AMBIENTE DE REQUISITOS

Maria Amélia B. Serrano
Arndt von Staa
{mameia, arndt}@inf.puc-rio.br
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
22453 – 900 Rio de Janeiro,
Brasil

PUC - RioInfMCC 36/95

Novembro 1995

Resumo

É definido um Ambiente de Requisitos composto por uma linguagem de representação para o registro de requisitos, um processo de geração da especificação de requisitos e ferramentas utilizadas neste processo. A especificação de requisitos define o porquê do desenvolvimento do software, deve espelhar de forma inequívoca o problema, e deve permitir avaliar se a solução futura é aceitável. O processo de especificação de requisitos sistematiza de forma controlada a aquisição da informação relevante à especificação dos sistemas de software, garantindo, dessa forma, a qualidade da especificação. Os problemas de criação e manutenção de especificações de requisitos foram analisados através do uso de técnicas de análise de negócios.

Palavras Chaves: requisitos, especificação de requisitos, processo de especificação, análise de domínio, análise do negócio, desenvolvimento de software, engenharia de software

Abstract

This paper describes a requirements environment, composed of a representation language used to record and validate requirements, a requirements specification process and tools to be used in this process. Requirements specifications state why a system should be developed, and precisely define the problem to be solved. They should allow forecasting whether a possible implementation will be acceptable. The requirements specification process organizes the requirements acquisition, recording, maintenance and evaluation, assuming a high level of quality both of the specification itself and of the specified system. The requirements specification and maintenance process is examined from a business analysis perspective.

Keywords: requirements, requirement specification, specification process, domain analysis, business analysis, software development, software engineering

Sumário

1. DESCRIÇÃO DO PROBLEMA **1**

1.1 SITUANDO O PROBLEMA **1**
1.2 OBJETIVOS **2**
1.3 ABRANGÊNCIA **2**
1.4 O PROBLEMA **3**
1.5 AVALIAÇÃO DO ESTADO DA ARTE **4**
1.6 ESTRUTURA DO TRABALHO **5**

2. AMBIENTE DE REQUISITOS **5**

2.1 RELAÇÃO AMBIENTE DE REQUISITOS - AMBIENTE DE NEGÓCIO *(CENÁRIOS DE USO)* **6**
2.2 NATUREZA DA ESPECIFICAÇÃO **7**
2.2.1 QUANTO À RELEVÂNCIA ESTRATÉGICA DO SOFTWARE 7
2.2.2 QUANTO À SOLUÇÃO TÉCNICA DO PROBLEMA 8
2.3 RELAÇÃO AMBIENTE DE REQUISITOS - AMBIENTE DE ENGENHARIA DE SOFTWARE *(PROCESSO DE DESENVOLVIMENTO)* **8**
2.4 CLASSIFICAÇÃO DE REQUISITOS **10**
2.4.1 QUANTO AO NÍVEL DE ABSTRAÇÃO 10
2.4.2 QUANTO À NECESSIDADE 13

3. LINGUAGEM DE REPRESENTAÇÃO **13**

4. PROCESSO DE GERAÇÃO **18**

4.1 FASE CAPTURA DA INFORMAÇÃO **19**
4.2 FASE REPRESENTAÇÃO **23**
4.3 FASE VALIDAÇÃO **23**
4.4 FASE APRESENTAÇÃO **23**
4.5 FASE ACESSO **26**

5. CONCLUSÕES **27**

ANEXO A : GLOSSÁRIO **30**

ANEXO B QUESTIONÁRIOS DE LEVANTAMENTO **36**

AMBIENTE DE NEGÓCIO **36**
SOFTWARE A DESENVOLVER **37**

ANEXO C FORMULÁRIOS DE ESPECIFICAÇÃO **39**

FORMULÁRIO DE REFERÊNCIAS BIBLIOGRÁFICAS **39**
FORMULÁRIO DE ESPECIFICAÇÃO DE NEGÓCIO **40**
FORMULÁRIO DE ESPECIFICAÇÃO DE PROBLEMA **42**
FORMULÁRIO DE ESPECIFICAÇÃO DE USUÁRIO **44**
FORMULÁRIO DE ESPECIFICAÇÃO DE SISTEMA **45**

1. DESCRIÇÃO DO PROBLEMA

1.1 Situando o Problema

A demanda crescente por software é inquestionável em todos os setores. A cada vez procura-se resolver problemas mais abrangentes, com maior funcionalidade, maior número de componentes, levando a uma maior dependência dos usuários e a riscos técnicos maiores. A busca de uma maior produtividade com maior qualidade no setor de produção de software é mandatória. Isto implica que os problemas precisam ser resolvidos de forma sistemática e disciplinada, para que o produto ou serviço resultante seja eficiente, eficaz, útil, utilizável, econômico, confiável e evolutível.

A construção de um sistema de software exige o perfeito entendimento de sua necessidade: o que fazer e porque de fazê-lo. É preciso identificar com precisão os objetivos do sistema, as condições a serem satisfeitas, e documentar tais informações de forma clara e não ambígua. Este registro está contido na especificação de requisitos.

Um dos determinantes de custo, qualidade e cronograma de produção de software é a qualidade da especificação de requisitos. Esta deve servir como base de comunicação para o processo de desenvolvimento do software [SHEMER 87, CORDES 89, WALZ 93]. Estudos feitos mostram que quase todos os projetos de software com falhas de desempenho, ou cujos custos extrapolaram e muito o orçamento previsto, são projetos com deficiências na especificação de requisitos [GOLDIN 94]. Estatísticas realizadas demonstram que mais de 80% dos problemas encontrados em software depois de entregues são conseqüência direta de erros ou omissões nas especificações de requisitos [BERRY94].

Para gerar especificações de requisitos de boa qualidade é necessário que o *ambiente de negócio* e o *ambiente de engenharia de software* formem um todo integrado. A relação *negócio-engenharia de software* é uma *relação de parceria*, onde a engenharia de software é um fornecedor de "*ambientes automatizados adequados às necessidades de uso*" para o negócio, de tal forma que este possa executar sua missão de forma competitiva. Para que isto ocorra é preciso que a comunicação entre ambos seja clara, precisa, completa e não ambígua, baseada em um mesmo modelo conceitual.

A especificação de requisitos é o resultado de um *processo de especificação* definido e gerenciável, onde a informação do que fazer e porque fazê-lo é extraída de forma sistemática e controlada, para então ser analisada, classificada, modelada e registrada, garantindo a completeza e minimalidade do modelo e a perfeita correspondência entre o que se pretende que o sistema de software venha a fazer e o que o sistema efetivamente fará.

