dessas técnicas possa também ser aplicados a equipes de desenvolvimento, visando melhorar a gerência com base nos objetivos definidos para cada membro da equipe, disponibilidade de recursos, entre outros (HENN, 2012).

Por conta dos motivos apresentados aqui, optou-se por adotar a abordagem de modelagem organizacional e como ferramenta desta abordagem, adotou-se o *framework* i\* o qual se pode notoriamente destacar o fato de que é possível realizar a modelagem social da equipe de desenvolvimento e também é possível representar as regras de negócios na modelagem.

A utilização do mesmo possibilita ao engenheiro de requisitos realizar a engenharia de requisitos com maior qualidade proporcionando a equipe um melhor entendimento do problema (YU; LI; LIU, 2011). Por meio da representação de atores e papéis disponíveis a partir do

*framework* i\* também é possível visualizar requisitos funcionais e não funcionais que não tinham sido visualizados.

Além dos motivos citados a escolha do *framework* i\* deve-se a grande gama de trabalhos presentes na literatura que utilizam o *framework*, bem como o suporte detalhado dos conceitos básicos do mesmo proporcionado pelo i\* *Wiki* (GRAU; *et al.*, 2008).

## 3.2 Uma Visão Sobre o Framework i\*

Proposto por (YU; LI; LIU, 2011) o *framework* i\* é uma técnica de modelagem organizacional que tem por objetivo modelar aspectos organizacionais envolvidos com o processo, através de diagramas de dependências estratégicas e de razões estratégicas, o qual possibilita melhor entendimento das razões inteiras dos atores, uma vez que elas são expressas explicitamente, levando ao melhor entendimento do engenheiro de requisitos durante a modelagem do sistema. Esse *framework* tem como principal objetivo representar através de modelos, os atores participantes e as dependências entre os mesmos, objetivando que as metas próprias sejam cumpridas, recursos sejam fornecidos, tarefas sejam cumpridas como planejado e metas flexíveis (*softgoals*) sejam “razoavelmente satisfeitas” (YU; LI; LIU, 2011).

Este *framework* auxilia na engenharia de requisitos haja visto que a extração de requisitos na maioria das vezes será uma tarefa difícil, pois pode haver problema de falta de compreensão do escopo por parte do engenheiro de requisitos.

Haja visto que a falta de compreensão pode ser motivada pelo fato de que o usuário muitas vezes não tem domínio sobre o problema que deseja retratar ou não consegue se expressar com precisão levando ao desentendimento por parte dos participantes este desentendimento só será percebido na entrega do sistema ou nas validações feita pelo mesmo. Esses desentendimentos quando percebidos levam a mudanças bruscas no desenvolvimento, comprometendo todo o planejamento e o cronograma.

Segundo (SANTANDER, 2002) “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”.

Este *framework* apresenta uma visão mais ampla do projeto para todos os envolvidos e no que se diz respeito a gerência de projetos traz uma visão mais detalhada do projeto por meio de atores e objetivos.

Tais constatações fazem do *framework* um importante aliado do engenheiro de requisitos para o entendimento inicial do sistema.

No aspecto de modelagem do processo de negócio, o *framework* i\* é um modelo que capta as motivações e fundamentos de maneira explícita e com isso ele consegue proporcionar um quadro mais sistemático para os esforços de reengenharia, trazendo melhores ligações para o desenvolvimento do sistema (YU; LI; LIU, 2011).

No *framework* i\* as partes interessadas são representadas como atores os quais possuem intenções em relação ao sistema em desenvolvimento, essas intenções são representadas como metas. Esses atores têm por objetivo satisfazer essas metas e para satisfaze-las o ator pode precisar de outros atores (YU, 1995).

O i\* é composto por dois modelos de diagramação, os quais possuem diferentes níveis de detalhamento, são eles: O diagrama de dependências estratégicas e razões estratégicas. Seus principais componentes são mostrados na Figura 4.

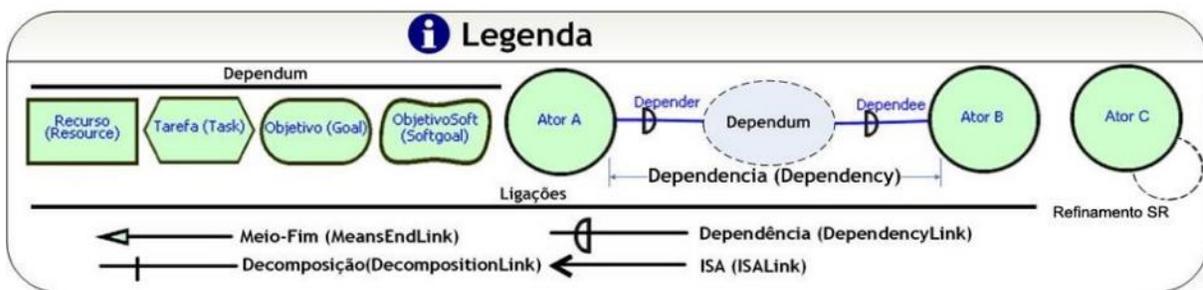


Figura 4: Os principais componentes do *framework* i\* (BRISCHKE, 2011).

### 3.2.1 O Modelo de Dependências Estratégicas (SD)

O modelo de dependências estratégicas concede uma descrição intencional do processo em termos de uma rede de relações entre atores (YU; LI; LIU, 2011).

Este modelo concentra-se em modelar as relações existentes no processo, tais relações são geradas no intuito de satisfazer as intenções dos atores. Com isso, este modelo pode auxiliar na identificação dos intervenientes, na análise das oportunidades e das vulnerabilidades.

O modelo de dependências estratégicas consiste em nós e links, onde um nó representa um ator, e um link representa a ligação entre dois atores, onde uma entidade ativa realiza ações para atingir metas no exercício da sua função. O termo ator é empreendido para referir genericamente qualquer unidade para a qual dependências intencionais possam ser atribuídas (YU; LI; LIU, 2011).

Segundo (ALENCAR, 1999), o modelo de dependências estratégicas representa as intencionalidades, motivações e dependências entre os atores integrantes do projeto, descritas por meio de nós e ligações, sendo uma relação constituída basicamente como apresentado na Figura 4:

- **Depender:** é representado pelo ator A, onde este se relaciona com o ator B (dependee) a fim de satisfazer uma dependência.
- **Dependee:** o ator B será o dependee, haja visto que 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 apresentado na Figura 4:
  - **Recurso:** pode ser uma informação, parte de um sistema ou subsistema, entre outros.
  - **Tarefa:** atividade da qual o depender precisa do dependee para conclusão.
  - **Objetivo:** tarefa ou atividade de suma importância para o projeto.
  - **Objetivo-soft:** meta ou intencionalidade objetivada pelo depender, a qual o dependee deve satisfazer.

Todos os três elementos são conectados pelas ligações do tipo dependência, como descritas na Figura 4.

Há seis tipos de associações entre atores os quais são representados na Figura 5:

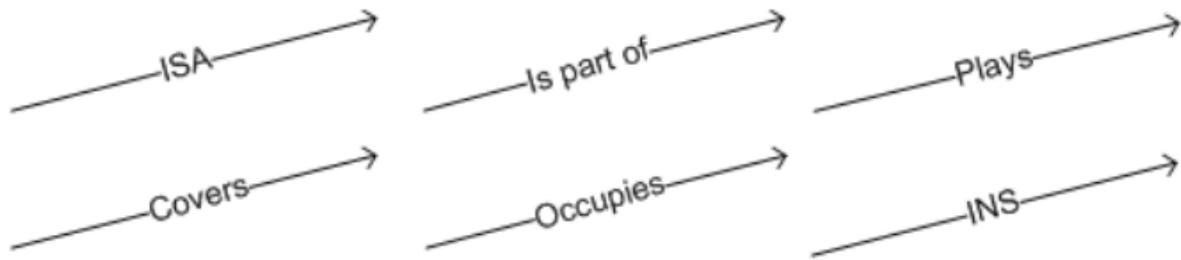


Figura 5: Associações entre atores (GRAU *et al.*, 2008)

- **Is part of:** descreve dependências intencionais entre o todo e sua parte (papéis ou posições).
- **ISA:** representa generalização, quando um ator é uma classe especializada de outro ator.
- **Play:** usada entre um agente e um papel, para determinar qual agente executara o papel
- **Cover:** descreve uma relação entre uma posição e os papéis, servindo para indicar qual posição cobre um papel.
- **Occupies:** indica que um agente ocupa uma posição.
- **INS:** indica uma instância de uma entidade, é usada para representar uma instância específica de uma entidade mais geral.

Segundo (YU; LI; LIU, 2011), esses elementos são suficientes para gerar representações do modelo de dependências estratégicas (SD), esses mesmos elementos fazem parte do modelo de razões estratégicas (SR), o qual ainda contém outros elementos.

### 3.2.2 O Modelo de Razões Estratégicas (SR)

O modelo de razões estratégicas surge como um complemento ao modelo de dependências estratégicas, visando detalhar tarefas, objetivos e atividades decompondo-as de maneira que possam ser identificadas as etapas necessárias para satisfazer as dependências.

Os demais elementos que fazem parte do modelo de razões estratégicas serão descritos a seguir, como proposto por (YU; LI; LIU, 2011):

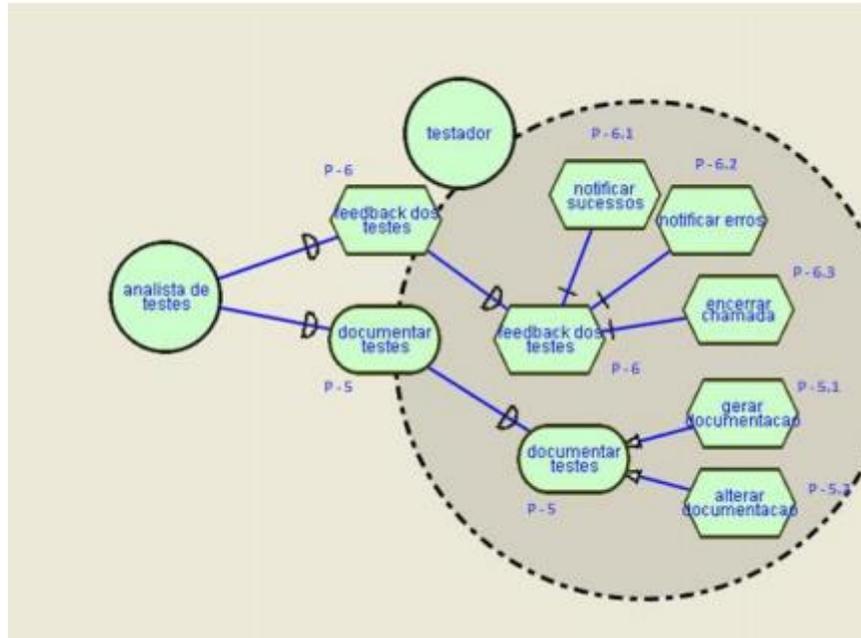


Figura 6: Exemplo de diagrama SR (Henn, 2012)

A área circundada pela linha pontilhada na Figura 6 corresponde ao escopo do ator “testador”, sendo assim, todas as dependências contidas em seu escopo são de sua responsabilidade satisfaze-las. Observando este contexto pode-se notar a inserção de dois novos tipos de ligações (YU; LI; LIU, 2011):

- **Ligações de Decomposição:** uma ligação do tipo AND, onde, para a conclusão de uma tarefa que é dividida em várias etapas é necessário que todas as sub-tarefas sejam concluídas, essas etapas podem ser do tipo: tarefa, recursos, objetivo ou objetivo-soft.
- **Ligações meio-fim:** reflete o relacionamento entre um nó fim e um meio para atingi-lo, é uma ligação do tipo OR, onde uma tarefa pode ser concluída de diversas maneiras.

Este modelo ainda apresenta algumas ligações complementares, estas ligações são chamadas de ligações de contribuição, as quais são utilizadas de maneira aditiva aos diagramas. No intuito de expressar informações, essas ligações são dos tipos apresentados na Figura 7.

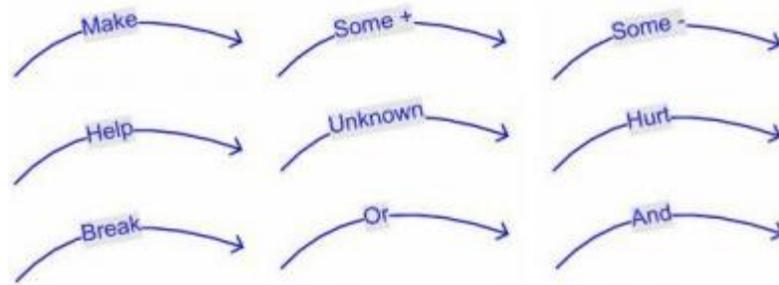


Figura 7: Ligações de contribuição (GRAU *et al.*, 2008).

Como pode ser percebido na Figura 7, as ligações ou links de contribuição podem ser de nove tipos distintos. A seguir serão apresentadas descrições destas ligações segundo (YU; LI; LIU, 2011):

- **Make:** indica uma contribuição que é capaz de satisfazer um objetivo.
- **Help:** indica uma ajuda para conclusão de uma tarefa ou objetivo, porém esta ajuda não é suficiente por si só, como o *make*.
- **Some +:** contribui mais que o *help*, porém também não é autosuficiente como o *make*, mas é um meio termo entre esses dois links de contribuição.
- **Break:** tem efeito negativo, anulando automaticamente outro objetivo-soft. Por exemplo: se uma tarefa não for desempenhada e um objetivo-soft depende desta tarefa, o objetivo será anulado.
- **Hurt:** inverso ao *help*, contribui de forma negativa para a conclusão de um objetivo-soft.
- **Some -:** inverso ao *some +*, dificulta na conclusão de uma tarefa.
- **Unknow:** indica que a contribuição para a conclusão de um objetivo-soft é desconhecida.
- **Or:** indica que uma tarefa se satisfeita pode ser concluída ou se a tarefa indicada pelo link *or* for satisfeita.
- **And:** demonstra que a tarefa indicada pelo link precisa ser satisfeita para que a tarefa em si seja satisfeita.

### **3.3 Considerações Finais**

Neste capítulo realizou-se uma introdução à modelagem organizacional, onde foi exposta a importância do mesmo na área de gerência de projetos e sua contribuição na engenharia de requisitos. Também, o capítulo apresentou o *framework* de modelagem organizacional i\*, dando ênfase em seus dois modelos: o modelo de dependências estratégicas, um diagrama que possibilita a representação de dependências, intenções e motivações entre os atores envolvidos no projeto; e o modelo de razões estratégicas, o qual tem um nível maior de detalhamento de tarefas e objetivos, permitindo dividi-los em elementos de menor complexidade para que se possa obter o máximo grau de simplificação das tarefas e entendimento do projeto.

## Capítulo 4

# Integrando o Framework i\* ao Planejamento de Qualidade

Neste Capítulo será apresentada a proposta de integração do *framework* de modelagem organizacional i\* ao planejamento de qualidade conforme descrito pelo guia de boas práticas (PMBOK, 2008), tendo por objetivo criar uma abordagem que apoie o processo de planejamento de qualidade de projetos de *software*.

A seção 4.1 traz uma introdução a proposta e ao contexto ao qual se visa auxiliar, esclarecendo os principais pontos que diferem a nova abordagem da tradicional.

Na seção 4.2 apresentar-se-á a proposta do processo de integração do *framework* i\* ao planejamento da qualidade e mais especificamente no uso deste *framework* para auxiliar na construção das matrizes de priorização proposta pela técnica L-shaped e seu uso na definição dos elementos que farão parte das métricas de qualidade e do *checklist* de qualidade.

### 4.1 Introdução

Almejando uma abordagem mais eficiente para o planejamento da qualidade, é desejável que aspectos como as dependências entre as pessoas envolvidas no projeto possam ser fragmentadas em dependências mais específicas, a fim de detalhar com maior completude as categorias de dependências. Assim essas dependências poderiam assumir a forma de um tipo mais condizente com a intenção, podendo ser categorizada como objetivo, objetivo-*soft*, tarefa, recurso, entre outros.

A decomposição de dependências também visa prover uma visão de todo o processo por meio de diagramas e modelos. Esta visão pode vir a suprir a carência encontrada no modelo tradicional,

onde há dificuldade em gerar uma visão do projeto que contemple fatores estratégicos relacionados aos envolvidos no projeto.

A presente proposta é apresentada considerando como base o processo proposto pelo guia (PMBOK, 2008), por este ser muito conceituado e utilizado em trabalhos na área de Engenharia de *Software*. No entanto, uma dificuldade em aplicar corretamente este guia deve-se a sua subjetividade por ele ser aplicável a diversas áreas, com isso ele acaba sendo genérico. Por isso este trabalho busca apresentar uma maneira de auxiliar a etapa de planejamento de qualidade com a modelagem organizacional, a fim de diminuir ou eliminar a subjetividade que ronda a utilização das matrizes de priorização a qual pode afetar nas saídas: documento de conformidade, métricas de qualidade e *checklists* de qualidade.

Para apoiar essa nova abordagem utiliza-se o *framework* i\*, por este facilitar a obtenção e visualização de dados como, os *stakeholders*, os requisitos do projeto e requisitos da qualidade. Estes são elementos essenciais para a elaboração das matrizes de priorização, que serão úteis para as saídas já citadas, além de também contribuir para a elaboração de listas de verificação de qualidade, na obtenção das métricas de qualidade e gerar o documento de conformidade. A utilização da técnica de priorização de requisitos de *software* e qualidade que é feita por meio da matriz *L-shaped*, também contribui para que os desenvolvedores tenham uma base metodológica para conduzir o trabalho de desenvolvimento, além de proporcionar enfoque nos requisitos que mais contribuem na percepção de qualidade do usuário.

Com esta abordagem espera-se auxiliar na obtenção de algumas saídas do processo de planejamento da qualidade, através da visão proveniente dos modelos gerados por meio do *framework* i\* no intuito de facilitar o entendimento das relações entre os atores e os requisitos ligados a eles. E, após a obtenção destes, tornar mais eficaz e confiável a etapa da priorização dos requisitos, apresentando os modelos SR do sistema aos *stakeholders* a fim de apoiar-los em suas avaliações, proporcionando-lhes maior entendimento do contexto, obtendo conseqüentemente uma avaliação mais fiel. Fazendo isso considerando a visão de vários *stakeholders*, os quais também serão priorizados, propõe-se uma abordagem complementar ao processo clássico proposto pelo (PMBOK,2008). A partir disto, os artefatos gerados por esta abordagem servirão para serem analisados junto às partes interessadas no projeto em etapas de validação. Na próxima seção esta abordagem será apresentada e exemplificada.

O estudo de caso será realizado no próximo capítulo.

## 4.2 Proposta

Esta seção tem por objetivo apresentar a proposta de integração do planejamento de qualidade e modelagem organizacional por meio do *framework* i\*. Será apresentado um conjunto de passos a serem executados para a obtenção de tal integração.

Serão apresentadas diretrizes associadas aos passos para a aplicação da proposta. O processo para o planejamento da qualidade é descrito no guia (PMBOK, 2008). Este processo é executado visando considerar informações resultantes dos modelos organizacionais SD e SR do projeto, dos requisitos de *software* e de qualidade, os quais são obtidos por meio do *framework* i\*.

Estas diretrizes são propostas considerando maneiras de realizar o planejamento da qualidade em um projeto, de forma a não depender necessariamente da existência da documentação completa do projeto, conforme apresentado na seção 2.2.1.1.

A proposta será apresentada conforme o conjunto de passos, onde cada um destes passos pode conter uma ou mais diretrizes e há passos que não contem diretrizes.

Para cada passo proposto, especificar-se-á quais entradas são necessárias para sua execução, quais diretrizes devem ser seguidas e as saídas decorrentes destas execuções.

Para melhor entendimento das diretrizes propostas, serão criados exemplos com base no estudo de caso que será detalhado no próximo capítulo. Este estudo de caso trata-se do desenvolvimento de uma aplicação web para o grupo do laboratório de engenharia de *software* da UNIOESTE. A representação gráfica do conjunto de passos pode ser vista na Figura 8.

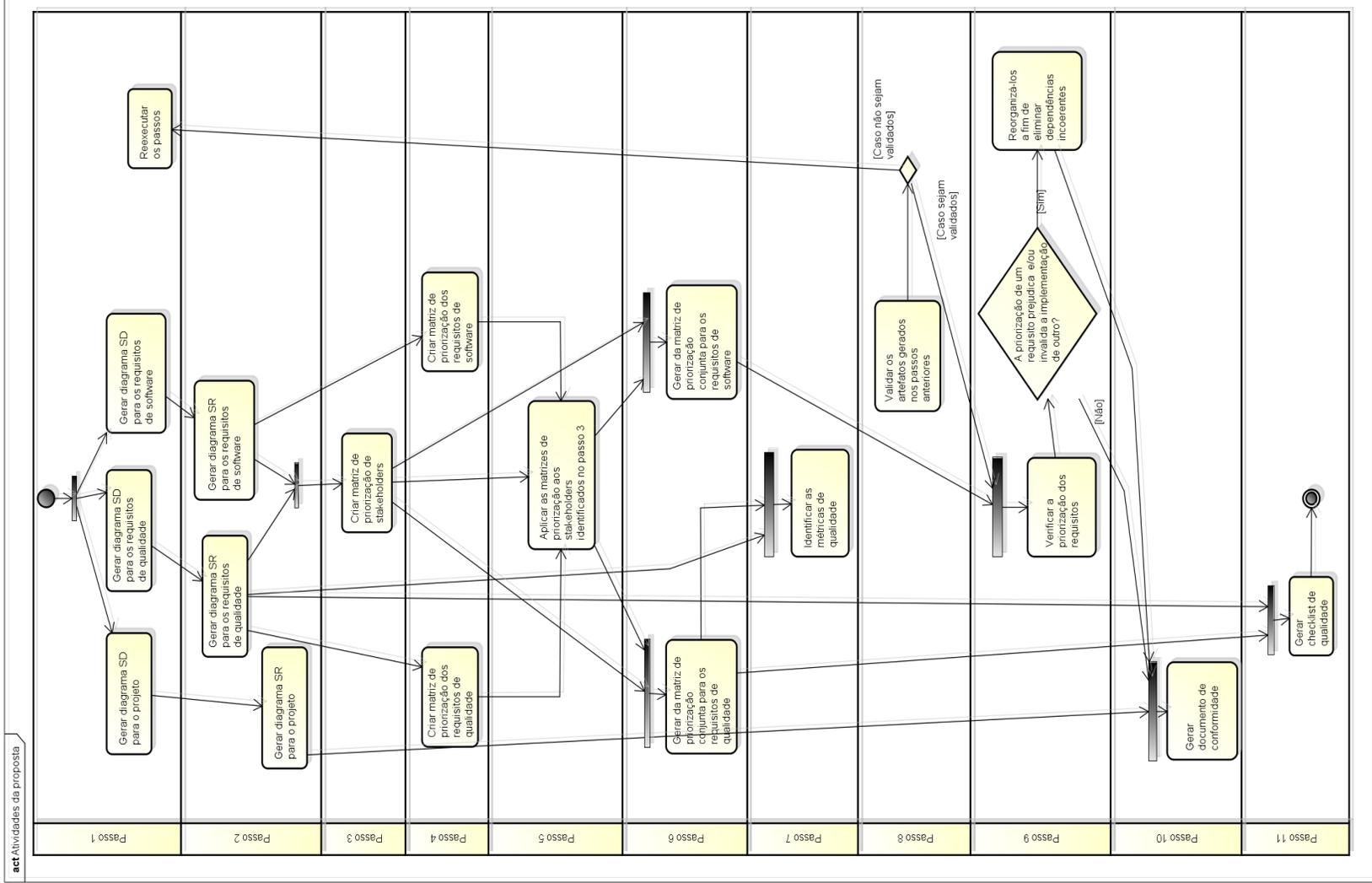


Figura 8. Diagrama de atividade da proposta.

No **passo 1** como saída deve-se obter os modelos SD gerados por meio de diretrizes que servirão para identificação de atores e suas dependências, neste passo gera-se um modelo SD para os requisitos funcionais do sistema computacional pretendido, um modelo SD para os requisitos não funcionais para o sistema e um modelo SD para o projeto, os quais incluem os *stakeholders*, as atividades pertinentes a cada ator identificado e as ligações de dependências entre eles. Estas informações serão úteis no contexto de planejamento de qualidade para realizar a elaboração das matrizes de priorização.

No **passo 2** como saída, serão obtidos os modelos SR, para o produto e para o projeto, os quais são gerados seguindo uma diretriz que considerará as razões estratégicas e terá foco nos atores. Este modelo auxiliará no planejamento de qualidade com relação aos requisitos de qualidade uma vez que esses requisitos serão mapeados para objetivos/objetivos-*soft* e estes serão decompostos em tarefas menores, para que sejam melhor gerenciados e entendidos. Além disso, onde o modelo SR do projeto servirá de base para o documento de conformidade, onde sempre que necessário, poderá ser consultado e rapidamente descobrir quais tarefas faltam ser realizadas para alcançar o objetivo e os modelos SR do sistema, para a auxiliar na descrição das métricas da qualidade e da elaboração do *checklist* de qualidade.

No **passo 3** será feita a priorização dos *stakeholders*, com base nos atores identificados através do diagrama SD do sistema gerado no passo 1. Para este passo e o passo 4, utilizar-se-á a técnica de criação da matriz L-Shaped. Essa priorização é feita pela equipe do projeto e busca identificar os *stakeholders* que terão opiniões mais significativas sobre os requisitos de qualidade. A priorização destes *stakeholders* será utilizada posteriormente no intuito de obter destes os requisitos de qualidade, os quais também serão mapeados para os modelos SR dos requisitos de qualidade e de *software*.

No **passo 4** será criada a matriz L-Shaped de priorização dos requisitos de *software* e dos requisitos de qualidade, com base nos requisitos identificados observando as dependências entre os *stakeholders* e o sistema computacional, então os elementos identificados nos modelos SD dos requisitos de *software* e de qualidade, que são do tipo *dependun* serão mapeados para as suas respectivas matriz de priorização ao fim desse processo, a saída obtida – os esqueletos das matrizes de priorização dos requisitos de *software*/qualidade – será aplicada no passo posterior.

No **passo 5** a matriz obtida no passo 4 será aplicada individualmente aos *stakeholders* a fim de obter a visão destes com relação aos requisitos funcionais do *software* e aos requisitos de qualidade. Como saída deste passo, será obtido um número de matrizes igual ao número de *stakeholders*. Lembrando que a priorização destes requisitos serve de condução para o trabalho de desenvolvimento e também para que se tenha um foco maior nos requisitos que terão maior influência sobre a qualidade percebida pelo usuário.

No **passo 6**, as matrizes obtidas nos passos 3 e 5 serão combinadas para obtenção de uma matriz integrada de requisitos e *stakeholders*. Essa matriz é construída listando os *stakeholders* ao longo do eixo horizontal e os requisitos ao longo do eixo vertical. Em seguida o valor para cada requisito é preenchido pela multiplicação da prioridade do *stakeholder* com o valor da priorização deste requisito na visão do *stakeholder* em questão.

No **passo 7** deve-se identificar as métricas de qualidade, essas métricas podem ser extraídas do modelo SR dos requisitos de qualidade, sendo que as decomposições dos objetivos-soft constituem informações relevantes para este.

No **passo 8** deve-se validar os resultados obtidos junto às partes interessadas, este passo é útil para analisar os resultados com os clientes e para confirmar o entendimento dos interessados antes de prosseguir.

No **passo 9** a equipe do projeto, deve analisar a priorização dos requisitos e ajustar a ordem dos requisitos se estes concluírem que um requisito que tem mais prioridade depende da implementação de outro que tem menos prioridade. Este passo serve para mitigar o problema de priorização inconsciente que foi relatado pelo estudo de (CORDEIRO; FREITAS, 2011), citado na seção 1.2.

No **passo 10** será criado o documento de conformidade do projeto, o qual utilizará como base o modelo SR do projeto, onde pode-se obter os responsáveis pelas atividades e subatividades necessárias para a execução concreta do projeto.

E por fim o **passo 11** deve ser feito após o desenvolvimento do *software* ou após a realização de iterações nas quais se obtém uma versão parcial do sistema computacional. Deve-se verificar por meio de um *checklist* se os requisitos prioritários e os requisitos de qualidade foram atendidos. Assim como no passo 7 a saída é gerada no planejamento de qualidade mas apenas é utilizada na segunda atividade do gerenciamento de qualidade que é a atividade de garantia de qualidade onde

estas duas saídas serão utilizadas para verificar a conformidade do que está sendo executado com o que foi planejado.

## 4.3 Diretrizes

A seguir são descritas e exemplificadas as diretrizes associadas a cada passo da proposta. Para facilitar a compreensão das mesmas serão utilizados breves exemplos com base no estudo de caso que será completamente descrito no próximo capítulo. Este estudo de caso envolve três clientes (orientador 1, orientador 2 e orientando) eles esperam obter como produto de *software* um *web* site para um grupo de pesquisa em engenharia de *software* com teor voltado para o âmbito acadêmico, esperando obter através do mesmo informações como publicações, informações de projeto, lista de parceiros, entre outros.

### 4.3.1 Passo 1: criar os modelos de dependências estratégicas

Neste passo, a presença de alguns documentos relativos ao projeto, ou um estudo da organização e da equipe de desenvolvimento são de grande auxílio porém não são obrigatórios. No caso em que entradas como, declaração de escopo do projeto, registro de *stakeholders*, plano de gerenciamento de recursos humanos e Fatores ambientais da empresa não estiverem disponíveis, o processo pode ser feito por meio do estudo etnográfico da organização e da análise organizacional, via entrevistas ou questionários aplicados aos membros chave da organização, entre outros. Neste passo serão gerados três modelos SD, onde o modelo SD do projeto será utilizado no passo 2 para obtenção do SR do projeto; o modelo SD dos requisitos de *software* e dos de qualidade serão utilizados no passo 2 para a partir dele obter o SR, no passo 4 onde seus elementos vão servir para auxiliar a elaborar as matrizes de priorização e no passo 5 para apoiar os *stakeholders* no momento de avaliação dos requisitos.

A diretriz descrita a seguir tratará de como gerar os modelos SD.

#### 4.3.1.1 Passo 1 – Diretriz 1: identificação dos atores participantes do projeto e stakeholders

Esta diretriz tem o objetivo de gerar modelos tanto para o projeto quanto para o produto de *software* que deseja-se desenvolver. Com base nas informações contidas na documentação do

registro das partes é possível identificar quais são os atores que irão exercer interação com o projeto, caso esse documento não exista, é possível estabelecer os atores através de entrevistas.

Como dito na subseção 4.3.1, os atores envolvidos no projeto podem ser identificados por meio da observação dos documentos como a declaração do escopo do projeto, do plano de gerenciamento de recursos humanos e também analisando-se a equipe envolvida no projeto.

Outra maneira possível para a identificação dos atores é observar o ambiente de trabalho e questionar de maneira individual quais são as tarefas executadas por cada membro da equipe. Entrevistas também podem ser uma maneira para identificar os atores.

Um exemplo de atores identificados neste passo para o estudo de caso envolvendo o desenvolvimento de uma página *web* para um grupo de pesquisa em engenharia de *software* é retratado na Figura 4.2 na qual são apresentados os atores para o modelo do sistema e na Figura 9 os atores envolvidos no projeto.

Uma vez identificado os atores, a próxima diretriz será identificar os objetivos, intencionalidades e motivações entre os atores e mapeá-los em termos de dependências entre os mesmos.



Figura 4.2: Atores identificados para o SD do sistema.



Figura 9: Atores identificados para o SD do projeto.

### 4.3.1.2 Passo 1 – Diretriz 2: identificação de dependências entre atores

Utilizando o *framework i\**, uma relação entre dois atores é mapeada em termos de dependência. A partir desta premissa pode-se entrevistar esses atores no intuito de identificar quais são as atividades indispensáveis para a conclusão da dependência. Este tipo de atividade será representada como dependência (ver seção 3.2.1). Cada *dependum* identificado possui uma característica diferente no relacionamento entre atores, podendo ser: um objetivo, um objetivo-*soft*, uma tarefa ou um recurso. Estes tipos podem ser identificados através das entrevistas realizadas e também pode ser auxiliado por documentos pré-existentes.

A fim de satisfazer uma dependência a priori, deve-se identificar dois tipos de atores, sendo eles: qual ator será responsável pela execução (*dependee*) e qual ator depende desta execução (*depender*).

A título de exemplificação, será apresentado como se efetua o mapeamento de uma dependência do tipo objetivo, onde o Orientando (*depender*), tem uma dependência com o Sistema (*dependee*) para alcançar o objetivo (*dependum*) de exibir publicações. A representação pictórica disto, está retratado na Figura 10.



Figura 10: Representação de dependência do tipo objetivo.

A fim de retratar uma dependência do tipo objetivo-*soft*, será apresentado na Figura 11. Pode-se observar que o Orientador 1 (*depender*) necessita do ator Sistema (*dependee*) para a satisfazer a missão de obter uma resposta rápida do site, o qual neste caso é representado pela dependência do tipo objetivo-*soft* (*dependum*) performance.



Figura 11: Representação de dependência do tipo objetivo-soft.

Uma dependência do tipo tarefa pode ser observada na Figura 12, na qual o ator Analista de testes (*dependor*) depende do ator Testador (*dependee*) para a realizar a tarefa (*dependum*) de realizar a os testes do sistema.



Figura 12: Representação de dependência do tipo tarefa.

Por último é apresentado o mapeamento de uma dependência do tipo recurso na Figura 13. No caso em que o ator Analista de qualidade (*dependor*) requer que o ator Cliente (*dependee*) especifique os requisitos de qualidade (*dependum*).

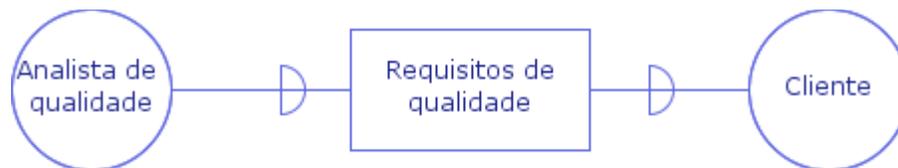


Figura 13: Representação de dependência do tipo recurso.

A fim de identificar possíveis dependências implícitas, pode-se novamente realizar uma entrevista com os atores identificados e/ou analisar documentos pré-existentes.

Após seguir/aplicar as diretrizes 1 e 2, encerra-se o passo 1. Um exemplo de modelo SD de envolvidos no projeto da página web para um grupo de pesquisa em engenharia de *software* é apresentado na Figura 14, um exemplo de modelo SD para representação dos requisitos funcionais obtidos por meio das entrevistas/questionários/documentos pré-existentes e também dos *stakeholders* envolvidos com o sistema onde eles possuem uma ligação do tipo ISA (ver seção 3.2.1) com o ator cliente, pode ser visto na Figura 15. E um exemplo de requisitos de qualidade também obtidos entrevistas/questionários/documentos pré-existentes é representado na Figura 16.

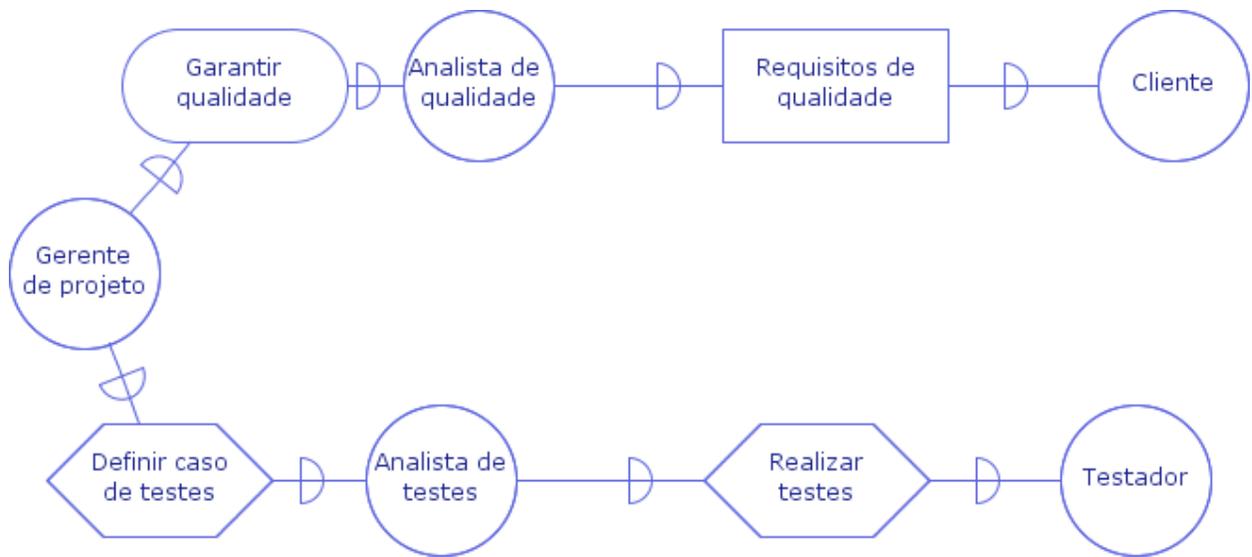


Figura 14: Modelo SD do projeto gerado no passo 1.