Um ambiente de requisitos é um conjunto formado por pessoas, com visões de mundo, vivências, culturas e modelos mentais diferentes, equipamentos, ferramentas, técnicas, métodos, heurísticas, linguagem de representação, processo de especificação. Deve ser configurado de acordo com a natureza de especificação – relevância estratégica do produto a ser fornecido, do problema a resolver, do perfil de público alvo, e do nível de abstração dos requisitos a serem gerados – e definindo planos, produtos intermediários, níveis de qualidade, formas de gerenciamento.

O ambiente de requisitos precisa ser *evolutível* e *adequado ao problema*. Isto significa :

- usar uma linguagem de representação que favoreça a formalização necessária e que seja, ao mesmo tempo, flexível;
- ter um processo de geração que seja um processo de aprendizado, de negociação e de resolução de conflitos, onde modificações constantes são necessárias para se convergir para o consenso

- usar ferramentas automatizadas que apoiem as tarefas de rotina e que transformem a especificação de requisitos em uma atividade sistematizada e disciplinada, adequada às necessidades.

As ferramentas apresentadas neste artigo têm por objetivo organizar as especificações e prover um mecanismo de exploração do documento completo. Cada elemento desta organização, ou estrutura, referencia diversos fragmentos de texto. A especificação propriamente está contida nestes fragmentos de texto. Neste artigo não estabelecemos uma sintaxe para os fragmentos de texto. Esta poderia ser uma linguagem formal ou não, sem que isto afete o aspecto de organização. Com esta estrutura desejamos contribuir para facilitar a manipulação e o entendimento das especificações de requisitos.

1.2 Objetivos

Os objetivos do presente trabalho são:

- definir uma linguagem de representação para documentar os requisitos do sistema de software,
- especificar um processo de geração para estes documentos,
- identificar ferramentas de apoio ao processo.

A *especificação de requisitos* define o porquê do desenvolvimento do software e o conteúdo essencial do mesmo. Tem por objetivos :

- definir de forma inequívoca o problema, seu contexto e permitir avaliar se a solução futura é aceitável;
- fornecer diretrizes para as fases subsequentes de projeto e implementação;
- servir de base de comunicação durante o processo de desenvolvimento de software [SHEMER87].

O *processo de especificação* de requisitos tem por objetivo sistematizar a obtenção da informação relevante à especificação do sistema de software, de forma controlada para garantir a qualidade da especificação.

Examina-se e procura-se resolver os problemas de criação e manutenção de especificações de requisitos, utilizando técnicas de *análise de negócios*.

1.3 Abrangência

A área de especificação de requisitos é uma área multidisciplinar que apresenta uma gama grande de problemas, com um espectro amplo de abordagens [DAVIS 91]. Do ponto de vista do *conceitual,* a especificação de requisitos aborda a área de análise de domínio e a especificação do comportamento externo do sistema de software.[1] De maneira semelhante define-se *análise de domínio de soluções*. Neste caso procura-se identificar propriedades, requisitos e funcionalidades comuns a todas as soluções pertinentes a uma área de solução de problemas. A *análise de domínio de aplicações* leva a linguagens de representação nas quais se pode especificar ou projetar aplicações específicas e onde estas linguagens estão próximas ao problema a especificar. A *análise de domínio de soluções* leva a bibliotecas de soluções.

No contexto da análise de domínio, a especificação de requisitos tem por objetivo:

[1] *Análise de domínio de aplicações* é o estudo e registro do entendimento de uma área de aplicação, visando determinar as propriedades, requisitos e funcionalidades comuns a todas as aplicações pertinentes a esta área.

- a exploração do domínio da aplicação, visando extrair tudo o que é comum a todas as aplicações deste domínio e
- a exploração detalhada de um problema particular para automação das necessidades do usuário, visando identificar o que é específico deste problema.

No que diz respeito ao comportamento externo do sistema, é feita a especificação para atender às necessidades dos usuários dentro de seu contexto e o relacionamento do desenvolvimento de sistemas e requisitos de software [DAVIS 91].

Do ponto de vista *tecnológico* a especificação de requisitos envolve todas as atividades de:

- aquisição de informações de requisitos,
- organização da especificação em linguagem natural,
- o uso de técnicas de especificação formal para definir precisamente o comportamento do sistema e
- técnicas de acesso controlado a esta especificação [DAVIS 91, GOLDIN 94].

Do ponto de vista do *processo de produção de software* é a atividade que inicia o desenvolvimento e envolve todas as atividades de evolução e manutenção dos requisitos, alterações estas resultantes da aprendizagem do problema e seu contexto e/ou da solução utilizada.

1.4 O Problema

O grande problema da especificação de requisitos se origina na comunicação informal entre indivíduos com visões de mundo, vivências, culturas e modelos mentais diferentes. A comunicação informal entre indivíduos provoca problemas de ambigüidade, contradições, falta de completeza e imprecisões na troca de informações quanto às necessidades dos usuários [REUBENSTEIN 91], à identificação do apoio a ser prestado pelo software e quanto ao domínio do problema.

As ambigüidades podem ser resolvidas aumentando o volume de informações a cerca do domínio do problema. As contradições podem ser eliminadas através de um processo interativo de validação e pelo uso de premissas pré estabelecidas. Resolver a falta de completeza é difícil, principalmente quando as informações omitidas são ortogonais aos conhecimentos adquiridos e registrados. As imprecisões são as mais difíceis de evitar e podem ocorrer em qualquer uma das fases do processo de desenvolvimento, mesmo em fases mais avançadas onde se faz necessário um maior detalhamento do software [REUBENSTEIN 91].

Enquanto os problemas da definição forem principalmente de notação, eles podem ser minimizados ou até mesmo eliminados através do uso de técnicas formais de registro e validação. No entanto, quando os problemas da definição forem do entendimento do problema a resolver, pouco pode ser feito através do uso de técnicas mais elaboradas. Precisamos então aumentar a participação *eficaz* dos interessados, de modo que a chance de ocorrência de tais problemas seja tornada muito pequena.

Requisitos de software estão em constante mutação, em virtude das mudanças de necessidades e de tecnologias usadas, das alterações provocadas pelo aprendizado do problema, seu contexto e sua solução. Esta contínua evolução é considerada inerente, ou seja, impossível de ser eliminada, mesmo se dispuséssemos de técnicas perfeitas de elicitação de requisitos.

São conseqüências de *especificação de requisitos não produzida*, ou *produzida de forma inadequada* são, entre outras:

- um grande *volume* de alterações durante e após o desenvolvimento. As alterações se tornam necessárias pois o sistema ou componente posto em uso simplesmente não atende às necessidades veladamente esperadas.

- *custo* e *tempo* demasiadamente grande, incompatível com a dimensão do efetiva do problema a resolver. Este esforço adicional é conseqüência direta do volume de retrabalho devido ao "método" tentativa e erro empregado.
- *risco técnico* grande. São exemplos de tais riscos: sistema incompleto, sistema com funções irrelevantes, sistema não evolutível, sistema não integrável [STAA.93].