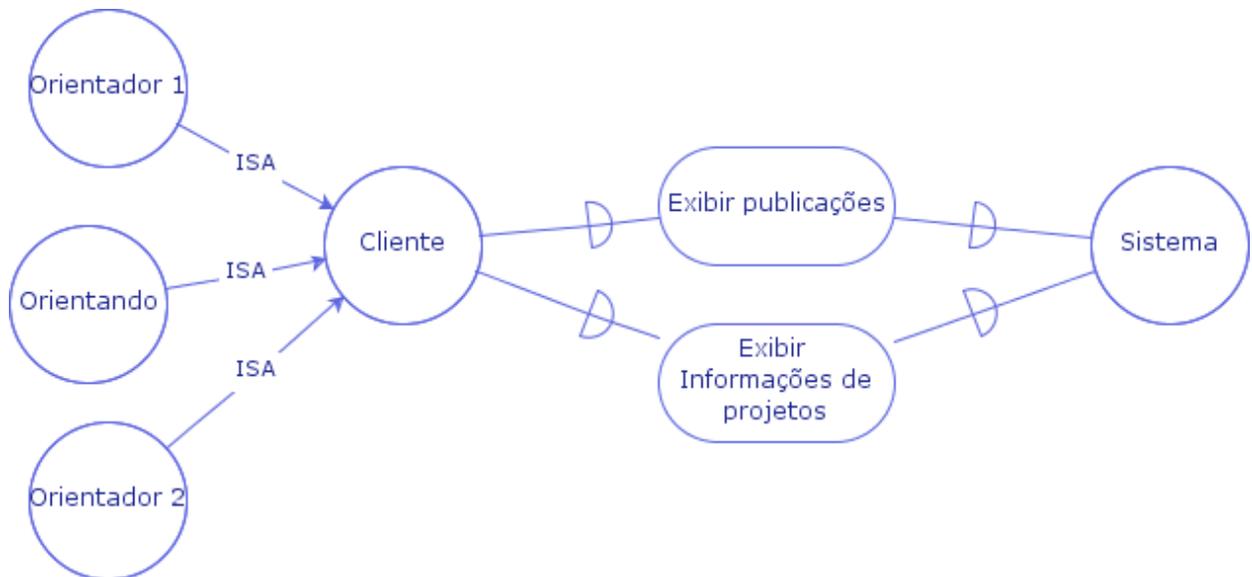


Figura 15: Modelo SD do sistema para requisitos do produto.

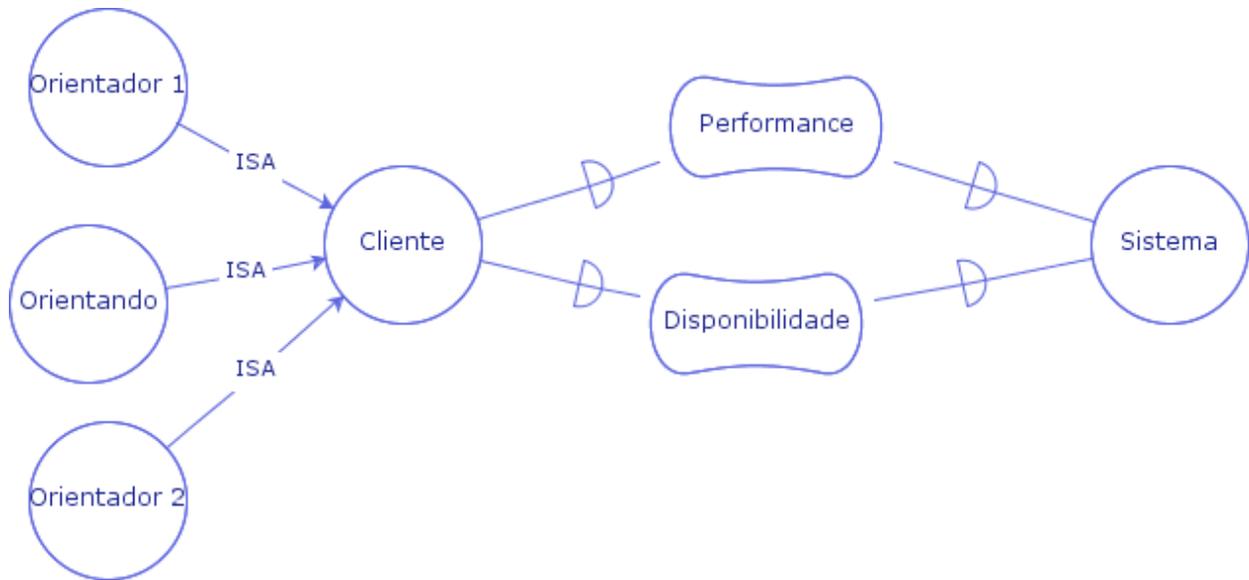


Figura 16: Modelo SD do sistema para requisitos de qualidade.

### 4.3.2 Passo 2: criar os modelos de razões estratégicas

Com base nos diagramas SD obtidos durante o passo 1, para a equipe de projeto e para o *software* pretendido, cada ator que tem o papel de *dependee* – responsável pela execução – deverá detalhar as atividades de sua responsabilidade, por conseguinte, espera-se obter o maior detalhamento possível quando se trata de decomposição de atividades. Este passo contribui para o entendimento da equipe do projeto acerca do que necessita ser feito para o cumprimento dos requisitos. Este passo também facilita a percepção das dependências dos requisitos e por isso servirá para auxiliar a equipe do projeto na execução do passo 8 e ajudará os *stakeholders* a terem uma percepção maior acerca dos requisitos e com isso poderão responder às matrizes de priorização de requisitos tanto de software quanto de qualidade de maneira mais coerente. Porém, este passo não contribui para a elaboração da matriz L-Shaped para priorização de requisitos. Outra contribuição esperada derivada do modelo gerado para representar os requisitos de qualidade é a utilização deste para geração de um *checklist* de qualidade, onde posteriormente se verificará se os requisitos estão sendo cumpridos.

A seguir serão listados fatores, artefatos ou documentos que podem ser utilizados para o auxílio da modelagem de um SR:

- Modelos de dependências estratégicas SD.
- Entrevistas com os *dependee*.

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

A identificação das atividades pode ser feita pela declaração de escopo do projeto, mas caso ela não exista as mesmas podem ser obtidas através da entrevista com cada *dependee* no intuito de rastrear as sub-tarefas realizadas por este para identificar possíveis fatores que influenciem na realização de uma dependência. Neste passo então são gerados três diagramas SR, sendo gerado um diagrama respectivo a cada um dos diagramas SD gerados no passo anterior.

#### **4.3.2.1 Passo 2 – Diretriz 3: geração dos modelos SR com foco nos atores envolvidos no processo**

Este modelo é utilizado para detalhar as razões estratégicas sobre as dependências do tipo tarefa ou objetivo. Para obter informações para a criação do mesmo, é recomendado adotar como base o modelo SD do projeto gerado no passo 1 e também é possível consultar os documentos e informações citados anteriormente, uma vez que estes estejam disponíveis.

Relembrando, as atividades podem ser obtidas por meio de entrevistas com os *dependees* e também por meio de documentos pré-existentes. Tomando por base o modelo SD, deve-se analisar cada atividade ou objetivo a fim de decompô-lo em atividades de menor complexidade – em casos onde isto é possível.

Em última instância, considera-se que o SD possui elementos que podem ser considerados como pacotes de atividades. O que se propõe no modelo SR é desmembrar estes pacotes de atividades em subatividades, a fim de detalhar aquelas atividades.

No intuito de realizar tal desmembramento pode-se ainda questionar cada *dependee*, a fim de obter mais informações relativas as atividades atribuídas a ele para com isso melhor entender a atividade melhor detalhar a atividade, haja vista que o *dependee* pode ser considerado o melhor elemento para detalhar os passos necessários para a conclusão da atividade concernente a ele.

Na Figura 17, pode-se observar o modelo SR para o projeto, na Figura 18, o modelo SR para os requisitos do sistema encomendado e na Figura 19 o modelo SR para os requisitos de qualidade do sistema encomendado, os quais foram obtidos como saída deste passo.



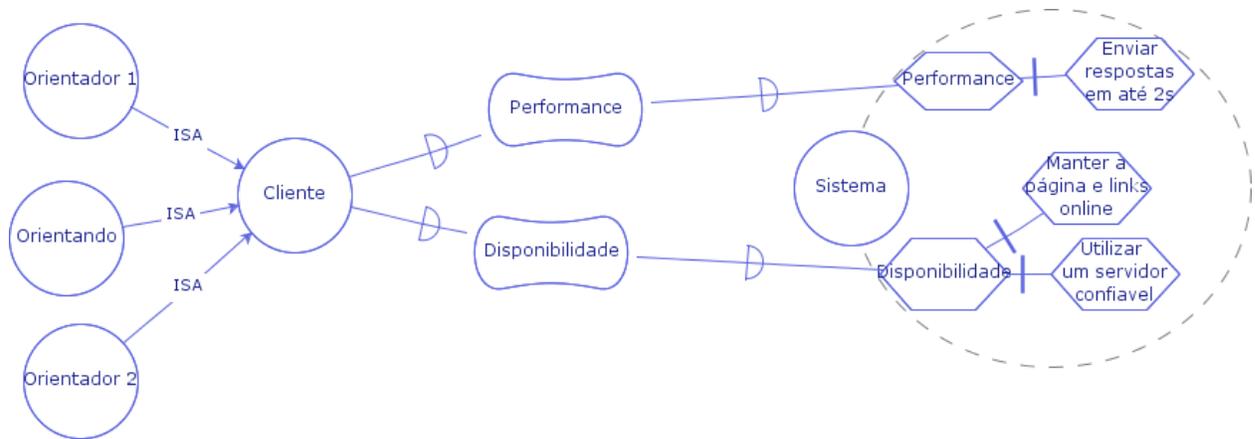


Figura 19: Modelo SR dos requisitos de qualidade do sistema.

### 4.3.3 Passo 3: criar a matriz de priorização dos stakeholders

Para a execução deste passo, será utilizado o diagrama SD com foco nas intencionalidades dos *stakeholders* em relação ao sistema obtido no passo 1. Os atores contidos nesse modelo serão mapeados para a matriz L-Shaped, a qual foi abordada anteriormente no capítulo 2 seção 2.2.1.2 e então serão priorizados. Na Figura 20 pode ser visto como os elementos da matriz de priorização de *stakeholders* são obtidos. É apresentado na Tabela 1 um exemplo de resultado da avaliação dos *stakeholders* feita pela a equipe do projeto. Esta tabela é o artefato gerado neste passo.

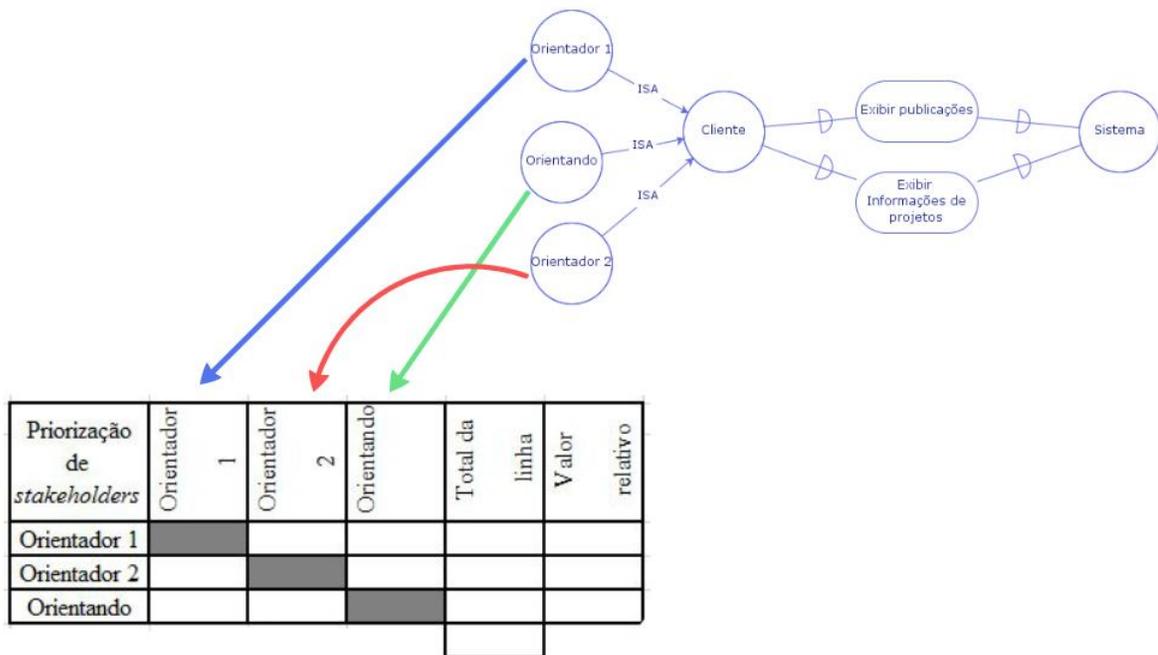


Figura 20: Obtendo os *stakeholders* para priorização a partir do modelo SD.

Tabela 1: Exemplo de matriz de priorização de *stakeholders*.

Priorização de <i>stakeholders</i>	Orientador 1	Orientador 2	Orientando	Total da linha	Valor relativo
Orientador 1		1	10	11	0.636
Orientador 2	1		5	6	0.347
Orientando	1/10	1/5		0.3	0.017
				17.3	

Cabe lembrar que o valor 10 representa que um elemento é muito mais importante, 5 para mais importante, 1 para igual importância, 1/5 para menos importante e 1/10 para muito menos importante.

#### 4.3.4 Passo 4: criar a matriz de priorização de requisitos

Este passo tem por objetivo a criação da matriz L-Shaped para priorização de requisitos de *software* e de qualidade. Os elementos que são utilizados nessas matrizes de priorização são os *dependuns* que foram mapeados no passo 1 para os diagramas SD dos requisitos funcionais do *software* e dos requisitos de qualidade. É apresentado na Figura 21a como são obtidos os elementos para priorização dos requisitos de *software* e na Figura 21b como são mapeados os requisitos de qualidade do diagrama SD de qualidade para a matriz de priorização dos requisitos de qualidade. Então os elementos que foram inseridos nessas matrizes serão avaliados no passo 5 pelos clientes.

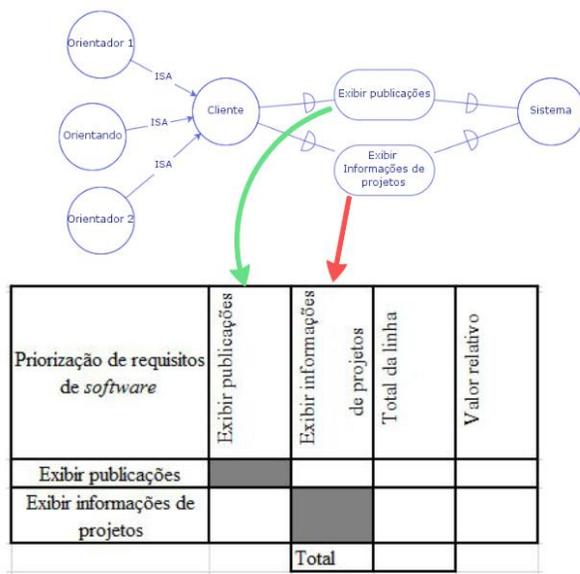


Figura 21a: Obtendo os requisitos para priorização do modelo SD requisitos de *software*.

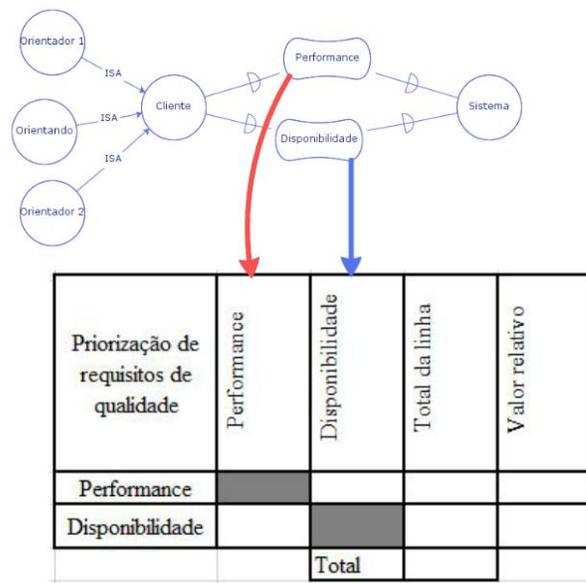


Figura 21b: Obtendo os requisitos de qualidade dos a partir para priorização a partir do modelo SD dos requisitos de qualidade.

O esqueleto da matriz de priorização para requisitos de *software* obtido no passo 4 com base no diagrama SD dos requisitos de *software* da Figura 15 pode ser visto na Tabela 2.

Tabela 2: Matriz L-shaped para priorização de requisitos de software.

Priorização de requisitos de <i>software</i>	Exibir publicações	Exibir informações de projetos	Total da linha	Valor relativo
Exibir publicações				
Exibir informações de projetos				
		Total		

E na Tabela 2.1 é representado a o esqueleto da matriz de priorização de requisitos de qualidade que foi obtida a partir do diagrama da Figura 16.

Tabela 2.1: Matriz L-shaped para priorização de requisitos de qualidade.

Priorização de requisitos de qualidade	Performance	Disponibilidade	Total da linha	Valor relativo
Performance				
Disponibilidade				
		Total		

#### 4.3.5 Passo 5: aplicar a matriz de priorização de requisitos aos atores

Com a execução deste passo objetiva-se descobrir quais requisitos são mais importantes de acordo com a visão de cada ator identificado por meio do diagrama SD do sistema (ver passo 1).

Para este passo, os diagramas SR do sistema gerados no passo 2 estão à disposição dos clientes, a fim de que a visão macro proveniente dos diagramas do sistema que foi encomendado apoie-os na avaliação dos requisitos de *software* e de qualidade. A título de exemplificação neste momento só serão apresentadas as matrizes de priorização para os requisitos de qualidade, mas o mesmo tem de ser feito para os requisitos de *software*. Na Tabela 3, 3.1 e 3.2 pode-se observar um exemplo do que poderia ser uma avaliação do Orientador 1, Orientador 2 e Orientando, respectivamente, quanto aos requisitos de qualidade.

Tabela 3: L-shaped para requisitos de qualidade na visão do Orientador 1.

Priorização de requisitos de qualidade visão do Orientador 1	Performance	Disponibilidade	Total da linha	Valor relativo
Performance		5	5	0,962
Disponibilidade	1/5		1/5	0,038
		Total	5,2	

Tabela 3.1: L-shaped para requisitos de qualidade na visão do Orientador 2.

Priorização de requisitos de qualidade visão do Orientador 2	Performance	Disponibilidade	Total da linha	Valor relativo
Performance		1	1	0,5
Disponibilidade	1		1	0,5
		Total	2	

Tabela 3.2: L-shaped para requisitos de qualidade na visão do Orientando.

Priorização de requisitos de qualidade visão do Orientando	Performance	Disponibilidade	Total da linha	Valor relativo
Performance		10	10	0,990
Disponibilidade	1/10		1/10	0,010
		Total	10,1	

#### 4.3.6 Passo 6: gerar matriz conjunta de priorização de requisitos com a influência de cada ator

Neste passo as matrizes obtidas nos passos 3 e 5 serão combinadas para obtenção da matriz final de priorização, a qual considera as avaliações dos atores e também a influência que estes atores tem no contexto do projeto. Na Figura 22 pode ser vista a interação entre as matrizes a fim da obtenção da matriz conjunta onde multiplica-se o valor de priorização de cada *stakeholder* com o valor da priorização de cada elemento que este avaliou, ao final obtêm-se uma matriz que considera as visões de cada *stakeholder* com a influência que este tem no projeto.

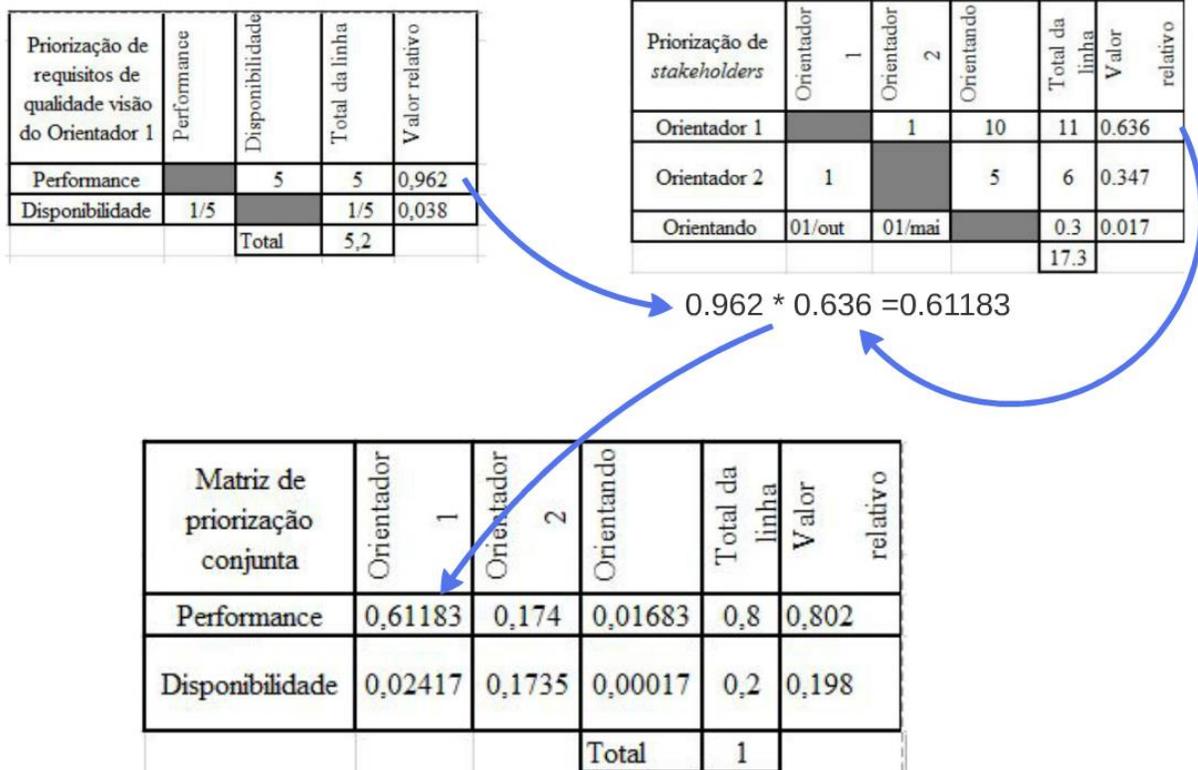


Figura 22: Processo para a obtenção da matriz de priorização conjunta.

Uma das matrizes que deve ser obtida como resultado deste passo pode ser vista na Tabela 4. Cabe lembrar que deve-se obter uma matriz de priorização conjunta para os requisitos de *software* e outra para os requisitos de qualidade.

Tabela 4: Matriz de priorização conjunta para requisitos de qualidade.

Matriz de priorização conjunta para requisitos de qualidade	Orientador 1	Orientador 2	Orientando	Total da linha	Valor relativo
Performance	0,61183	0,17350	0,01683	0,8	0,802
Disponibilidade	0,02417	0,17350	0,00017	0,2	0,198
			Total	1	

### 4.3.7 Passo 7: identificar as métricas de qualidade

Este passo é apoiado pelo diagrama SR dos requisitos de qualidade gerado no passo 2 (ver exemplo na Figura 19). As informações contidas nele servem para identificação das métricas de qualidade e o que é necessário para alcançá-las. Este passo também pode ser apoiado pelas descrições obtidas nas entrevistas com os *stakeholders*.

A fim de exemplificar este passo, na Tabela 4.1 pode-se analisar uma métrica de qualidade obtida através do diagrama SR segundo a Figura 19. Essa métrica está relacionada ao objetivo-*soft* de performance, o critério adotado para satisfazer essa métrica é derivado das subatividades que foram mapeados no diagrama SR dos requisitos de qualidade como pode ser visto na Figura 19.

A utilização da matriz de priorização para requisitos de qualidade pode ser um determinador que auxiliará a tomada de decisão quanto aos requisitos que farão parte ou não do produto de *software*. Portanto a matriz de priorização dos requisitos de qualidade (ver Tabela 4) é importante, uma vez que a presença ou ausência de um requisito de qualidade pode modificar as métricas obtidas nessa saída, ou seja, as métricas de qualidade relacionadas a requisitos que foram cortados fora do plano, não devem aparecer nesta saída, ou seja, se dispõe-se de recursos suficientes para satisfazer apenas um requisito, a análise da matriz de priorização da Tabela 4 indica que o requisito que deveria ser satisfeito é o requisito de qualidade Performance.

Tabela 4.1: Métrica de qualidade.

Métrica	Critério
Obter resposta rápida do servidor.	As páginas devem ser carregadas em tempo menor ou igual a dois segundos.

### 4.3.8 Passo 8: validação dos artefatos

No passo 8 todos os artefatos, modelos SD e SR do projeto e do sistema encomendado (tanto para requisitos de *software* quanto os de requisitos de qualidade); as matrizes L-shaped de priorização para *stakeholders*, requisitos de *software* e de qualidade e matrizes conjuntas e as métricas de qualidade, devem ser reunidos e apresentados perante as partes interessadas no intuito de analisar os dados obtidos, verificar se nada está faltando e também garantir que todos estejam cientes do que vai ser feito para que assim possam prosseguir.

#### **4.3.9 Passo 9: verificação da priorização dos requisitos de *software* e qualidade**

Para este passo, a equipe do projeto deve basear-se no modelo SR gerado no passo 2. A equipe do projeto deve analisar se não há nenhuma dependência que afete a execução do planejamento. Esta dependência como já foi citada, acarreta no caso em que um requisito A para ser implementado, necessita que um requisito B tenha sido implementado antes. Pela priorização de requisitos considerando a visão dos usuários, podem haver casos em que isto ocorra. Portanto os diagramas SR dos requisitos de *software* e de qualidade devem ser utilizados para este fim.

Tomando como exemplo a saída da Tabela 4, que foi obtida no passo 6, pode-se observar que o requisito de qualidade performance é mais prioritário do que o requisito de disponibilidade, porém avaliando que não adianta garantir que o servidor responda em até dois segundos, se a página estiver indisponível. Isto é caracteriza-se como uma priorização inconsciente. Então a ordem deve ser alterada, a fim de garantir primeiramente que se tenha um sistema que estará disponível o tempo todo, para depois garantir o tempo de resposta. Ou seja, de nada adianta ter um tempo desejável de resposta se o sistema estiver *off-line*.

#### **4.3.10 Passo 10: gerar documento de conformidade**

Será criado um documento para conformidade do projeto que conterà os requisitos do processo, ou seja, as atividades descritas no diagrama SR do projeto. Lembrando que este diagrama pode ser facilmente interpretado pela equipe do projeto e a partir da análise dos mesmos podem ser fácil e rapidamente obtidos os objetivos a serem realizados e as sub-tarefas necessárias para a conclusão desse objetivo e também os atores responsáveis por satisfaze-los. Com base no diagrama SR da Figura 17, podemos criar um documento como o exemplo apresentado na Tabela 4.2. Este documento é composto por três colunas, sendo que a coluna Responsável receberá o ator *depende* ou seja o responsável por satisfazer a dependência ligada a ele, a coluna Atividade que é preenchida com o *dependum* e a coluna Como será realizada é preenchida com base nas sub-tarefas que são necessárias para satisfazer o *dependum*.

Tabela 4.2: Exemplo de documento de conformidade criado com base no diagrama SR do projeto.

Responsável	Atividade	Como será realizada
Programador	Entregar versões executáveis	<ol style="list-style-type: none"> <li>1. Desenvolver funcionalidades</li> <li>2. Integrar novas funcionalidades.</li> <li>3. Documentar Desenvolvimento</li> </ol>
Programador	Realizar correções	<ol style="list-style-type: none"> <li>1. Abrir chamado para novos testes</li> <li>2. Atualizar documentação</li> <li>3. Efetuar correções</li> </ol>

#### 4.3.11 Passo 11: gerar checklist de qualidade

Este é o último passo desta abordagem e este é o passo necessário para a obtenção da última saída (*checklist* de qualidade) do planejamento da qualidade que este trabalho auxilia. O *checklist* é elaborado com base no diagrama SR dos requisitos de qualidade (ver Figura 19) e um exemplo do artefato obtido como saída deste passo pode ser visto na Tabela 4.3. Este documento é composto de duas colunas, onde na primeira especifica-se o requisito, ou seja, o *dependum* obtido por meio do diagrama SR da Figura 19 e a coluna Satisfeito onde deverá especificar se o requisito foi implementado e para fazer essa verificação, pode-se apoiar novamente no diagrama SR da Figura 19 a fim investigar se as sub-tarefas necessárias para a conclusão do *dependum* foram satisfeitas.

Tabela 4.3: *Checklist* de qualidade.

Requisito	Satisfeito?
Performance	Sim
Usabilidade	Sim

A utilização do artefato gerado por este passo é feita na atividade de garantia de qualidade que é a segunda atividade da gerência de qualidade. Este artefato deve ser utilizado depois que iterações já tenham sido feitas e após o desenvolvimento do *software*. A aplicação deste passo também serve para reunir informações para monitorar os resultados do projeto a fim de garantir que os resultados estão em conformidade com as especificações. Sugere-se que estas verificações sejam feitas a cada

nova atualização no intuito de prevenir erros encadeados, uma vez que o custo associado ao retrabalho de uma funcionalidade é menor do que o associado a um conjunto de funcionalidades.

## 4.4 Considerações Finais

Neste capítulo foram apresentados os passos e as diretrizes que os constituem, de modo que a obtenção das saídas – documento de conformidade, métricas de qualidade e *checklist* de qualidade - do processo de planejamento de qualidade de *software* seja auxiliado e realizado utilizando uma abordagem de modelagem organizacional via *framework* i\*.

Os passos e diretrizes foram pensados, não somente considerando aspectos organizacionais presentes no ambiente do projeto como também para auxiliar gerentes de projeto e engenheiros de requisitos quando a documentação sugerida no processo pelo (PMBOK, 2008) esteja ausente. E também a melhoria na tarefa de elaboração das matrizes de priorização através do *framework* i\* - sendo que esta tarefa era considerada um desafio para a equipe de projeto - a qual permite obter saídas como, as métricas de qualidade e o *checklist* de qualidade, de maneira mais confiável.

Neste trabalho foi proposto que o *framework* i\* permite mapear e representar intencionalidades e relações entre os membros integrantes de um projeto, sobre a forma de tarefas, recursos e objetivos, tornando-o uma alternativa de grande valia para servir de base ao processo de planejamento de qualidade.

A utilização do *framework* i\* permite identificar de maneira rápida e eficaz os atores e as atividades a eles relacionadas nos modelos, possibilitando obter uma visão macro do processo provendo maior entendimento às partes interessadas no projeto.

O próximo capítulo apresenta a aplicação desses esses passos e diretrizes a um estudo de caso.

# Capítulo 5

## Avaliação da integração do *framework* i\* ao planejamento da qualidade

Neste capítulo realiza-se um estudo de caso aplicando a abordagem proposta a um projeto de *software* que envolve a construção de uma aplicação *web* para o grupo de pesquisa em engenharia de *software* da UNIOESTE.

Uma vez que os passos para aplicação da abordagem já foram descritos no capítulo anterior, este capítulo terá enfoque nos métodos usados em cada passo.

Será apresentada a abordagem *Goal Question Metric* (GQM) que será utilizada para analisar a eficiência da proposta. Posteriormente será apresentado o contexto do estudo de caso e em seguida os passos propostos na abordagem serão aplicados e exemplificados.

### 5.1 Contexto

O projeto a ser desenvolvido consiste na elaboração de uma aplicação *web* para o Laboratório de Engenharia de Software (LES) da UNIOESTE. Esta página deve ser capaz de apresentar conteúdos como: informações sobre os projetos em andamento, a lista de parcerias, lista de membros do LES, relatórios técnicos, notícias, artigos científicos e frameworks desenvolvidos pelo grupo. Esses requisitos foram previamente elicitados por meio de um questionário, o qual foi aplicado a três indivíduos, sendo eles dois orientadores do LES os quais desempenham o papel de cliente, e um orientando o qual desempenha o papel de usuário (PESSINI, SANTANDER, 2013).

### 5.2 Utilizando o GQM para orientar a realização do estudo de caso.

Utilizando o GQM a fim de elaborar uma estrutura para a experimentação, obtém-se os seguintes dados pertinentes à proposta deste trabalho:

**Objetivo global:** estudar a técnica de modelagem organizacional i\* objetivando compreender se sua utilização auxilia na obtenção das saídas documento de conformidade, métricas de qualidade e *checklist* e na elaboração/utilização da matriz de priorização, a qual também impacta nas saídas métricas de qualidade e *checklist* de qualidade, onde estas saídas fazem parte do planejamento da qualidade do PMBOK.

**Objetivo do estudo:**

**Analisar →**

a eficácia da técnica de modelagem organizacional quando integrada a fase de planejamento da qualidade do PMBOK, mais especificamente na obtenção das saídas: documento de conformidade, identificação das métricas de qualidade e da criação do *checklist* de qualidade.

**Com a finalidade de →**

comparar as características entre abordagem com i\* e a abordagem clássica.

**Com respeito a →**

integração do modelo organizacional i\* ao planejamento da qualidade.

**Do ponto de vista dos →**

stakeholders.

**No contexto de →**

desenvolvimento de uma página web no âmbito acadêmico.

**Questões (Q) /Métricas (M):**

Q01: A abordagem facilita a compreensão do contexto do *software*?

M01: O(s) elemento(s) da abordagem que permite(m) maior compreensão do contexto.

Q02: A abordagem é mais clara ao representar requisitos do produto?

M02: O(s) elemento(s) da abordagem que permite(m) representar os requisitos do produto.

Q03: A abordagem facilita a obtenção de métricas de qualidade?

M03: O(s) elemento(s) da abordagem que permite(m) identificação de métricas de qualidade.

Q04: A abordagem apoia a elaboração de *checklists* de qualidade?

M04: O(s) elemento(s) da abordagem permite a obtenção de dado(s) para a elaboração de *checklists*.

Q05: O(s) elemento(s) da abordagem facilitam o entendimento do processo?

M05: O(s) elemento(s) visuais provenientes da utilização da técnica tornam os processos mais claros.

Q06: A abordagem auxilia na obtenção do documento de conformidade?

M06: O(s) elemento(s) da abordagem que auxiliam na realização do documento.

Q07: A abordagem auxilia na utilização da técnica de matriz de priorização?

M07: O(s) elementos(s) da abordagem que permite(m) auxiliar na elaboração e avaliação da matriz.

Q08: A abordagem considera diversas visões dos *stakeholders*?

M08: O(s) elemento(s) da abordagem que permite(m) expressar a(s) visão(ões) dos *stakeholders*.

Q09: A abordagem permite representação visual para verificar se um conjunto de etapas necessárias para satisfazer um requisito foi executado?

M09: O(s) elemento(s) da abordagem que permite(m) a representação visual para verificação.

Q10: A abordagem facilita a compreensão do contexto do projeto?

M01: O(s) elemento(s) da abordagem que permite(m) maior compreensão do contexto do projeto.

## 5.3 Aplicação da proposta

A aplicação da proposta é feita pelo analista de qualidade e deve ser realizada no objetivo de responder as questões elaboradas no GQM as quais servem para guiar a aplicação desta proposta. Esta proposta é aplicada seguindo os passos da mesma e ao final da aplicação desta os artefatos gerados servirão para responder as questões elaboradas por meio da abordagem GQM.