1.5 Avaliação do Estado da Arte

A área de especificação de requisitos é uma área multidisciplinar e vem despertando cada vez mais a atenção dos pesquisadores. Vários são os motivos para que isto ocorra, entre os quais o fato de serem poucas as ferramentas que dão suporte a esta fase do ciclo de desenvolvimento. Do ponto de vista da Engenharia de Software, apesar de ser talvez a etapa mais crítica do processo de desenvolvimento, é uma das fases de que menos conhecimento se tem a respeito [GOLDIN 94, REUBENSTEIN 91].

As pesquisas mais antigas se concentram na análise de requisitos, nas linguagens de representação utilizadas e nas ferramentas de análise que utilizam tais linguagens. Já os trabalhos de pesquisa mais recentes se concentram na elicitação de requisitos [BERRY 94, LEITE 88].

Várias são as técnicas para modelagem e validação de requisitos. Entre as técnicas de validação tem-se a simulação, a execução simbólica, a análise lógica da especificação formal [REUBENSTEIN 91]. Um resumo dos métodos e ferramentas atuais é explicitado no trabalho executado por Goldin [GOLDIN 94]. Transcrevemos alguns desses resumos a seguir. O texto detalhado se encontra nas referências indicadas.

PSL/PSA - *Problem Statement Language / Problem Statement Analyzer* [TEICHROEW 77] - é uma linguagem formal usada para expressar objetos e relacionamentos entre objetos do sistema que esta sendo desenvolvido. A linguagem possui uma biblioteca rica em objetos padrão para sistemas de informação, suas propriedades e tipos de relações usadas neste contexto.

SADT - *Structured Analysis and Design Technique* [ROSS 77a, ROSS 77b] é uma técnica que usa diagramas, para modelar o sistema em desenvolvimento. São modelados processos, entradas e saídas, possibilitando uma decomposição hierárquica de subprocessos.

RSL - *Requirements Statement Language* [ALFORD 77] - linguagem usada pela ferramenta REVS (Requirements Engineering and Validation System) para descrição de processos. A REVS faz parte do programa de pesquisa de ferramentas desenvolvido pela TRW, para especificação de requisitos de sistemas de software de tempo real assíncronos.

RDL - *Requirement Definition Language* [WINCHESTER 82] linguagem usada pelo sistema SARA para expressar objetos e relacionamentos entre objetos, onde cada sentença representa uma tupla de um banco de dados relacional.

PAISLey - *Process-Oriented Applicative and Interpretable Specification Language* [ZAVE 82] - linguagem de especificação que gera especificações operacionais, no sentido de que as especificações são modelos executáveis do sistema proposto interagindo com seu ambiente.

RML - *Requirements Model Language* [BORGIDA 85] - é uma linguagem que tem como premissa a contextualização dos requisitos, ou seja, os requisitos só tem significado em uma determinada área do conhecimento do mundo real. Dessa forma as ferramentas de especificação consultam a base de conhecimento do domínio para extrair informações não fornecidas explicitamente. A RML organiza o "mundo" como uma coletânea de objetos que interagem.

IORL - *Input/Output Requirements Language* [SALTON 83] - é uma linguagem gráfica e tabular que possibilita a especificação de cada software, hardware e sistema de gerencia-

mento do sistema que esta sendo especificado e projetado. Esta linguagem é usada pelo TAGS (Technology for Automated Generation of Systems), método de desenvolvimento de sistemas com foco na especificação em detrimento da especificação.

Os trabalhos encontrados na literatura mais recentemente são da área de *elicitação de requisitos*, mais especificamente em :

- ferramentas e ambientes de aquisição do conhecimento, como o trabalho que vem sendo desenvolvido por Leite e Oliveira [LEITE 95]. Dentro deste contexto também se encontra o *Requirements Apprentice* (RA) [REUBENSTEIN 91] ferramenta de suporte à criação e modificação dos requisitos de software, que utiliza a tecnologia baseada em conhecimento para transformar os conceitos informais de requisitos em uma especificação formal. O RA é parte do projeto "*Programmer's Apprentice*" do MIT, cuja finalidade é pesquisar como os analistas de software analisam, sintetizam, modificam, especificam, verificam e documentam os sistemas de software e como estas tarefas podem ser automatizadas.
- processamento da linguagem natural, pesquisa de técnicas que permitam a transformação de uma especificação em linguagem natural restrita em uma especificação de requisitos formal [GOLDIN 94] e.
- aspecto sociológico da aquisição do conhecimento, desenvolvimento de ferramentas de apoio à elicitação de requisitos utilizando técnicas de entrevistas, questionários, interação, conversação e análise de discurso.

1.6 Estrutura do Trabalho

As seções que se seguem estão organizadas em – 2. *Ambiente de Requisitos*, 3. *Linguagem de Representação*, 4. *Processo de Geração*. Na seção 2 discutem-se as características principais do *ambiente de requisitos*, os principais componentes que influenciam o processo de geração e a modelagem adotada para os requisitos conceituais. Na seção 3 aborda-se a linguagem de formulários utilizada no trabalho, apresentam-se as classes de objetos e as relações entre estas classes, tendo como base o modelo de ambiente utilizado. Na seção 4 discutem-se as fases principais do processo de geração da especificação e é apresentado um caso exemplo. O trabalho é finalizado com algumas considerações a cerca do modelo utilizado.

2. AMBIENTE DE REQUISITOS

Existe uma forte interdependência entre a organização do trabalho, as ferramentas disponíveis, os papéis envolvidos no processo de fabricação, no ciclo evolutivo dos mais diversos setores da indústria. Isto não se mostra diferente na Engenharia de Software [GOULD 91]. O serviço do software deve atender às necessidades de cada usuário e às necessidades do ambiente de negócio onde se encontra inserido. Os requisitos, identificados sob a ótica de cada usuário e os identificados sob a ótica do negócio, precisam ser consistentes, não conflitantes e completos. A geração da especificação de requisitos torna-se então um *processo de negociação*, de *resolução de conflitos* onde modificações freqüentes são necessárias para se atingir o consenso. Junte-se a isto fatores externos imprevisíveis que levam a mudanças. O consenso e adaptação à interação externa só podem ser obtidos através de realimentações e negociações sucessivas.

Na fase de aquisição dos requisitos a informalidade é inevitável e até mesmo desejável, uma vez que é inerente ao processo do pensamento humano. Torna-se portanto pré-requisito essencial do processo de aquisição de requisitos lidar com estes tipos de informalidade [REUBENSTEIN 91]. Paralelamente a isto o produto final do processo precisa ser uma especificação completa, clara, não ambígua e precisa. Dessa forma, faz-se necessário ter como parte integrante do *ambiente de requisitos* uma *linguagem de representação* que seja flexível para favorecer a informalidade inicial, e formal o suficiente para garantir a qualidade do produto final.

O processo de especificação de requisitos é um *processo de aprendizado*, onde o ciclo de especificação, validação e nova especificação, se repete continuamente até que o problema, seu contexto e solução estejam suficientemente entendidos e determinados, e um conjunto de requisitos consensuais precisos, completos e inequívocos tenha sido gerado. Essencialmente este processo *converge* para uma especificação de requisitos aceitável ao invés de gerá-la em poucos passos. Para isso o processo de especificação de requisitos requer um *ambiente de requisitos* evolutível e que favoreça a integração harmônica entre o *ambiente de negócio* e o *ambiente de engenharia de software*. A figura 1 identifica de forma simplificada os elementos que participam deste ambiente e que influenciam o processo de especificação.

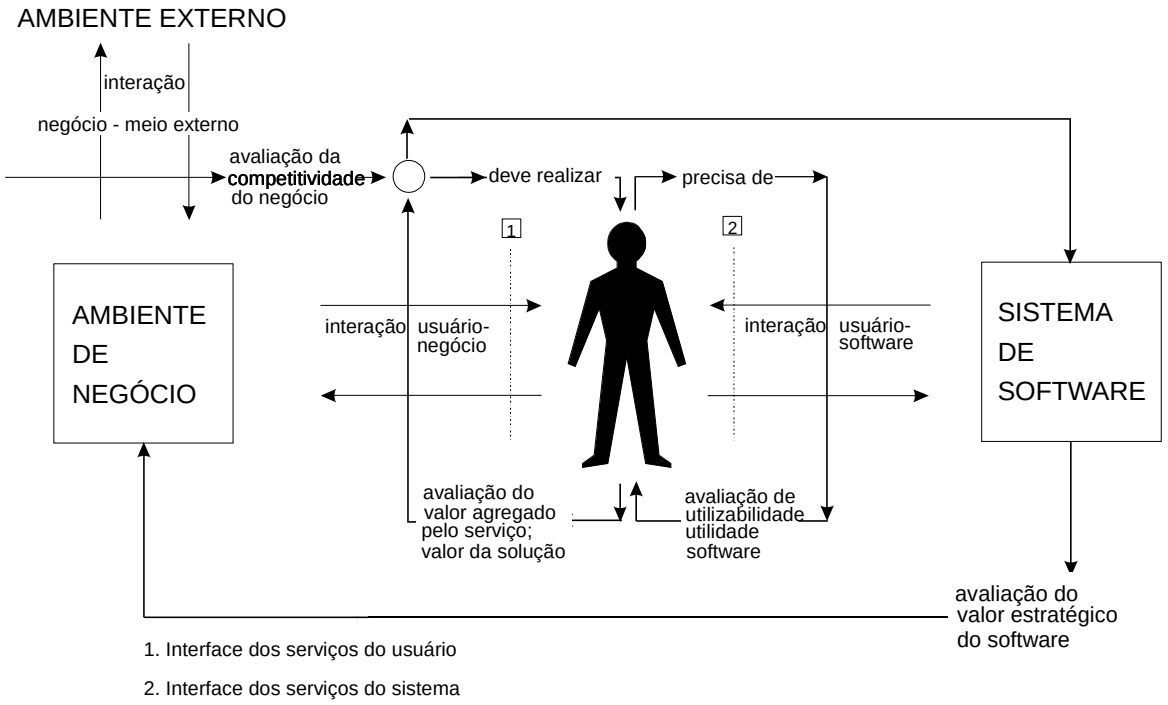


Figura 1 - Componentes do Processo de Especificação

Torna-se imprescindível neste tipo de ambiente [REUBENSTEIN 91], o uso de ferramentas que forneçam mecanismos para facilitar a evolução dinâmica da especificação de requisitos toda vez que o projeto é modificado por imprecisões detectadas ou por alterações das necessidades.

2.1 Relação Ambiente de Requisitos - Ambiente de Negócio *(Cenários de Uso)*

Neste documento entende-se *negócio* como sendo o conjunto de tarefas (atribuições) desempenhadas por uma pessoa, grupo de pessoas, empresas, no intuito de alcançar resultados (objetivos) traçados por estas pessoas, grupos ou empresas. Esta definição permite estruturar negócios. Por exemplo, o negócio da empresa é alcançado através de uma cooperação sinergética dos negócios das divisões desta empresa. Estas, por sua vez, realizam seus negócios através de grupos e estes através de indivíduos. A interação entre componentes de um negócio pode ser diferente da estrutura hierárquica insinuada acima. Foge do escopo deste trabalho discutir e avaliar as diferentes organizações de trabalho em uso hoje em dia. O que é relevante, segundo a ótica do presente trabalho, é o fato de um sistema de software dever apoiar usuários de forma que estes possam servir ao negócio em que atuam, em níveis de qualidade satisfatórios para este negócio.

O negócio de um ambiente de engenharia de software é a criação e manutenção, com elevada produtividade, de software com qualidade assegurada, onde este software apoia eficazmente as tarefas de seus usuários. Vemos, então, que o exame do ambiente de engenharia de software e a análise de sistemas se fundem quando são examinados sob a ótica da análise de negócios.

Com base nesta observação, e visando reduzir o tamanho deste documento, passa-se a examinar o processo de análise de requisitos, como sendo um processo de análise de sistemas, já que neste nível de abordagem não há necessidade de se fazer uma diferenciação explícita.

2.2 Natureza da Especificação

A especificação de requisitos é dinâmica. É uma fotografia instantânea do ambiente do usuário e depende do observador e de seu processo de aquisição de conhecimento. O sistema de software deve atender às necessidades dos papéis desempenhados por cada um de seus usuários (ambiente de negócio) e às necessidades intrínsecas do usuário efetivo. Ou seja, o requisito básico mais abstrato de um sistema de software é o de eficaz e efetivamente apoiar os diferentes papéis desempenhados pelos usuários. Este requisito muda a preocupação com interfaces do sistema para a preocupação com papéis dos usuários, que terão como conseqüência a definição das interfaces

A especificação de requisitos é usada como base de comunicação entre o ambiente de negócio e o ambiente de engenharia de software. Com isto ela torna-se o elemento responsável pelo fornecimento de um ambiente automatizado adequado às necessidades de uso. Para estabelecer os requisitos mínimos necessários a cada classe de problema é preciso a definição de alguns cenários característicos, ou seja, há que considerar a natureza da especificação. Esta pode ser analisada :

2.2.1 Quanto à relevância estratégica do software

Com a consolidação cada vez maior da relação de *parceria negócio-engenharia de software*, a informática assume um novo papel dentro da sociedade moderna, transformando a informação e os sistemas que a manipulam em um recurso estratégico do negócio. Dentro destes contexto, o sistema de software que irá atuar no ambiente de negócio, terá maior ou menor relevância estratégica. A *especificação de requisitos*, conseqüentemente será de natureza diferente dependendo da:

Missão do Negócio

De acordo com a interrelação com a *missão do negócio*, o sistema de software pode ser classificado como : *"missão crítico"* e *"não missão crítico".* A forma de especificação de um e de outro tipo de sistema é bastante diferente uma da outra [ROSENTHAL 90].