### 5.3.1 Passo 1: criar os modelos de dependências estratégicas

Para a aplicação desta proposta, foram utilizados dados levantados previamente pelo trabalho de (PESSINI; SANTANDER, 2013), na forma de documentos, onde os dados que foram obtidos serviram para uma elaboração parcial do modelo SD de requisitos funcionais do sistema. Então a partir dessa elicitação de requisitos de software previamente realizada, foi utilizado o passo 1 da proposta, para a construção do modelo SD do sistema para os requisitos de *software* e de qualidade (Diretrizes 1 e 2). Após o mapeamento dos objetivos ter sido realizado, o artefato gerado foi apresentado aos clientes, o qual ao ver o modelo, fizeram novas considerações e novos requisitos de *software* surgiram, essas entrevistas duraram cerca de uma hora e quarenta minutos. Após esta

nova validação dos requisitos da aplicação *web* para o LES por meio da análise do modelo SD feita pelos clientes o modelo foi atualizado. O artefato SD final dos requisitos de *software* gerado pode ser observado na Figura 23. Pode-se observar através deste diagrama SD da Figura 23 que foram mapeados três *stakeholders* que desempenham o papel de cliente, também foram mapeados os requisitos funcionais de *software* os quais foram obtidos através das entrevistas de cunho aberta, esses *stakeholders* esperavam ao carregar a aplicação *web* obter informações pertinentes a publicações, informações de projetos, os parceiros do grupo do laboratório de engenharia de *software*, os membros do grupo, além dos eventos, agenda do grupo e fazer *download* dos *frameworks* desenvolvidos pelo grupo.

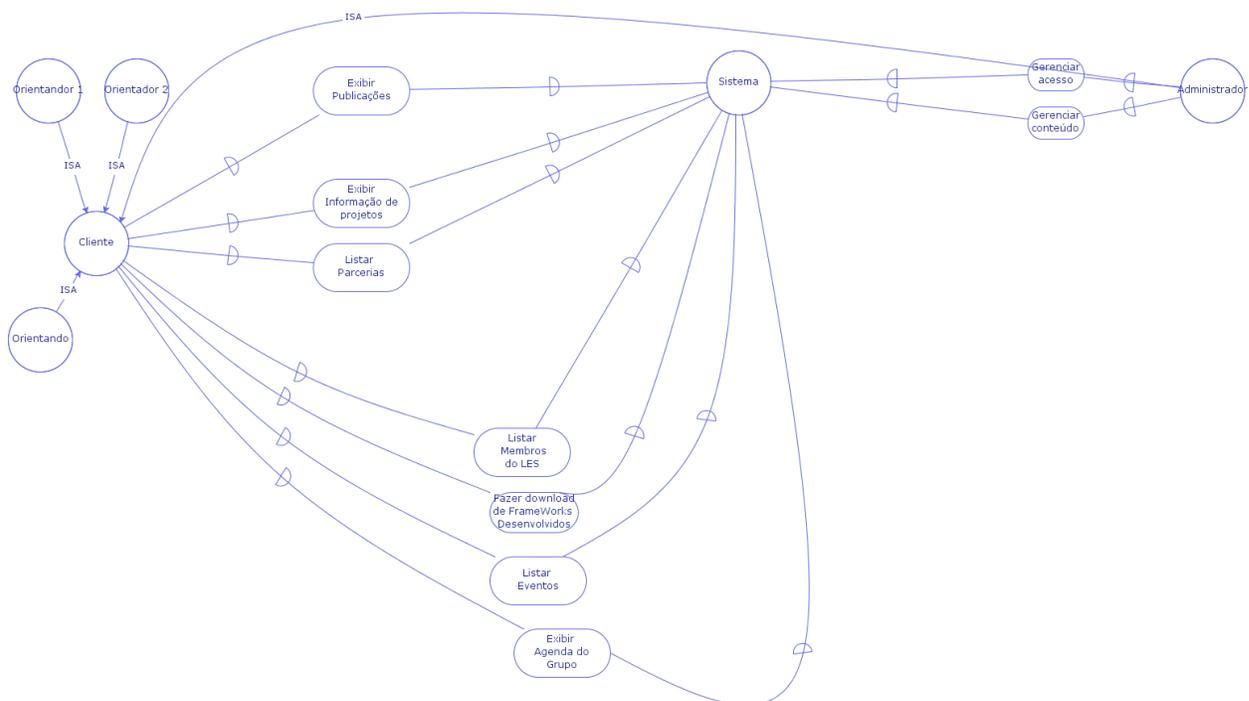


Figura 23: Modelo SD do sistema.

Posteriormente através de uma nova entrevista do tipo aberta, que durou duas horas e quinze minutos, foi possível identificar os atores elencados para o desenvolvimento do projeto, e por meio de entrevistas com cada um destes atores, as atividades de responsabilidade de cada ator puderam ser mapeadas. Na Figura 24 é apresenta-se o modelo SD do projeto. Neste diagrama pode-se

observar os atores envolvidos no desenvolvimento do produto encomendado bem como as funções exigidas de cada um deles, as quais foram mapeadas como dependência entre eles, como resultado disto obteve-se as tarefas, recursos e objetivos inerentes aos atores para que o planejamento fosse executado com sucesso. Sendo assim pode-se observar que o ator gerente de projeto tem dependência do tipo recurso com o programador, ele depende que o programa entregue versões executáveis da aplicação, o Gerente de projeto também depende do analista de sistemas para obter os requisitos não funcionais e funcionais, os quais servirão para a elaboração dos diagramas SD para requisitos funcionais de *software* e SD para os requisitos de qualidade, além destes dois ele ainda possui dependência com o analista de testes, para que este defina o caso de testes e com o analista de negócios para que esse informe os objetivos organizacionais e as regras de negócio. O analista de negócio por sua vez possui dependência com o cliente, sendo que ele necessita o cliente provenha a análise empresarial. Já o cliente depende do gerente de projeto para obter um produto com qualidade. O cliente ainda deve proporcionar ao analista de qualidade os requisitos/padrões de qualidade, avaliar os elementos das matrizes de priorização dos requisitos de *software* e de qualidade além de dar um *feedback* sobre a sua satisfação com o produto final. Nesse modelo ainda podemos observar que o analista de testes depende do ator testador para que ele execute as tarefas de *feedback* dos testes, realizar os testes e documentar os testes e o analista de testes ainda depende do programador para que ele realize as correções na aplicação.

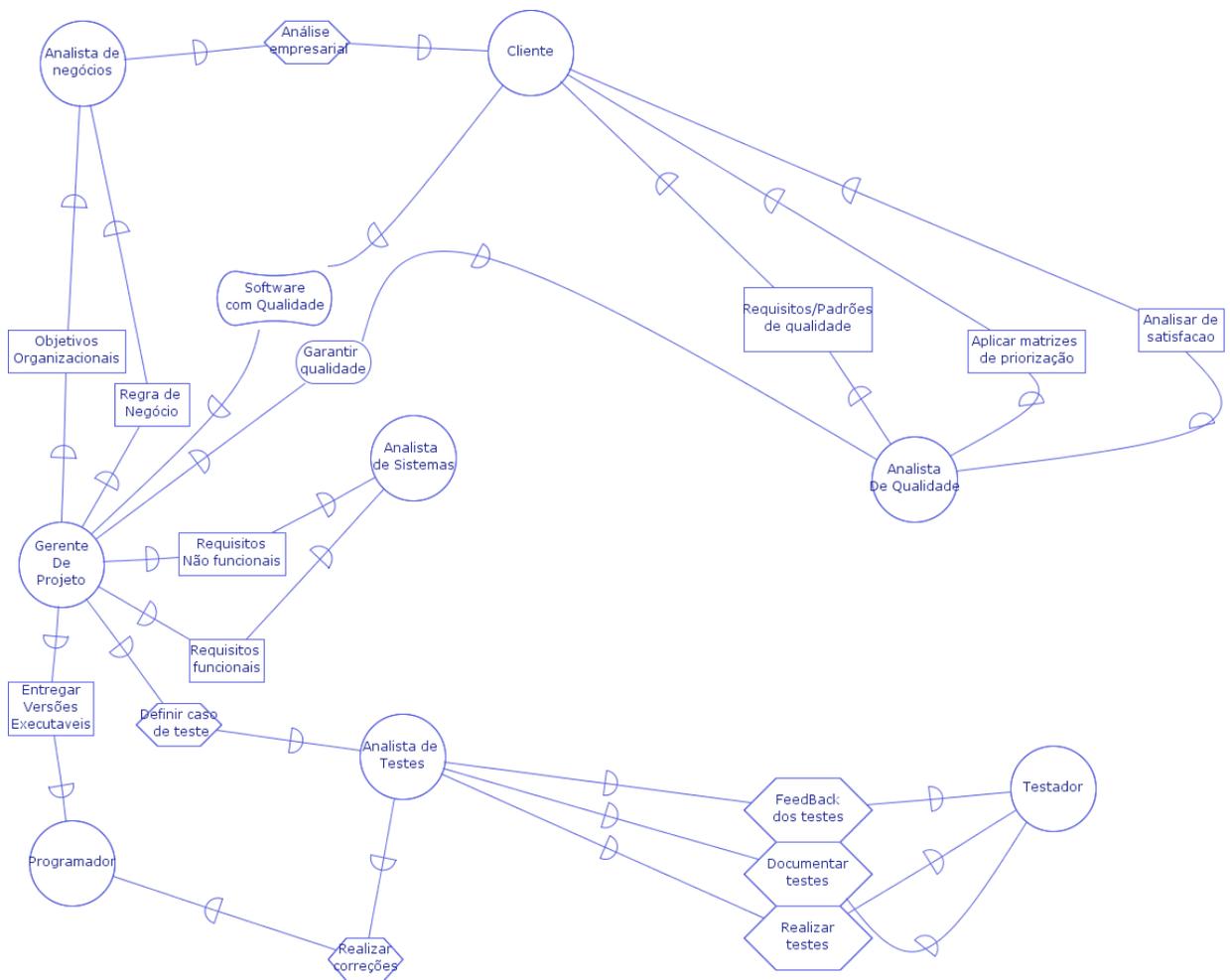


Figura 24: Modelo SD do projeto.

E por fim realizou-se uma nova entrevista com os clientes a fim de elicitar os requisitos de qualidade do sistema, esta entrevista durou cerca de uma hora e vinte e cinco minutos, então o modelo para os requisitos de qualidade foi gerado conforme pode ser observado na Figura 25.

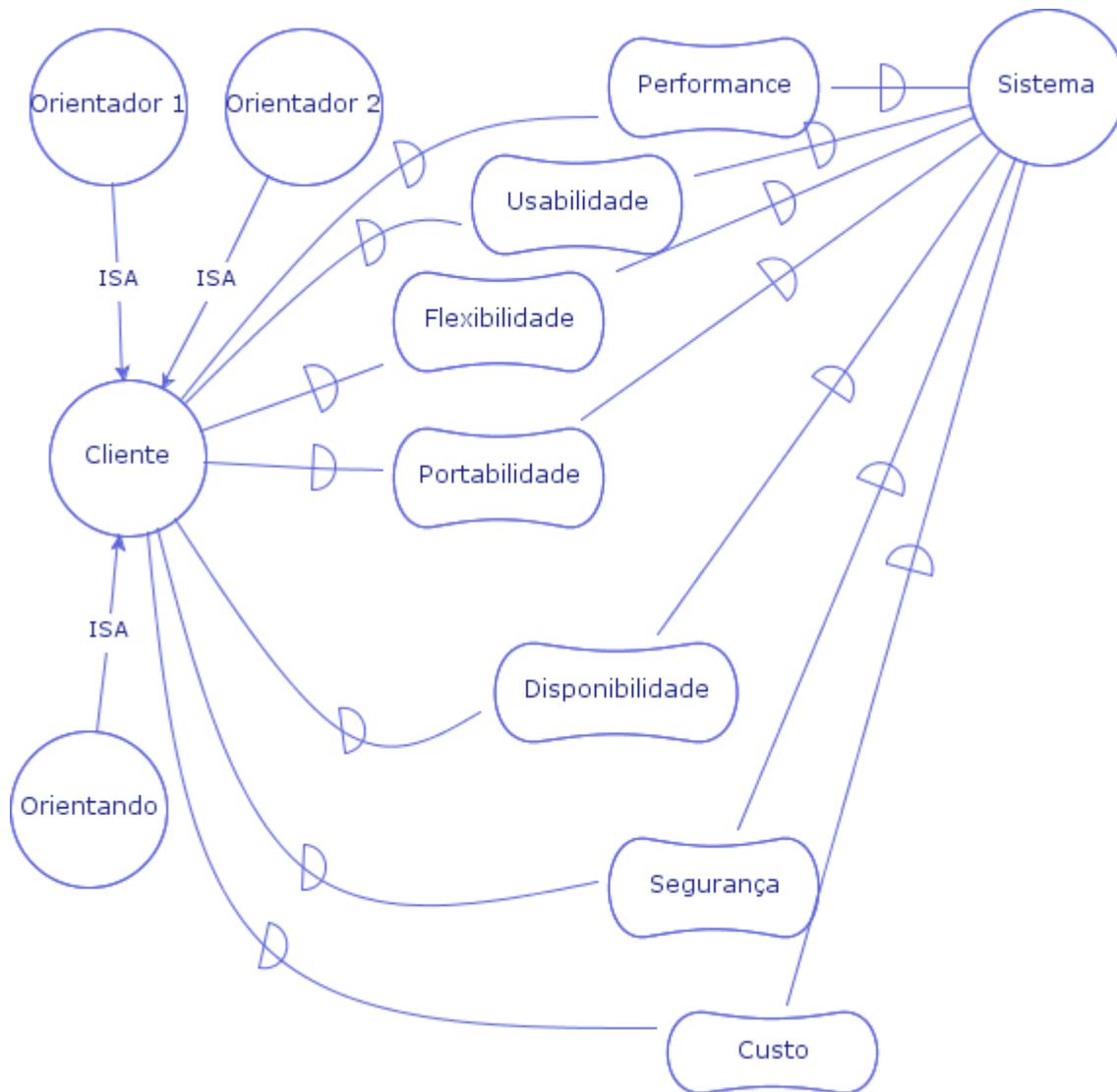


Figura 25: Modelo SD dos requisitos de qualidade do sistema.

A partir dos dados obtidos com a entrevista pode-se mapear sete requisitos de qualidade, os clientes esperavam que o sistema tivesse um custo de até R\$ 4800, este valor é referente ao total do pagamento da bolsa de iniciação científica, eles também desejavam que a aplicação *web* estivesse 99.9% do tempo online, esse requisito não funcional foi mapeado para disponibilidade, também pode-se observar que estes desejavam que apenas as pessoas com cadastro pudessem *logar* no site para fazer alteração no conteúdo, este requisito não funcional foi mapeado para o *dependum* segurança, outro requisito não funcional identificado é que a aplicação deveria ser fácil de ter seu conteúdo alterado/atualizado esse requisito foi mapeado para flexibilidade, outro requisito elicitado foi que a aplicação deveria ser possível de ser acessada de diferentes

navegadores e dispositivos portanto este foi mapeado para portabilidade, outro requisito não funcional obtido foi que a interface da aplicação deveria ser bem entendível e ter cores apropriadas tal requisito foi mapeado para usabilidade e o último requisito de qualidade identificado foi que a resposta do servidor deveria ser obtida em até dois segundos, então este requisito foi mapeado para performance.

### **5.3.2 Passo 2: criar os modelos de razões estratégicas**

Tendo como base os diagramas SD obtidos ao final do passo 1 como pode ser visto nas Figura 23 para o diagrama SD dos requisitos funcionais do *software*, Figura 24 para o modelo SD do projeto e Figura 25 para SD dos requisitos de qualidade, os atores identificados no diagrama SD do projeto foram novamente entrevistados a fim de obter um maior nível de detalhamento sobre o que era necessário para satisfazer os *dependums*, no modelo do projeto cada ator descreveu as atividades pertinentes a ele, essas entrevistas duraram cerca de uma hora e cinquenta minutos.

Na Figura 26, pode-se observar o modelo SR dos requisitos de *software* do sistema a elaboração deste e mapeamento dos *dependums* foram obtidos com base no modelo SD dos requisitos de *software* vide Figura 23. O modelo SR para o projeto que foi construído com base nos elementos presentes no diagrama SD do projeto vide Figura 24 e também com base nas entrevistas feitas com os atores identificados através deste mesmo diagrama SD pode ser visto na Figura 27.