O software *"missão crítico"* é essencial para atingir os objetivos estratégicos do negócio, apoiar a tomada de decisão e garantir a posição competitiva. A funcionalidade do software precisa suportar os serviços prestados pelo negócio. Isto implica no desenvolvimento de novos tipos de serviços e na modificação da missão de prestação de serviços (interação usuário-negócio e operacionalização da prestação do serviço). Para a solução automatizada de tais questões não basta investigar a interface usuário-software, onde a informação que aparece é a automação de procedimentos claramente definidos. É preciso , primeiro, refletir sobre a missão do negócio para depois investigar o projeto do sistema em si.

A *especificação de requisitos* deste tipo de sistema implica resolver questões do tipo [ROSENTHAL 90]:

- que funcionalidades e serviços irão facilitar mudanças estratégicas na liberação de serviços ?
- como usar o projeto do sistema de software para atender aos usuários ?

O software *"não missão crítico"* automatiza a estrutura atual do negócio, com seus objetivos atuais, provoca uma melhoria do produto/serviço oferecido. Isto significa que:

- a solução técnica é conhecida, precisa ser automatizada:
- os requisitos do problema são mais ou menos conhecidos
- os requisitos dos usuários precisam ser especificados
- a solução computacional precisa ser especificada

A *especificação de requisitos* deste tipo de sistema de software implica em resolver questões do tipo [ROSENTHAL 90]:

- quais das funcionalidades ou serviços de software facilitarão os produtos liberados pelas atividades atuais ?
- que tipo de automação de informação é compatível com os perfis e responsabilidades existentes no negócio ?

Abrangência

De acordo com o *público alvo* envolvido e *áreas afetadas*, o sistema de software pode ser de "*alto nível de integração*" ou de "*baixo nível de integração*" mudando bastante o compromisso com a qualidade do produto fornecido.

O software de "*alto nível de integração*" integra o trabalho de *múltiplos usuários* distribuídos em *múltiplas áreas* do negócio, de tal forma que qualquer falha técnica na prestação de serviço de uma das áreas envolvidas, afeta rapidamente várias outras, provocando o caos na organização como um todo.

O software de "*baixo nível de integração*" integra o trabalho de *múltiplos usuários* lotados em *uma única área* do negócio, tal que qualquer falha técnica é de efeito localizado e restrito.

Riscos

A classificação da natureza de software quanto a "riscos" é por demais abrangente, implicando em uma área específica de pesquisa, não sendo abordada dentro do escopo do presente trabalho.

2.2.2 Quanto à solução técnica do problema

Os sistemas de software podem ser classificados em categorias, de acordo com suas propriedades. Uma classificação possível é apresentada por Davis [DAVIS 90], que propõe a divisão do universo das aplicações em cinco eixo ortogonais:

- dificuldade de conhecimento do domínio (*difícil* - não conhece o domínio; *não difícil* - solução técnica já é conhecida)
- relação temporal dados-processamento (*estático; dinâmico)*
- dificuldade de especificação (*dados; controle; algoritmo)*
- simultaneidade de tarefas (*seqüencial; paralelo)*
- determinismo da solução (*determinístico; não determinístico)*

2.3 Relação Ambiente de Requisitos - Ambiente de Engenharia de Software *(Processo de Desenvolvimento)*

O novo uso da tecnologia de informação fez do computador, não só um processador de dados, mas um meio de comunicação e coordenação. A modelagem do processo deve contemplar não só o desenvolvimento de software, como a organização da empresa, a tecnologia de coordenação, a engenharia do negócio e a cultura e perfil da equipe envolvida [KRASNER 92].

Há necessidade de um modelo de processo mais abrangente, que contemple vários tipos de produtos/serviços, em vários níveis de abstração. Neste sentido um dos modelos de maior aceitação atu-

almente é o modelo desenvolvido pelo SEI (Software Engineering Institute) para definir e avaliar a maturidade dos processos de desenvolvimento utilizados por empresas que desenvolvem software (CMM - *Capability Maturity Model*) [PAULK 93].

O *processo de desenvolvimento de software* apresentado pelo SEI é definido no contexto empresarial em três níveis de abstração (conforme apresentado na figura 2), cada um deles com uma determinada abrangência e grau de detalhamento de informações.

No nível mais alto define-se o *processo genérico de desenvolvimento de software*, onde a partir de políticas estabelecidas pelo nível conceptual do negócio, o nível organizacional da empresa estabe-

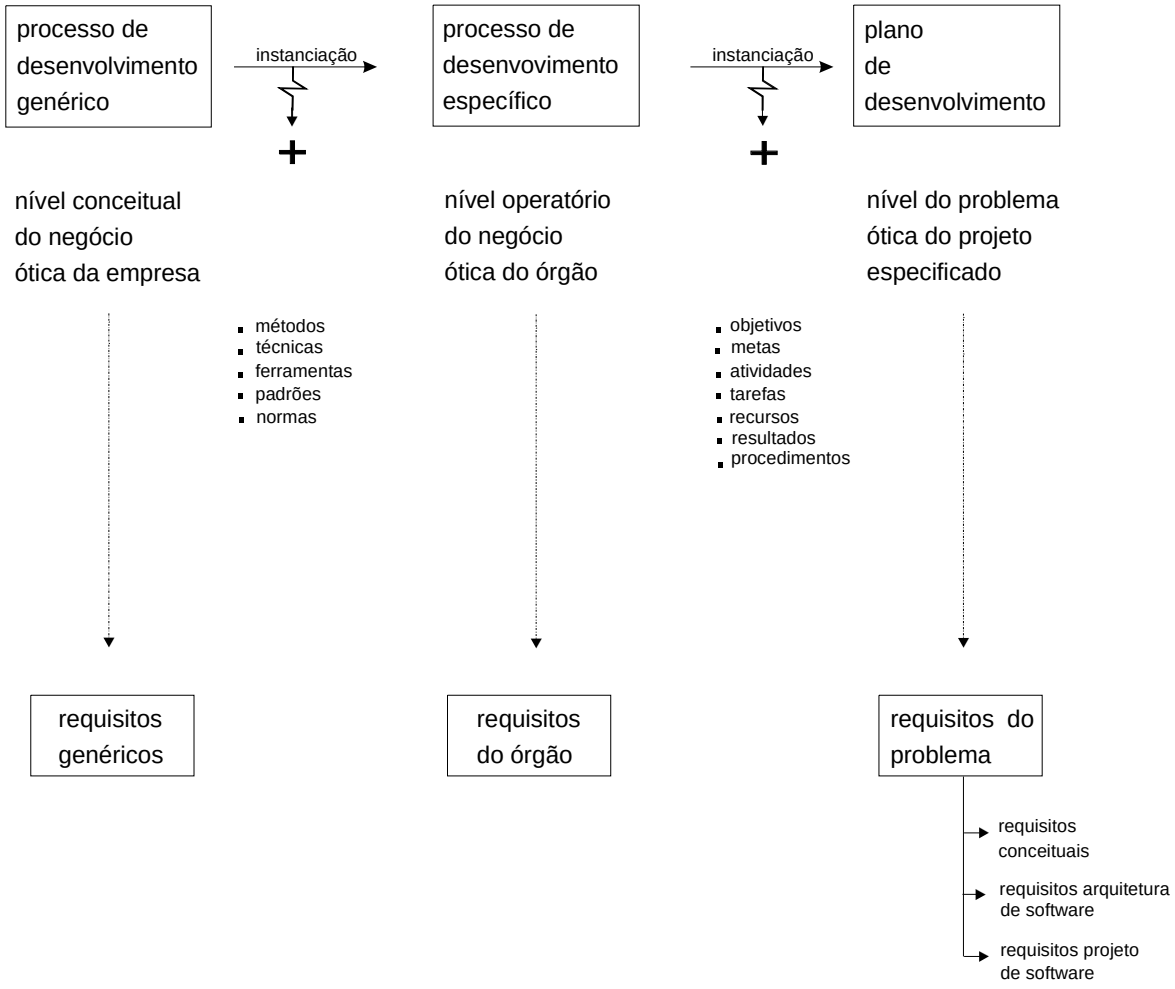


Figura 2 - Requisitos - Processo de Desenvolvimento

lece padrões de desenvolvimento para todos os projetos de software. É um processo capaz de produzir resultados que define as características e áreas de atuação do processo, sob a ótica do negócio, tendo como visão a empresa como um todo. Os componentes do processo neste caso são diretrizes para especificação de requisitos, hipóteses, objetivos, normas para especificação da arquitetura do processo, regras para a customização do processo genérico para uma área específica, natureza das interfaces das atividades.

No segundo nível de abstração é definido um *processo específico de desenvolvimento de software*, pela instanciação do processo genérico para uma área de atuação determinada. O nível operativo da empresa, com foco em projetos específicos, e de acordo com as regras de customização estabelecidas no processo genérico, complementa a definição do processo de desenvolvimento com a especificação de métodos, técnicas, ferramentas, procedimentos característicos dos projetos de sua área de atuação.

O *plano de desenvolvimento de software* é o terceiro nível de definição do processo de desenvolvimento de software, quando então o processo específico é customizado para um problema específico dentro de uma determinada área de atuação da empresa. São definidas as metas a alcançar, atividades, tarefas, recursos disponíveis e resultados esperados.

Para poder planejar é necessário saber o que se quer desenvolver, ou seja, é necessário conhecer os requisitos. De maneira semelhante, para poder controlar qualidade e conduzir o processo de desenvolvimento de modo a garantir qualidade é necessário conhecer os requisitos. A fase inicial do processo de desenvolvimento de software é a especificação de requisitos. É uma fase crucial para todo e qualquer esforço sistemático de desenvolvimento e um fator crítico de sucesso na obtenção de um produto/serviço de qualidade.

Os requisitos dependem do ponto de vista e do nível de abstração. Formam uma estrutura hierárquica organizada em : requisitos genéricos, requisitos dos projetos característicos do órgão, requisitos específicos do problema à resolver.

E dentro da categoria de *requisitos específicos do problema*, que o presente trabalho se propõe investigar o ambiente de engenharia de requisitos, bem como apresentar mecanismos/ técnicas de uso de requisitos, levando em consideração *requisitos conceituais*, referentes à interdependência do problema, o ambiente que o cerca e o sistema de software (modelo essencial do sistema). Os *requisitos de arquitetura*, relativos à escolha da alternativa de solução e os *requisitos de projeto*, relacionados à concretização da solução selecionada, fogem ao escopo do presente trabalho.

2.4 Classificação de requisitos

2.4.1 Quanto ao Nível de Abstração

O processo de desenvolvimento de sistemas de software sugere que o modelo usado para comunicação entre o usuário e o computador seja feito em etapas distintas, executadas em vários níveis de abstração. As atividades realizadas abrangem desde a concepção do sistema, até o seu sucateamento, passando por etapas intermediárias de especificação, arquitetura, projeto, implementação e testes.

Os objetivos a alcançar também variam em nível de abstração e abrangência. Como os *requisitos* são condições, propriedades ou características necessárias para se alcançar um ou mais objetivos [STAA92], eles também formam uma estrutura de abstração.

A especificação de requisitos ocorre em todas as etapas do processo de desenvolvimento : ao *iniciar* (objetivos do sistema; modelo essencial do sistema); ao *arquitetar* (objetivos dos programas; objetivos das bases de dados; modelo de arquitetura do sistema; modelos essenciais dos programas); ao *projetar* (objetivos e requisitos dos módulos; objetivos e requisitos das estruturas de dados; objetivos e requisitos das funções). Assim sendo podem ser classificados em: requisitos conceituais, requisitos de arquitetura, requisitos de projeto