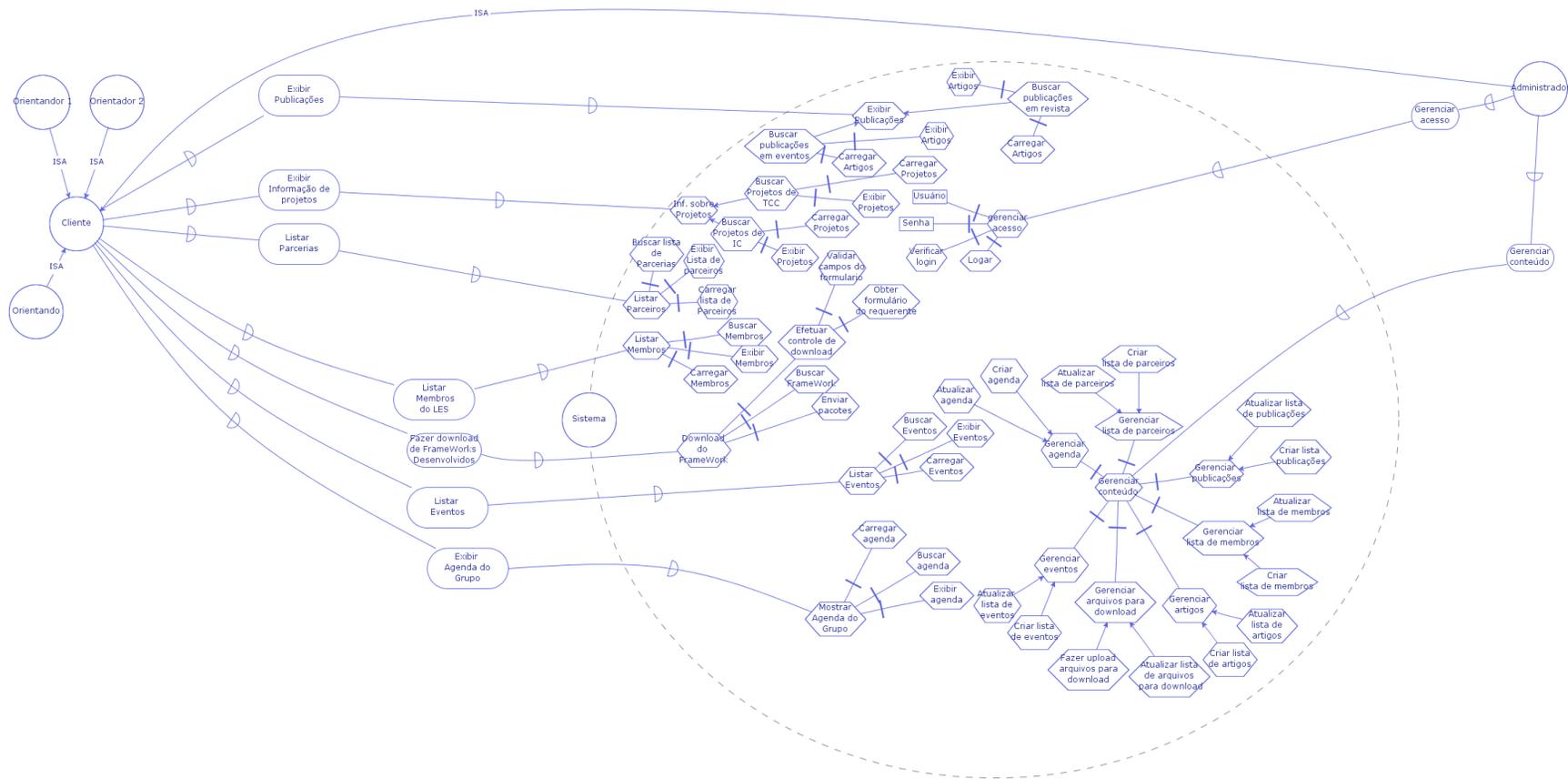


Figura 26: Modelo SR dos requisitos de *software* sistema.

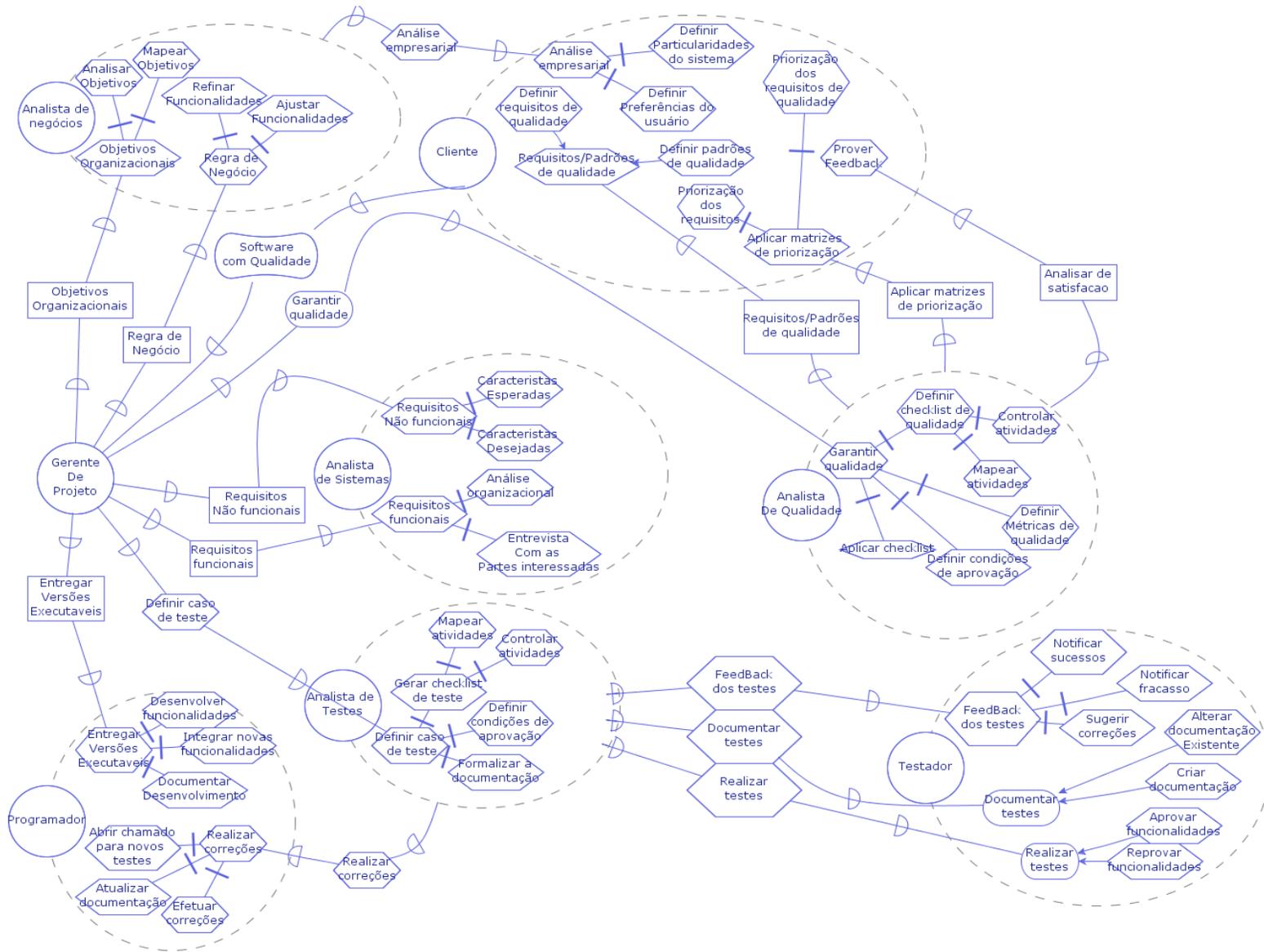


Figura 27: Modelo SR do projeto.

E o modelo SR para requisitos de qualidade o qual foi criado com base nos *dependums* encontrados no diagrama SD dos requisitos de qualidade vide Figura 25 e também por meio das entrevistas realizadas com os clientes é retratado na Figura 28.

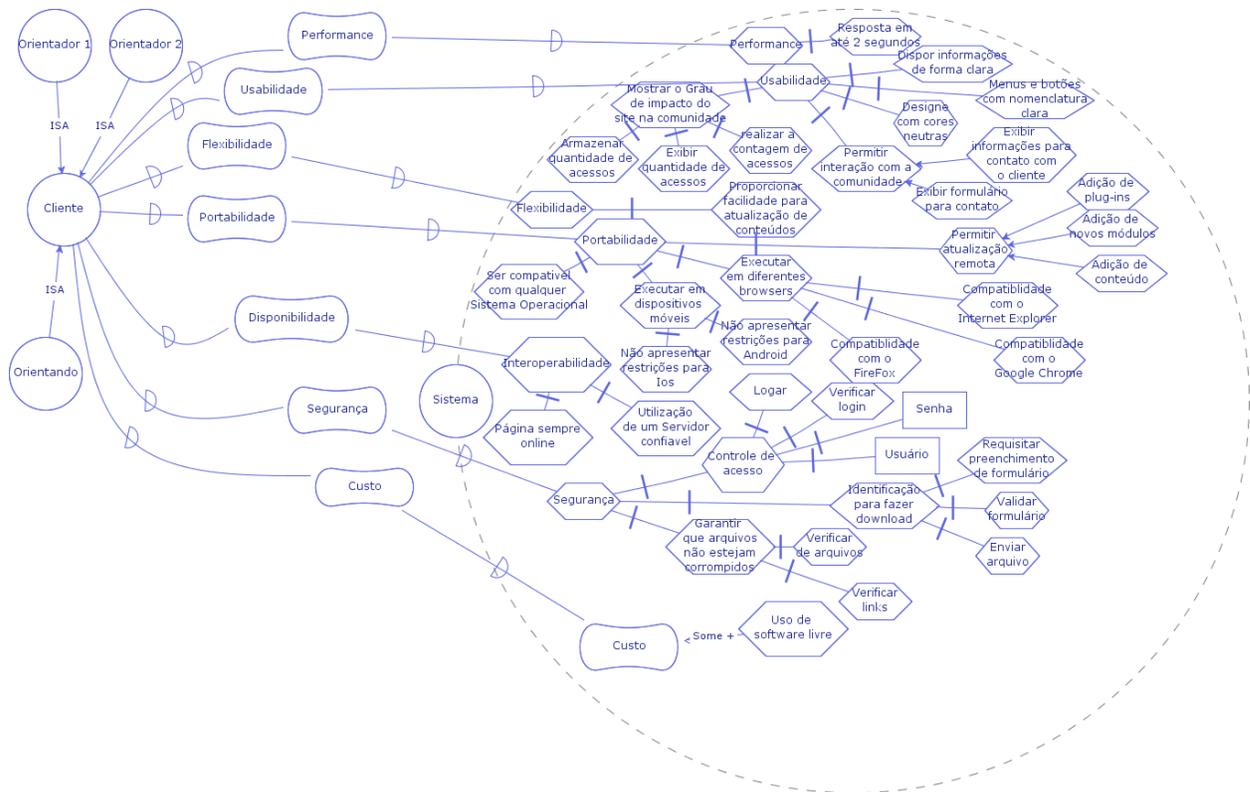


Figura 28: Modelo SR dos requisitos de qualidade do sistema.

### 5.3.3 criar a matriz de priorização dos stakeholders

Analisando o documento de (PESSINI; SANTANDER, 2013) pode-se observar que há três *stakeholders*, sendo eles o Orientador 1 o qual desenvolve o papel de Cliente que encomenda o sistema, o Orientador 2 que representa o papel de Cliente minoritário e por fim o Orientando o qual incorpora o papel de usuário geral do sistema. Esses são os *stakeholders* importantes para este passo. Os atores retratados no modelo SD assumiram o mesmo papel de cliente, porém eles possuem visões diferentes, sendo assim, para a elaboração desta matriz de priorização os elementos a serem considerados são os atores identificados, no caso, o Orientador 1 e 2 e o Orientando. Então

a partir desta análise eles são dispostos ao longo do eixo horizontal e vertical e em seguida são avaliados pela equipe do projeto. A avaliação foi feita da seguinte forma, o ator Orientador 1 é o cliente que mais influência na especificação dos requisitos, além de ser o cliente pagante, o Orientador 2 por sua vez, auxilia na especificação dos requisitos e o Orientando faz o papel do usuário, ou seja, tem mais influência que o Orientador 2, por este representar o usuário final o qual tem interesse em acessar as informações da aplicação *web*. O artefato gerado conforme o passo 3 pode ser visto na Tabela 5.1, essa priorização foi feita pelo analista de qualidade em conjunto com o gerente de projeto.

Tabela 5.1: Matriz de priorização de stakeholders.

Priorização de stakeholders	Orientador 1	Orientador 2	Orientando	Total da linha	Total ponderado
Orientador 1		5	5	10	0,666
Orientador 2	1/5		1/5	0,2	0,013
Orientando	1/5	5		5	0,333
			Total	15,02	

#### 5.4.4 criar as matrizes de priorização de requisitos

Tanto a matriz de requisito de software quanto a de requisitos de qualidade são construídas a partir dos modelos SD do sistema, onde os *dependums* são elementos a serem avaliados vide Figura 21a e Figura 21b do capítulo 4 seção 4.3.4. Assim, observando as dependências na Figura 23 e Figura 25 entre os atores e o sistema. A partir dos diagramas SD dos requisitos de qualidade e dos requisitos de *software* são montadas as matrizes, onde todos os *dependums* identificados nesses diagramas serão mapeados para as suas respectivas matrizes de priorização.

Na Tabela 5.2 pode ser observada a matriz de priorização de requisitos do software, nesta pode-se observar que os elementos contidos ao longo do eixo vertical/horizontal são os requisitos funcionais que estão mapeados no diagrama SD como pode ser visto na Figura 23. O mesmo acontece para os requisitos da qualidade que podem ser obtidos por meio do diagrama SD da Figura

25 estes *dependums* encontrados no diagrama SD são mapeados para a matriz de priorização dos requisitos da qualidade assim como está ilustrado na Tabela 5.3. O número de matrizes de priorização desses dois tipos criadas nesse passo é igual ao número de matrizes no caso duas, sendo uma para priorizar os requisitos de *software* e outra para os requisitos de qualidade multiplicado pelo número de *stakeholders* identificados na matriz de priorização de *stakeholders* referente ao passo três.

Tabela 5.2 Matriz de priorização de requisitos de software

Priorização de requisitos de software	Exibir publicações	Exibir informações de projetos	Listar Parcerias	Listar membros do Les	Fazer download de frameworks desenvolvidos	Listar eventos	Exibir agenda do grupo	Gerenciar acesso	Gerenciar conteúdo	Total da linha	Total ponderado
Exibir Publicações											
Exibir Informação de projetos											
Listar parcerias											
Listar membros do LES											
Fazer download de frameworks desenvolvidos											
Listar Eventos											
Exibir agenda do grupo											
Gerenciar acesso											
Gerenciar conteúdo											
									Total		

Tabela 5.3 Matriz de priorização de requisitos de qualidade.

Priorização dos requisitos de qualidade segundo a visão do Orientador 1	Performance	Usabilidade	Flexibilidade	Portabilidade	Disponibilidade	Segurança	Custo	Total	Total ponderado
Performance									
Usabilidade									
Flexibilidade									
Portabilidade									
Interoperabilidade									
Segurança									
Custo									
							Total		

#### **5.4.5 aplicar a matriz de priorização de requisitos aos atores**

A fim de auxiliar os *stakeholders* na avaliação dos elementos de ambas as matrizes, os diagramas SR gerados para os requisitos do software e para os requisitos da qualidade, foram novamente apresentados aos *stakeholders*, esses foram identificados por meio da matriz de priorização de *stakeholders* que foi realizada no passo 3 dessa proposta, a fim de garantir que os elementos fossem avaliados de forma fidedigna. Essa atividade teve duração total de uma hora e meia. Os resultados obtidos para priorização dos requisitos de software podem ser vistos nas Tabelas 5.4, 5.5 e 5.6, as quais consideram a visão do Orientador 1, Orientador 2 e Orientando, respectivamente. Os resultados obtidos para priorização dos requisitos da qualidade podem ser vistos nas Tabelas 5.7, 5.8 e 5.9.

Tabela 5.4: priorização dos requisitos de software na visão do orientador 1.

Priorização de requisitos visão do Orientador 1	Exibir publicações	Exibir informações de projetos	Listar Parcerias	Listar membros do Les	Fazer download de frameworks desenvolvidos	Listar eventos	Exibir agenda do grupo	Gerenciar acesso	Gerenciar conteúdo	Total da linha	Total ponderado
Exibir Publicações		1	1	5	1	1	1/5	1/5	1	10,4	0,067
Exibir Informação de projetos	1		1	1	1/5	1/5	1	1/5	1/5	4,8	0,031
Listar parcerias	1	1		1	1/5	1/5	1/5	1/5	1/5	4	0,026
Listar membros do LES	1/5	1	1		1/5	1/5	1/5	1/5	1/5	3,2	0,021
Fazer download de frameworks desenvolvidos	1	5	5	5		1	5	1/5	1/5	22,4	0,144
Listar Eventos	1	5	5	5	1		5	1/5	1/5	22,4	0,144
Exibir agenda do grupo	5	1	5	5	1/5	1/5		1/5	1/5	16,8	0,108
Gerenciar acesso	5	5	5	5	5	5	5		1/5	35,2	0,227
Gerenciar conteúdo	1	5	5	5	5	5	5	5		36	0,232
									Total	155,2	

Como pode ser observado na Tabela 5.4 a priorização que foi realizada pelo cliente Orientador 1 demonstra que para ele o requisito de *software* gerenciar conteúdo é prioritário, seguido do requisito Gerenciar acesso, posteriormente do requisito fazer *download* de *frameworks* desenvolvidos, depois listar eventos, exibir agenda do grupo e os requisitos menos prioritários segundo este cliente foram exibir publicações, exibir informação de projetos, listar parcerias do grupo e listar os membros do grupo de pesquisa de engenharia de *software*.

Essas observações podem ser feitas com base no total de cada linha dividido pelo total da coluna dos totais das linhas, analisando os valores estabelecidos pelo Orientador 1 pode-se notar que ele não considerou que algum elemento pudesse ser muito mais importante que algum outro nem o inverso disso, dando a impressão de considerar todos os elementos importantes para o produto encomendado.

Já segundo a visão do Orientador 2 pode-se notar que há elementos que este considera como muito menos importante e muito mais importante, dando a ideia de que há elementos que não seriam essenciais para o produto final, segundo a análise que a Tabela 5.5 nos permite fazer, pode-se notar que o cliente Orientador 2 considera exibir publicações como requisito prioritário seguido de exibir informação de projetos, isso nos dá ideia de que ele está mais preocupado em poder acessar informações atualizadas sobre as publicações e os projetos que foram/vêm sendo desenvolvidos. A ordem de priorização dos requisitos de *software* segundo o orientador 2 pode ser obtida observando a coluna de total ponderado conforme pode ser visto na Tabela 5.5.

Analisando a matriz de priorização segundo a visão do Orientando pode-se notar que este tem uma visão parecida com a do orientador 2, também tendo exibir publicações como requisito prioritário, mas também há um ponto que é parecido com a avaliação do orientador 1, onde o orientando avaliou apenas um elemento como sendo muito mais importante que outro e conseqüentemente apenas um sendo muito menos importante. A ordem de priorização pode ser identificada analisando a coluna do total ponderado.

Tabela 5.5: priorização dos requisitos de software na visão do orientador 2.