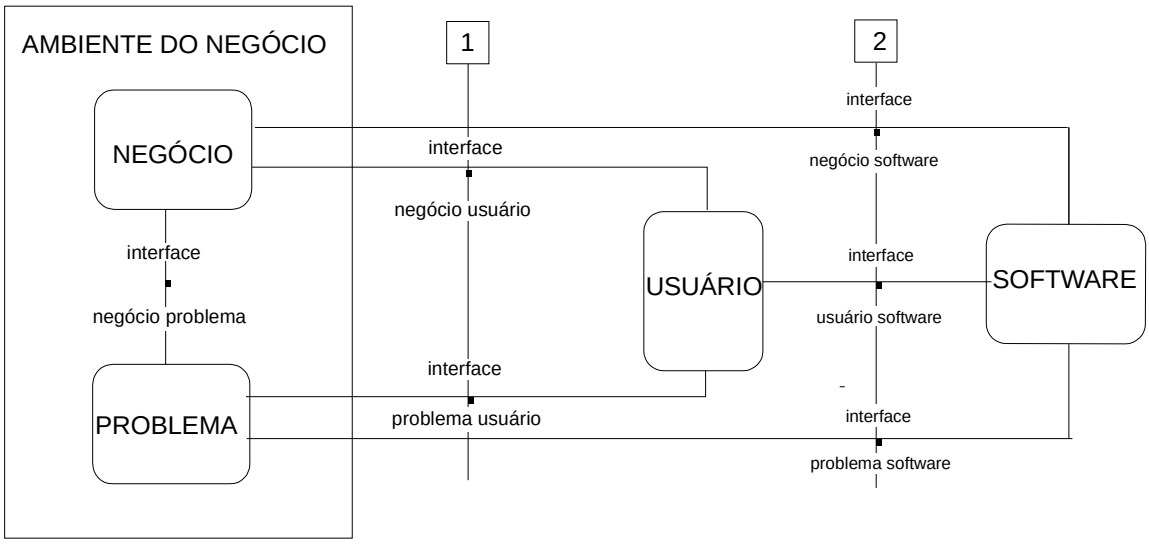
Requisitos Conceituais

Requisitos conceituais são os requisitos adquiridos na fase inicial de criação do serviço de software. Definem com precisão o problema. De acordo com o modelo simplificado apresentado pela figura 1 os requisitos conceituais podem ser categorizados (figura 3) em requisitos conceituais: do *ambiente de negócio*, das conseqüentes *necessidades do usuário*, do *problema a resolver* e do modelo essencial do *sistema de software* a desenvolver.

A seguir apresentamos um conjunto de perguntas a serem respondidas. Embora longa, a relação é parcial.

Negócio

É importante que se tenha uma visão sistêmica do negócio. É preciso ter consciência que existe uma integração e uma interdependência entre todas as partes que compõem o todo empresarial, e



1 - interface dos serviços do usuário

2 - interface dos serviços do sistema

Figura 3 - Categorias de Requisitos Conceituais

que a modificação em uma delas afeta o conjunto e vice-versa. Convém lembrar que a "*empresa*" é entendida aqui como um elemento de uma estrutura, englobando desde a empresa propriamente dita até uma tarefa de um funcionário. Engloba, inclusive empreendimentos visando a definição, ou reengenharia, de outros negócios. É preciso que fique claro qual o impacto que será provocado no ambiente de negócio pela introdução do serviço de software. Para que isto seja possível é necessário conhecer:

- quais são seus objetivos estratégicos.
- quais são suas premissas,
- quais são os seus critérios de sucesso,
- como se relaciona com o ambiente externo.

Ter o entendimento de como o software a desenvolver se insere neste contexto é:

- saber quais dos objetivos estratégicos estão sendo apoiados pelo produto/serviço de software;
- avaliar o valor estratégico, direto ou indireto do sistema de software para o negócio;
- determinar quais das políticas, normas, padrões e procedimentos do negócio, têm influência sob a solução computacional;
- investigar qual a interdependência do software com a organização do negócio, ou seja :
  - quais são os processos empresariais apoiados pelo software,
  - qual é a plataforma tecnológica (computacional e telecomunicações) existente,
  - quais são os órgãos afetados pelo produto/serviço
  - quais são os papéis apoiados pelo software.

Usuário

Conhecer os requisitos dos *usuários* é :

- conhecer o papel – tarefas – que o usuário desempenha no ambiente de negócio;

- fazer o levantamento dos diferentes "cenários de uso", visando a identificação das necessidades e preferências dos usuários
- explicitar as necessidades específicas de cada usuário – estilos de interface, formas de interação homem-máquina – a ser atendido pelo software de apoio às tarefas.

Problema

Os requisitos conceituais relacionados ao *problema a resolver* procuram identificar com clareza o problema, sua solução e a contextualização no ambiente de negócio. É preciso conhecer com precisão:

- quais são os objetivos a perseguir e a alcançar;
- quais são as premissas válidas;
- por que o problema deve ser resolvido (relevância) e quais os benefícios esperados;
- quais são os fatores determinantes de sucesso, critérios de aceitação
- quais os riscos associados;
- qual o embasamento teórico do problema - modelos conceituais, domínios de problema;
- quais as soluções conhecidas.

Para a contextualização do problema no ambiente de negócios é preciso delimitar de forma não ambígua a região do problema no ambiente de negócio, ou seja, é preciso:

- conhecer quais são as relações do problema com o ambiente de negócio e com os usuários envolvidos;
- identificar quais são as políticas, normas, padrões e procedimentos do ambiente de negócio que afetam ou são afetados pelo problema a resolver.

Sistema

Os requisitos conceituais relacionados ao *sistema de software* se referem ao comportamento do sistema. São requisitos que descrevem o modelo essencial que irá atender às necessidades explicitadas pelos diferentes usuários e fornecer as diretrizes para as fases subseqüentes de arquitetura, projeto e implementação. Para isto é necessário conhecer :

- qual a missão do sistema (relacionamento do sistema com o ambiente de negócio), quais os usuários e unidades de negócio envolvidas,
- quais são os objetivos a alcançar,
- quais são os critérios de aceitação,
- qual a sua funcionalidade (processos, estímulos, respostas/serviços) sob os mais diversos pontos de vista (usuários, operadores, suporte, integração);
- quais as interfaces (usuário; software; hardware; comunicação).

O que deve chamar atenção neste conjunto é:

- o volume de perguntas. Embora o conjunto apresentado seja parcial, e como muitas delas ainda estão na forma ambígua, é necessário desdobrá-las para que se tornem precisas. Evidentemente o conjunto completo é grande.

- o volume de referências cruzadas entre perguntas. As perguntas são fortemente interdependentes. Alterando uma reposta implica a alteração de diversas outras. É importante, então, conhecer explicitamente estas referências cruzadas.

O processo de especificação de requisitos pode ser visto, então, como um processo de convergência para uma especificação estável. A especificação estará estável quando não houver mais nenhuma pergunta a ser respondida e nenhuma alteração a ser propagada. É fundamental para isto que existam critérios de avaliação detalhados e específicos para a especificação de requisitos.

Requisitos de Arquitetura

São os requisitos que definem a arquitetura do sistema em diversos níveis de abstração (sistema, programa, módulos, sub-rotinas). No nível mais elevado especificam a plataforma tecnológica (software, hardware e comunicação), interfaces externas, mecanismos de segurança e integridade de dados, volume de dados. Esta categoria de requisitos foge ao escopo deste documento.

Requisitos de Projeto

São requisitos que definem as características de implementação do sistema. O foco da especificação, neste caso, esta no desempenho operacional do sistema em vários níveis de abstração (sistema, programa, módulo, subrotina, algoritmo). Esta categoria de requisitos foge ao escopo deste documento.