Priorização de requisitos visão do Orientador 2	Exibir publicações	Exibir informações de projetos	Listar Parcerias	Listar membros do Les	Fazer download de frameworks desenvolvidos	Listar eventos	Exibir agenda do grupo	Gerenciar acesso	Gerenciar conteúdo	Total da linha	Total ponderado
Exibir Publicações		1	10	5	10	10	5	10	5	56	0,301
Exibir Informação de projetos	1		5	5	5	5	5	10	5	41	0,221
Listar parcerias	1/10	1/5		1	1	1	1/5	5	1/5	8,7	0,047
Listar membros do LES	1/5	1/5	1		1	1	1/5	5	1/5	8,8	0,047
Fazer download de frameworks desenvolvidos	1/10	1/5	1	1		5	1	5	1/5	13,5	0,073
Listar Eventos	1/10	1/5	1	1	1/5		1/5	1	1/5	3,9	0,021
Exibir agenda do grupo	1/5	1/5	5	5	1	5		5	1/5	21,6	0,116
Gerenciar acesso	1/10	1/10	1/5	1/5	1/5	1	1/5		1/5	2	0,011
Gerenciar conteúdo	1/5	1/10	5	5	5	5	5	5		30,3	0,163
									Total	185,8	

Tabela 5.6: priorização dos requisitos de software na visão do orientando.

Priorização de requisitos visão do Orientando	Exibir publicações	Exibir informações de projetos	Listar Parcerias	Listar membros do Les	Fazer download de frameworks desenvolvidos	Listar eventos	Exibir agenda do grupo	Gerenciar acesso	Gerenciar conteúdo	Total da linha	Total ponderado
Exibir Publicações		1/5	10	5	1	5	5	1	1	28	0,202
Exibir Informação de projetos	5		5	5	1/5	1/5	1	1	1	18	0,132
Listar parcerias	1/10	1/5		1/5	1/5	1/5	1/5	1/5	1/5	2	0,011
Listar membros do LES	5	1/5	5		1/5	1/5	1/5	1	1	13	0,092
Fazer download de frameworks desenvolvidos	1	5	5	5		1	1	1	1	20	0,144
Listar Eventos	1/5	5	5	5	1		1	1	1	19	0,138
Exibir agenda do grupo	1/5	1	5	5	1	1		1	1	15	0,109
Gerenciar acesso	1	1	5	1	1	1	1		1	12	0,086
Gerenciar conteúdo	1	1	5	1	1	1	1	1		12	0,086
									Total	139,3	

Como pode ser observado na Tabela 5.7 a priorização dos requisitos de qualidade que foi realizada pelo cliente Orientador 1 demonstra que para ele o requisito de disponibilidade é prioritário a ordem de priorização segundo a visão deste cliente pode ser obtida analisando-se a coluna de total ponderado da Tabela 5.7.

Novamente analisando os valores estabelecidos pelo Orientador 1 pode-se notar que ele não considerou que algum elemento pudesse ser muito mais importante que algum outro nem o inverso disso, dando a impressão de considerar todos os elementos importantes para o produto encomendado.

Já segundo a visão do Orientador 2 pode-se notar que há elementos que este considera como muito menos importante e muito mais importante, dando a ideia de que há elementos que não seriam essenciais para o produto final, segundo a análise que a Tabela 5.8 nos permite fazer, pode-se notar que o cliente Orientador 2 considera performance e usabilidade como requisitos prioritários, isso nos dá ideia de que ele está mais preocupado com a navegação rápida e simplificada na aplicação *web*. A ordem de priorização dos requisitos de qualidade segundo o orientador 2 pode ser obtida observando a coluna de total ponderado conforme pode ser visto na Tabela 5.8.

Analisando a matriz de priorização segundo a visão do Orientando pode-se notar que este também tem usabilidade como requisito prioritário. A ordem de priorização pode ser identificada analisando a coluna do total ponderado como pode ser visto na Tabela 5.9.

Tabela 5.7: priorização dos requisitos da qualidade na visão do cliente

Priorização dos requisitos de qualidade segundo a visão do Orientador 1	Performance	Usabilidade	Flexibilidade	Portabilidade	Disponibilidade	Segurança	Custo	Total	Total ponderado
Performance		1	5	1	1/5	1	1/5	8,4	0,095
Usabilidade	1		5	1	1/5	5	1/5	12,4	0,140
Flexibilidade	1/5	1/5		1	1/5	1	1/5	2,8	0,032
Portabilidade	1	1	1		1/5	1	1/5	4,4	0,050
Disponibilidade	5	5	5	5		5	1	26	0,294
Segurança	1	1/5	1	1	5		1/5	8,4	0,095
Custo	5	5	5	5	1	5		26	0,294
							Total	88,4	

Tabela 5.8: priorização dos requisitos da qualidade na visão do orientador 2

Priorização dos requisitos de qualidade segundo a visão do Orientador 2	Performance	Usabilidade	Flexibilidade	Portabilidade	Disponibilidade	Segurança	Custo	Total	Total ponderado
Performance		1	10	5	5	5	10	36	0,246
Usabilidade	1		10	5	5	5	10	36	0,246
Flexibilidade	1/10	1/10		1	10	1	10	22,2	0,152
Portabilidade	1/5	1/5	1		10	1	10	22,4	0,153
Disponibilidade	1/10	1/10	1/5	1/5		1/5	1	1,8	0,012
Segurança	1/5	1/5	1	1	10		5	17,4	0,119
Custo	1/10	1/10	1/10	1/10	10	1/5		10,6	0,072
							Total	146,4	

Tabela 5.9: priorização dos requisitos da qualidade na visão do cliente orientando.

Priorização dos requisitos de qualidade segundo a visão do Orientando	Performance	Usabilidade	Flexibilidade	Portabilidade	Disponibilidade	Segurança	Custo	Total	Total ponderado
Performance		1/5	1	5	1	5	1/5	12,4	0,140
Usabilidade	5		5	10	5	1	1	27	0,305
Flexibilidade	1	1/5		1	1	1/5	1/5	3,6	0,041
Portabilidade	1/5	1/10	1		1	1/5	1/5	2,7	0,031
Disponibilidade	1	1/5	1	1		1/5	1/5	3,6	0,041
Segurança	1/5	1	5	5	5		1	17,2	0,194
Custo	5	1	5	5	5	1		22	0,249
							Total	88,5	

### 5.4.6 gerar matriz conjunta de priorização de requisitos com a influência de cada ator

Uma vez que todos os atores orientador 1, orientador 2 e orientando responderam as matrizes de priorização, o próximo passo é gerar a matriz final de priorização, a qual considera a influência de cada ator. Então os valores são obtidos multiplicando o total ponderado que o *stakeholder* recebeu ao final do passo 3 com os totais ponderados dos elementos que ele avaliou, vide Figura 22 do capítulo 4 seção 4.3.6. Na Tabela 5.10 pode-se observar o resultado da priorização de requisitos do sistema. Por exemplo, multiplicando o valor da priorização do *stakeholder* orientador 1, no caso 0.666, pela priorização que este fez sobre os requisitos que ele avaliou, tendo como exemplo o requisito de *software* que ele considerou prioritário no caso o gerenciar conteúdo que recebeu valor de 0.232, então na matriz conjunta teríamos para linha gerenciar conteúdo com a coluna orientador 1 o valor  $0.154 = 0.232 * 0.666$ , fazendo isso para todos os requisitos de *software* e *stakeholders*

Tabela 5.10: Matriz conjunta de priorização dos requisitos de software.

Matriz conjunta de priorização dos requisitos de software	Orientador 1	Orientador 2	Orientando	Total da linha	Total ponderado
Gerenciar conteúdo	0,154	0,002	0,029	0,1853	0,183
Gerenciar acesso	0,151	0	0,029	0,1799	0,178
Fazer download de frameworks desenvolvidos	0,096	0,001	0,048	0,1449	0,143
Listar Eventos	0,096	0	0,046	0,1423	0,141
Exibir Publicações	0,045	0,004	0,067	0,116	0,115
Exibir agenda do grupo	0,072	0,002	0,036	0,1099	0,109
Exibir Informação de projetos	0,021	0,003	0,044	0,0675	0,067
Listar membros do LES	0,014	0,001	0,031	0,0449	0,044
Listar parcerias	0,017	0,001	0,004	0,0214	0,021
			Total	1,012	

Então como pode ser observado na Tabela 5.10 o requisito gerenciar conteúdo é considerado prioritário devido a avaliação dos três *stakeholders*, mas principalmente devido à influência do *stakeholder* orientador 1 no produto encomendado. A mesma analogia pode ser feita sobre a matriz conjunta dos requisitos de qualidade como pode ser vista na Tabela 5.11.

Tabela 5.11: Matriz conjunta de priorização dos requisitos da qualidade.

Matriz conjunta de priorização dos requisitos da qualidade	Orientador 1	Orientador 2	Orientando	total da linha	total ponderado
Custo	0,196	0,001	0,083	0,28	0,276
Disponibilidade	0,196	0	0,014	0,21	0,207
Usabilidade	0,093	0,003	0,102	0,198	0,196
Segurança	0,063	0,002	0,065	0,13	0,128
Performance	0,063	0,003	0,047	0,113	0,112
Portabilidade	0,033	0,002	0,01	0,045	0,045
Flexibilidade	0,021	0,002	0,014	0,037	0,036
			Total	1,012	

#### 5.4.7 identificar as métricas de qualidade

As métricas e critérios apresentados na Tabela 5.12 foram obtidos a partir de uma análise do diagrama SR dos requisitos de qualidade (ver Figura 28), uma vez que a matriz conjunta de priorização dos requisitos da qualidade (ver Tabela 5.11) não mostrou nenhuma inconsistência na priorização e como não houve restrição de tempo ou de recurso que necessitasse a exclusão de algum elemento priorizado dessa matriz, pode-se então elaborar o artefato de métricas de qualidade, baseando-se no SR dos requisitos de qualidade a fim de descrever os critérios para avaliar as métricas. Este processo consiste basicamente em observar a satisfação de cada requisito de qualidade pelo ator sistema. Por exemplo analisando o diagrama SR da Figura 28 podemos descrever a métrica “a página web deve ser compatível com diferentes navegadores” isso pode ser

dito pois analisando o *dependum* portabilidade nota-se que para a satisfação deste, com forme as sub-tarefas nos indicam, é necessário que a página web possa ser exibida no navegador chrome, Firefox e internet explorer, sendo assim podemos também estabelecer o critério para essa métrica, que no caso é descrita como a página deve ser capaz de ser exibida em ao menos três diferentes navegadores.

Tabela 5.12: Métricas de qualidade e critérios de aceitação.

Métricas	Crítérios
A página web deve ter um custo baixo (Custo).	O custo no final do projeto tem que ser menor igual R\$4800,00.
A página deve sempre estar online (Disponibilidade).	Analisar logs do servidor e verificar se o tempo em que a página ficou disponível equivale a 99.9% do tempo desde que ela foi implantada.
A página deve ser de fácil interação (Usabilidade).	Para que a página seja considerada usual os seguintes pontos precisam estar presentes: Ter informações bem dispostas com textos de menu autoexplicativos e nomenclaturas claras; cores neutras; permitir contato com o cliente e ter contador de acesso.
A página deve requisitar um login válido para modificação de conteúdo (Segurança).	Os acessos de logins que não tenham permissão para modificar o conteúdo devem ser igual a 0.
Obter resposta rápida do servidor (Performance)	As páginas devem ser carregadas em tempo menor igual a 2 segundos.
A página web deve ser compatível com diferentes navegadores (Portabilidade)	Á página deve ser capaz de ser exibida em ao menos três diferentes navegadores.

#### **5.4.8 validação dos artefatos**

Os artefatos produzidos até aqui foram apresentados aos clientes identificados por meio da matriz de priorização feita no passo 3 desta proposta e validados pelas partes interessadas esta reunião para apresentar os artefatos durou um total de uma hora e vinte minutos. Mas vale ressaltar que esse passo foi executado duas vezes, onde na primeira surgiram algumas considerações, o que resultou na alteração dos diagramas SD e SR do sistema e consequentemente também das matrizes de priorização de requisitos. Sendo assim os artefatos apresentados nas subseções anteriores, são os que passaram deste ponto.

#### **5.4.9 verificação da priorização dos requisitos**

Analisando as matrizes de priorização conjuntas, a equipe do projeto, mas especificamente o gerente de projeto e o analista de qualidade, não observaram nenhum caso em que houvesse um requisito mais acima na lista de priorização que afetasse a execução/implementação de um que estivesse mais a baixo.

#### **5.4.10 gerar documento de conformidade**

Este documento é criado com base no diagrama SR do projeto (Figura 27), o qual determina as responsabilidades e tarefas associadas a cada ator. Esse documento pode ser consultado sempre que necessário a fim de acompanhar se o projeto está sendo executado como deve. O resultado pode ser observado na Tabela 5.13. Observando esta tabela pode-se verificar que as tarefas exigidas de cada ator encontrado no modelo SR do projeto (ver Figura 27) estão descritas na tabela, como por exemplo, analisou-se que no modelo SR o ator programador era *dependee* de uma dependência entre o gerente de projetos, a dependência existente entre eles era a necessidade que o ator gerente de projetos tinha com o ator programador para que este entregasse versões executáveis do produto encomendado, ainda por meio do diagrama SR pode-se analisar as sub-tarefas que o programador deveria desenvolver a fim de satisfazer o *dependum*.

Sendo assim, este documento foi construído analisando o diagrama SR do projeto afim de identificar todos os atores que comporiam a coluna Responsável da Tabela 5.13. Então após a identificação de todos os responsáveis, procurou-se analisar o modelo SR afim de obter as atividades que cada um dos responsáveis encontrados deveriam executar, essas atividades foram encontradas no modelo como sendo os *dependums* ou seja, as dependências que os atores responsáveis possuíam e por fim para a coluna como será realizada, buscou-se no modelo SR pelo

desmembramento desses *dependums*, mapeando-se para esta tabela todas as sub-atividades necessárias para a conclusão de um *dependum*.

Tabela 5.13: Documento de conformidade do processo.

Responsável	Atividade	Como será realizada
Programador	Entregar versões executáveis	<ol style="list-style-type: none"> <li>1. Desenvolver funcionalidades</li> <li>2. Integrar novas funcionalidades.</li> <li>3. Documentar Desenvolvimento</li> </ol>
Programador	Realizar correções	<ol style="list-style-type: none"> <li>1. Abrir chamado para novos testes</li> <li>2. Atualizar documentação</li> <li>3. Efetuar correções</li> </ol>
Analista de testes	Definir caso de teste	<ol style="list-style-type: none"> <li>1. Definir condições de aprovação</li> <li>2. Gerar checklist de teste               <ol style="list-style-type: none"> <li>2.1 Mapear atividades</li> <li>2.2 Controlar atividades</li> </ol> </li> <li>3. Formalizar a documentação</li> </ol>
Testador	FeedBack dos testes	<ol style="list-style-type: none"> <li>1. Notificar sucessos</li> <li>2. Notificar fracassos</li> <li>3. Sugerir correções</li> </ol>
Testador	Documentar testes	<ol style="list-style-type: none"> <li>1. Criar documentação ou alterar documentação existente.</li> </ol>
Testador	Realizar testes	<ol style="list-style-type: none"> <li>1. Aprovar funcionalidades ou Reprovar funcionalidades.</li> </ol>
Analista de sistemas	Prover requisitos funcionais	<ol style="list-style-type: none"> <li>1. Descrever características esperadas.</li> <li>2. Descrever características desejadas.</li> </ol>
Analista de sistemas	Prover requisitos não funcionais	<ol style="list-style-type: none"> <li>1. Realizar análise organizacional.</li> <li>2. Realizar entrevista com as partes interessadas</li> </ol>
Analista de qualidade	Garantir a qualidade	<ol style="list-style-type: none"> <li>1. Definir checklist de qualidade</li> </ol>

		<ol style="list-style-type: none"> <li>1.1 Mapear as atividades.</li> <li>1.2 Controlar as atividades.</li> <li>2. Definir métrica de qualidade.</li> <li>3. Definir condições de aprovação</li> <li>4. Aplicar o checklist.</li> </ol>
Analista de negócios	Prover objetivos organizacionais	<ol style="list-style-type: none"> <li>1. Analisar os objetivos organizacionais.</li> <li>2. Mapear os objetivos organizacionais.</li> </ol>
Analista de negócios	Prover regras de negócio	<ol style="list-style-type: none"> <li>1. Refinar funcionalidades.</li> <li>2. Ajustar funcionalidades.</li> </ol>
Cliente	Prover análise empresarial	<ol style="list-style-type: none"> <li>1. Definir particularidades do sistema.</li> <li>2. Definir preferências do usuário.</li> </ol>
Cliente	Prover requisitos/padrões de qualidade	<ol style="list-style-type: none"> <li>1. Definir os requisitos de qualidade ou os padrões de qualidade.</li> </ol>
Cliente	Prover a priorização dos requisitos	<ol style="list-style-type: none"> <li>1. Priorizar os requisitos do sistema.</li> <li>2. Priorizar os requisitos de qualidade.</li> </ol>
Cliente	Prover feedback	<ol style="list-style-type: none"> <li>1. Fornecer informações sobre satisfação ou relatório de bugs.</li> </ol>

#### 5.4.11 gerar *checklist* de qualidade

O checklist de qualidade é elaborado com base nos requisitos identificados por meio da matriz conjunta de priorização de requisitos de qualidade (ver Tabela 5.11) e do diagrama SR dos requisitos de qualidade (ver Figura 28) do sistema e pode ser visto na Tabela 5.14. Lembrando que o checklist também pode ser aplicado aos requisitos de software da mesma maneira. A Tabela 5.14 foi elaborada com base na análise da matriz de priorização e posteriormente do modelo SR dos

requisitos da qualidade (ver Figura 27), a análise da matriz de priorização serviu para indicar quais requisitos ainda permaneciam no planejamento do produto, após esta identificação analisou-se o modelo SR do projeto onde os dependums encontrados neste modelo foram mapeados para a coluna Requisito. A coluna satisfeito desta tabela no momento é meramente ilustrativa, pois como dito anteriormente esta saída é parte da atividade de planejar qualidade, porém ela é entrada da atividade de garantia da qualidade que é uma das três atividades do gerenciamento da qualidade (PMBOK, 2008). Mas no caso em que ela estivesse sendo utilizada neste momento o critério para dizer se a linha desta coluna recebe SIM ou NÃO é apoiado pela a análise do diagrama SR dos requisitos da qualidade, onde pode-se buscar pelas sub-atividades relativas a cada Requisito mapeado nessa tabela a fim de investigar se essas sub-atividades foram concluídas, uma vez que todas as sub-atividades necessárias tenham sido concluídas então o requisito pode ser validado.

Tabela 5.14: Checklist de qualidade.

Requisito	Satisfeito?
Performance	Sim
Usabilidade	Sim
Flexibilidade	Sim
Portabilidade	Sim
Disponibilidade	Sim
Segurança	Sim
Custo	Sim

## 5.4 Análise da abordagem

Após a fase de planejamento e definição dos objetivos, questões e métricas, a coleta de dados da experimentação foi realizada. A etapa seguinte foi a interpretação, nesta os dados foram absorvidos, com eles as questões definidas na seção 5.3 puderam ser respondidas como pode ser visto na Tabela 6 e as conclusões extraídas. Foram analisados os dados presentes na Tabela 6 e então uma avaliação foi elaborada. Conforme descrito na próxima seção.

## 5.5 Discussão

Observando os resultados alcançados no estudo de caso realizado na Seção 5.4, algumas considerações acerca da utilização do *framework* i\* e das matrizes de priorização L-shaped puderam ser obtidas.

O *framework* i\* mostrou-se útil para auxiliar na tarefa de construção das matrizes de priorização. Através da diagramação dos modelos SD e SR foi possível notar a utilidade das representações visuais provenientes dos elementos mapeados nesses diagramas, o que tornou mais rápido e simples a identificação dos elementos que foram utilizados para elaborar as matrizes de priorização tanto para os requisitos de *software* quanto para os de qualidade e para a de *stakeholders*. Acredita-se que a aceitação deste para problemas que envolvem um número maior de elementos e *stakeholders* seja ainda maior, uma vez que a utilização dos diagramas possibilita uma análise mais rápida do contexto geral, o que sem eles seria feito por meio da análise de uma grande quantidade de dados descritos textualmente, sendo isto algo útil tanto para a equipe do projeto quanto para os *stakeholders*.

A matriz de priorização L-shaped por sua vez, também mostrou-se eficaz e o processo para a sua elaboração tornou-se mais claro como já dito, apesar de neste estudo de caso não ter sido constatado nenhuma restrição quanto aos requisitos tanto de *software* quanto de qualidade a serem implementados, nem do número de *stakeholders*, visto que neste estudo de caso haviam apenas três, o seu auxílio fica um tanto quanto oculto, porém no caso onde houvessem mais *stakeholders* e não pudesse considerar os requisitos/visões provenientes de todos, estas matrizes de priorização seriam consideradas de grande auxílio para estipular quais deveriam ser considerados e quais não. Apesar da matriz de priorização de *stakeholders* não contribuir diretamente para nenhuma das questões levantadas pelo estudo GQM, ela ainda é considerada importante por fazer parte da construção da matriz conjunta.

Além destes pontos também percebeu-se que o modelo SR gerado para os requisitos de qualidade auxiliam no entendimento dos requisitos e das métricas de qualidade relacionadas a eles e isso contribuiu para obtenção da saída métricas de qualidade do planejamento.

Outro ponto importante em utilizar as matrizes de priorização é a credibilidade das saídas métricas de qualidade e *checklist* de qualidade, uma vez que pode-se ter confiança de que os requisitos dos quais derivaram essas saídas serão utilizados. Ou seja, esta identificação que é feita

logo nas primeiras fases de um projeto contribui para que este seja mais conciso e também na diminuição do retrabalho.

Tabela 6: Resultado das questões conforme as métricas.

Métricas	I*						Matriz de priorização		
	SD dos requisitos de software	SR dos requisitos de software	SD dos requisitos de qualidade	SR dos requisitos de qualidade	SD do projeto	SR do projeto	Requisitos de software	Requisitos de qualidade	Stakeholders
M01 - Permite maior compreensão do contexto do sistema	A	A	A	A	N	N	N	N	N
M02 - Representa os requisitos do produto	A	A	A	A	N	N	A	A	N
M03 - Identificação das métricas de qualidade	N	N	A	A	N	N	N	A	N
M04 - Permite a obtenção de dados para elaboração de <i>checklists</i> de qualidade	N	N	A	A	N	N	N	A	N
M05 - Tornam o processo mais claro	A	A	A	A	A	A	N	N	N
M06 - Auxilia na realização do documento de conformidade	N	N	N	N	A	A	N	N	N
M07 - Auxilia na elaboração e avaliação das matrizes de priorização	A	A	A	A	N	N	N	N	N
M08 - Permite expressar as visões dos <i>stakeholders</i>	A	A	A	A	N	N	A	A	N
M09 - Representação visual para verificação	P	A	P	A	P	A	N	N	N
M10 - Permite maior compreensão do contexto do projeto	N	N	N	N	A	A	N	N	N

Legenda: **A:** aplica-se **N:** não aplica-se **P:** parcialmente aplicável.

A fim de responder as perguntas elaboradas com a abordagem GQM, os artefatos gerados pelos passos dessa proposta foram analisados. Notou-se que a utilização dos modelos SD e SR dos requisitos de *software* e de qualidade proporcionaram uma visão macro do sistema a qual foi de grande utilidade para que os clientes identificados por meio da matriz de priorização dos *stakeholders* pudessem entender o escopo geral do sistema computacional encomendado.

Para a métrica M02 pode-se notar que os artefatos que envolviam os requisitos do produto sendo eles, os modelos SD e SR dos requisitos de *software* e de qualidade e as matrizes de priorização de requisitos de *software* e de qualidade, tem a capacidade de retratar todos os requisitos do produto encomendado.

Quanto há métrica M03 observou-se que apenas os artefatos que estavam diretamente ligados com os requisitos de qualidade tinham a capacidade de auxiliar na identificação dos requisitos de qualidade, ou seja, apenas os diagramas SD e SR dos requisitos da qualidade e a matriz de priorização dos requisitos de qualidade. Sendo que o mesmo ocorre para a métrica M04.

Com relação a métrica M05 é possível afirmar que todos os diagramas provenientes do I\* que foram criados ao longo desta proposta mostraram-se de grande utilidade para facilitar a compreensão do sistema computacional encomendado bem como do projeto.

Já para a métrica M06 notou-se que apenas os elementos dos diagramas SD e SR do projeto exerciam influência na elaboração do documento de conformidade.

Quanto a métrica M07 observou-se que os diagramas SD e SR dos requisitos de *software* e de qualidade possuíam elementos que são muito úteis tanto para a identificação dos elementos a serem priorizados quanto para dar uma visão macro dos requisitos do sistema encomendado e assim apoiar os *stakeholders* no momento em que estes têm de avaliar os elementos que foram mapeados para as matrizes de priorização.

Com relação a métrica M08 pode-se notar que os diagramas SD e SR dos requisitos de *software* e de qualidade, assim como as matrizes de priorização de requisitos de *software* e de qualidade, possibilitam expressar as intencionalidades e visões dos *stakeholders*.

Para a métrica M09 observou-se que o poder representativo para o acompanhamento das atividades é parcial quando se trata dos diagramas SD dos requisitos de *software* e de qualidade, devido a estes não contemplarem as sub-tarefas necessárias para satisfazer uma dependência, diferente do que acontece com os modelos SR dos requisitos de *software* e de qualidade, sendo por isso que estes foram considerados como aplicáveis.

Com relação a métrica M10 que é mais voltada para o contexto do projeto, apenas os diagramas SD e SR do projeto expressaram elementos que permitiam maior contexto do projeto, isso deve-se a eles poderem ver em um contexto geral as tarefas exigidas dos membros da equipe.

## 5.6 Considerações finais

Neste capítulo a proposta foi aplicada a um estudo de caso de uma aplicação *web* desenvolvido, no âmbito acadêmico, o qual contava inclusive com artefatos como o documento de requisitos de *software* e o registro das partes interessadas, o que reduziu os esforços necessários para a obtenção das informações referentes ao projeto.

Algumas informações referentes aos atores e a as atividades necessárias para concluir o sistema foram obtidas pelo documento do estudo (PESSINI; SANTANDER, 2013), sendo necessária a consulta a toda a equipe do projeto para obter as atividades necessárias ao projeto e também necessária a consulta aos *stakeholders* a fim de elicitar os requisitos de qualidade.

Os objetivos foram mapeados para dois modelos SD sendo um para os requisitos do *software* e outro para os requisitos da qualidade, sendo que esta separação foi feita no intuito de facilitar o entendimento e também por considerar que os requisitos de qualidade têm pesos diferentes dos requisitos do *software*.

Além destes, foi criado um modelo SD para o mapear os envolvidos no projeto para que posteriormente as intencionalidades pudessem ser exploradas e utilizadas para a elaboração do documento de conformidade.

O processo de elaboração das matrizes de priorização foi realizado com o auxílio dos diagramas SD onde os *dependums* foram mapeados para os eixos da matriz L-shaped, para depois serem avaliados comparando-os da maneira um-para-um.

Para a obtenção das três saídas que esta proposta auxilia, foram utilizados os artefatos dos modelos SR do sistema e do projeto, mais especificamente, foram utilizados os atores destes modelos bem como os *dependums* encontrados nestes modelos e a segregação dos *dependums*, sendo que estes foram importantes para a avaliação das matrizes de priorização o que levou a produzir resultados mais confiáveis devido a confiança dos *stakeholders* em avaliar os requisitos.

Além das as matrizes de priorização dos requisitos de *software* e qualidade servirem de linha base para o desenvolvimento e também por serem úteis para estabelecer quais requisitos

permanecem no desenvolvimento, isso é muito importante pois afeta diretamente no conteúdo dos documentos métricas de qualidade e *checklist* de qualidade – os quais são saídas do planejamento da qualidade.

# Capítulo 6

## Considerações finais

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

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

### 6.1 Vantagens do uso da proposta

O processo do planejamento da qualidade considerando características organizacionais apresenta vantagens sobre a abordagem tradicional, tanto por permitir identificar tais características, quanto por oferecer um maior número de possibilidades em termos de documentação, para que a equipe do projeto possa gerar o documento de conformidade, métricas de qualidade e *checklist* de qualidade, os quais são saídas da fase de planejamento de qualidade segundo o PMBOK.

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

Os diagramas juntamente com as matrizes geradas permitem representar os requisitos do *software* e da qualidade, em termos de dependências, cuja satisfação permite conseqüentemente satisfazer os requisitos, ou seja, validá-los.

Os requisitos podem ser satisfeitos e a sua validação pode ser feita através da realização das atividades referentes aos mesmos. Este ponto é interessante para a elaboração do documento de conformidade que utiliza o diagrama voltado para o projeto e o diagrama dos requisitos da qualidade além da matriz L-shaped para os mesmos. Estes artefatos gerados também são de grande valia para o entendimento e efetiva elaboração das métricas de qualidade e do *checklist* de qualidade.

Os diagramas SR dos requisitos de *software* e de qualidade, mostraram-se eficientes no apoio aos clientes, sendo que com o auxílio destes percebeu-se que os clientes ficavam mais confortáveis ao avaliar os elementos e assim produzindo resultados mais confiáveis.

## 6.2 Contribuições futuras

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

### 6.2.1 Experimento

A fim de verificar a eficiência da proposta em projetos maiores, pode ser realizado um experimento que permita mensurar a efetividade desta em um contexto maior sendo ele no âmbito comercial, submetendo esta proposta a um maior número de dados como mais requisitos e *stakeholders*.

### 6.2.2 Ferramenta

Uma contribuição para a proposta seria o desenvolvimento de um *software* que permitisse a construção das matrizes de priorização a partir dos diagramas, simplificando o processo.

Uma ferramenta possibilitaria ainda a integração com outras áreas, uma vez que as matrizes de priorização não são uma preocupação apenas no planejamento da qualidade.

## 6.3 Considerações finais

Com a proposta de planejamento da qualidade via modelagem organizacional espera-se que o processo para a obtenção das saídas – documento de conformidade, métricas de qualidade e *checklist* de qualidade – seja melhor atendido e que as preocupações levantadas ao longo deste trabalho sejam sanadas, além de simplificar a tarefa de elaboração das matrizes de priorização, a

tarefa de avaliação dos elementos e a construção dos três artefatos da saída. Realizando estes de modo tão ou mais confiável que o método tradicional proposto pelo (PMBOK, 2008).

A automatização da abordagem proposta reduziria o tempo de aplicação, além de torná-la mais flexível, sendo que alterações nos diagramas resultariam em alterações diretas nas matrizes de priorização.

# Referências Bibliográficas

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.

ANDRADE, S. C. DE; TAIT, T. F. C. Uma aplicação do guia PMBOK na gestão de projetos de software. , p. 2–11, 2012.

BASIL, V., CALDEIRA, G., ROMBACH, H. Encyclopedia of Software Engineering, capítulo: Experience Factory, p. 469-479, John Wiley & Sons, v. 1, 1994.

BAUER, F. L. Software Engineering: Report on a Conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968, P. Naur and B. Randell, eds., Scientific Affairs Division, NATO, 1969.

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

CHICHINELLI, M. Contribuição da técnica de modelagem organizacional i\* ao processo de engenharia de requisitos, com destaque aos requisitos não funcionais – Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos – SP, 2002.

CORDEIRO, A. G., FREITAS, A. L. P. Priorização de requisitos e avaliação da qualidade de software segundo a percepção dos usuários, p. 160–179, 2011.

COUGHLAN, J.; MACREDIE, R.D. Effective communication in requirements elicitation: a comparison of methodologies. *Requirements Engineering*,v. 7, p. 47–60, 2002.

ENAMI, Lucia; TAIT, Tania F. C.; HUZITA, Elisa H M. A project management model to a distributed software engineering environment. Em: ICEIS 2006 - INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 2006, Papus. Anais do ICEIS 2006.

GOMES, A., OLIVEIRA, K. M., ROCHA, A. R. Métricas para Medição e Melhoria de Processo de Software. COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação, Rio de Janeiro, 2001.

GRANDO, F. L. Avaliando técnicas de modelagem organizacional no contexto de desenvolvimento de sistemas computacionais: uma abordagem empírica (Monografia) – Colegiado de Ciências da Computação – Universidade Estadual do Oeste do Paraná, Cascavel - PR, Novembro 2010.

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: 18/07/2014.

HENN, F. A. Integrando o Framework i\* ao Processo de Gerência de Tempo (Monografia) - Colegiado de Ciências da Computação – Universidade Estadual do Oeste do Paraná, Cascavel - PR, Outubro 2012.

HWANG, S.-M. Quality Metrics for Software Process Certification Based on K-model. **2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops**, p. 827–830, 2010. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5480777>>. Último acesso em: 20/07/2014.

INTERNACIONAL TRADE CENTRE UNCTAD/WTO – INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL – IMETRO. Gestão da dualidade na exportação – um livro de respostas para exportadores de pequeno e médio porte Brasil. Rio de Janeiro. ITC/IMETRO, 2003

INSTITUTE, P. M. O que é Gerenciamento de Projetos? **PMI**, 2013. Disponível em: <<http://www.pmi.org>>. Acesso em: 09/07/2014.

INSTITUTE, P. M. Project management institute **PMI**, 2013. Disponível em: <<http://brasil.pmi.org/brazil/AboutUS/WhatIsProjectManagement.aspx>>. Último acesso em: 09/07/2014.

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

Juran, J.M. and Godfrey, A.B., Eds., Juran’s Quality Handbook, 5th ed., McGrawHill, New York, 1999, pp. 2.1–2.2.

KARLSSON, J.; RYAN, K. Supporting the selection of Software Requirements. In: INTERNATIONAL WORKSHOP ON SOFTWARE SPECIFICATION AND DESIGN (IWSSD’96), 8<sup>th</sup>. Proceedings... 1996, p. 146-149.

LEITE, J. Gerenciando a qualidade de software com base em requisitos. Qualidade de software: teoria e prática. São Paulo. p. 1–8, 2001. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Gerenciando+a+Qualidade+d e+Software+com+Base+em+Requisitos+□#0>>. Último acesso em: 22/07/2014.

MAGALHÃES, A.L.C. A Garantia da qualidade e o SQA: sujeito que ajuda e sujeito que atrapalha. *ProQualiti – Qualidade na produção de software*, v. 2, n.2, p. 9-14, 2006.

PESSINI, T; SANTANDER, V. F. A. Avaliando metodologias de desenvolvimento de aplicações web. 4 MEDITEC – Medianeira in Technology, Medianeira, 2013.

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

PMI. Project management institute. Disponível em: <http://www.pmi.org/>. Último acesso em: 19 de Julho de 2013.

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

ROSE, Kenneth H. Project Quality Management - Why, What and How J. Ross Publishing. 2005

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: 10/07/2014.

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.

SEI. Software engineering institute. Disponível em: <http://www.sei.cmu.edu/>. Último acesso em: 10/07/2014.

SRIDHAR, V.; NATH, D.; MALIK, A. Analysis of User Involvement and Participation on the Quality of IS Planning Projects: An Exploratory Study. *Journal of Organizational and End User*, p. 80–98, 2009. Disponível em: <<http://www.igi-global.com/article/journal-organizational-end-user-computing/4148>>. Último acesso em: 21/7/2014.

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

VARGAS, Ricardo V. Gerenciamento de projetos: Estabelecendo diferenciais competitivos. 5.ed. Rio de Janeiro : Brasport, 2003.

TEIXEIRA, C. A. N.; CUKIERMAN, H. L. Por que falham projetos de implantação de processos de software?. Em: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE/WOSES (Workshop Um Olhar Sociotécnico sobre a Engenharia de Software), 2007, Porto de Galinhas – PE. 3. WOSES. Rio de Janeiro: PESC/COPPE – UFRJ, 2007. p. 1-12.

TRAVASSOS, G. H., GUROV, D., AMARAL, E. A. G. Introdução à Engenharia de Software Experimental. Rio de Janeiro: COPPE/UFRJ, Programa de Engenharia de Sistemas e Computação, 2002. Relatório Técnico RT-ES-590/02.

VASCONCELOS, A. M. L.; DE LA VARA, J. L.; SANCHEZ, J.; PASTOR, O. Towards CMMI-compliant Business Process-Driven Requirements Engineering. **2012 Eighth International Conference on the Quality of Information and Communications Technology**, p. 193–198, 2012. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6511808>>. Acesso em: 20/7/2014.

VASCONCELOS, A. et al., Towards CMMI-compliant MDD Software Processes. In: THE SIXTH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING ADVANCES, 2011. Barcelona: [s.n], 2011. p. 522-527.

YU, E. Modelling Strategic Relationships for Process Reengineering, Thesi (Phd Thesi) – Software Enginnering - University of Toronto, Toronto, Canadá, 1995.

YU. E. P. Giorgini, N. Maiden and J. Mylopoulos. Social Modeling for Requirements Engineering. The MIT Press. Massachusetts Institute of Technology, 2011.

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