2.4.2 Quanto à Necessidade

Os requisitos podem ser classificados em requisitos mandatórios e requisitos desejáveis quanto a necessidade de uso. Os requisitos mandatórios são aqueles que precisam ser satisfeitos de qualquer maneira, sob pena de comprometer o projeto como um todo. Os requisitos desejáveis podem ser relegados, caso isso seja "indicado" por uma análise custo x benefícios. Isto significa dizer que o atendimento do requisito pode :

- implicar em custos excessivos, ou
- comprometer excessivamente o cronograma, ou
- apresentar qualquer outra dificuldade e

em contra partida a falta de atendimento do requisito não coloca em risco a integridade do produto/serviço de software.

3. LINGUAGEM DE REPRESENTAÇÃO

O objetivo primeiro de uma linguagem de representação, no processo de desenvolvimento de software, é estabelecer uma comunicação eficiente e eficaz nas três interfaces: Homem-Homem, Homem-Máquina e Máquina-Máquina, de tal forma que fatos do mundo real sejam modelados, codificados e registrados no computador, para que este execute satisfatoriamente as tarefas solicitadas. Cada uma destas vias, em função do nível de abstração, do tipo de problema a resolver e dos interlocutores, possui requisitos próprios de comunicação. Várias são as linguagens necessárias para registrar e transmitir a informação. Entre elas se encontra a linguagem de representação usada para registrar a especificação de requisitos, cuja importância não pode ser subestimada. Uma representação adequada é essencial para o entendimento do modelo conceptual do domínio, identificação precisa do problema e comunicação inequívoca entre os indivíduos envolvidos [SHEMER 87].

Os elementos que participam no processo de especificação de requisitos foram apresentados de forma simplificada na *figura 1*. Usando como base este modelo, as informações relativas aos requisitos conceituais do sistema, podem ser armazenadas nas *classes de objetos* [2]: *negócio*, *problema*, *usuário* e *sistema de software*. Estas classes são resultado da agregação de uma coletânea de outras classes de objetos. A *tabela 1* apresenta a coletânea de classes agregadas.

As relações entre classes definem associações potenciais entre as classes e são instanciadas por relacionamentos. Usando como base o modelo simplificado da figura 3 e a discussão sobre requisitos conceituais apresentado anteriormente, identificamos as relações entre classes, algumas delas explicitadas na tabela 2, traduzem a semântica das interfaces :

- *usuário - negócio-* descreve como o usuário se relaciona no ambiente de negócio : qual o seu papel, suas atribuições, qual o grupo ao qual pertence, qual o projeto onde esta alocado, quais as tarefas que executa

- *usuário - problema* - associa os problemas que são pertinentes ao papel desempenhado pelo usuário. Esta associação deve ser consistente com os projetos onde o usuário se encontra alocado.

- *problema - negócio* - descreve a contextualização do problema no ambiente de negócio, porque o problema precisa ser resolvido, quais os riscos associados.

- *usuário - sistema de software* - descreve como o usuário interage com o sistema, quais os requisitos da interface homem-máquina, como o sistema deve apoiar suas atividades.

- *sistema de software - problema* - define a abrangência da solução automatizada, quais os objetivos e premissas do problema a serem apoiados pelo sistema.

- *sistema de software - negócio* - descreve o comportamento do software no ambiente de negócios onde vai ser implantado. Os vínculos semânticos do relacionamento são : valor agregado e riscos potenciais

[2] São componentes da linguagem que representam conjuntos de elementos agrupados segundo características comuns e carregam a informação a ser transformada, produzida e/ou armazenada pelo sistema.

TABELA 1- Classes de Objetos - Requisitos Conceituais

Negócio	**Usuário**	**Problema**	**Sistema**
missão do negócio	missão usuário (papel)	especificação do problema	missão do sistema
objetivo estratégico	atividade	objetivo do problema	objetivo do sistema
política do negócio	solicitação (evento externo)	solução técnica	solução computacional
processos	respostas do usuário	premissas do problema	premissas do sistema
objetivos operacionais	critérios de aceitação das repostas	procedimento de cálculo - embasamento teórico	repositório/memória
papéis		sistema de equações	processos funcionais
pessoal		dados de entrada	interfaces
plataforma tecnológica		medidas experimentais	algoritmo
estrutura organizacional		resultados de cálculo	estímulos
procedimentos		critérios de aceitação de resultados	respostas/serviços
normas			mensagens
projeto			critérios de aceitação
atividade			
recursos de projeto			
resultados de projeto			
critérios de sucesso			

TABELA 2 –Relações entre Requisitos Conceituais

Usuário/Negócio	usuário é instância de pessoal e está alocado em uma unidade de negócio
	usuário desempenha um papel que é definido por processo
	usuário é recurso pessoal alocado por projeto
	resposta de usuário é resultado de projeto
Usuário/Problema	usuário resolve um problema relacionado com o papel que desempenha
	lista de solicitação do usuário é limitada pela especificação do problema
Problema/Negócio	negócio estabelece problemas a resolver.
	projeto planeja solução de problemas
	objetivo do problema é subconjunto de objetivos operacionais
	solução técnica do problema é um resultado de projeto
Usuário/Sistema	atividade executada por usuário gera solicitação ao sistema
	lista de solicitações especifica missão do sistema
	resposta usuário é resultado de uma avaliação ou processamento das respostas do sistema
	lista de solicitação de usuário define estímulos e respostas/serviços do sistema
	interface do usuário é especificada por estímulos e respostas/serviços do sistema
Sistema/Negócio	sistema apoia a resolução de problemas que são estabelecidos por negócio
	objetivo sistema é subconjunto de objetivo operacional que é objetivo do processo empresarial
	plataforma tecnológica do negócio é restrição para interface do sistema
	sistema automatiza procedimento do negócio
	sistema integra unidades de negócio
Sistema/Problema	especificação do problema define abrangência do sistema (missão do sistema)
	objetivo do sistema é subconjunto de objetivo do problema
	premissa do sistema é subconjunto de premissa de problema
	respostas/serviços do sistema é subconjunto da solução do problema
Negócio/Negócio	negócio decompõe em negócio
	missão do negócio decompõe em objetivo estratégico
	objetivo estratégico decompõe em objetivo operacional

	objetivo operacional estabelece problemas a resolver
	objetivo estratégico gera política, procedimento e normas de funcionamento
	política do negócio organiza processos.
	política do negócio especifica plataforma tecnológica
	política do negócio gerência pessoal
	processo possui objetivo operacional
	processo define papéis
	processo é gerenciado por projeto
	projeto aloca recursos de projeto
	projeto controla resultado de projeto
	projeto é composto por atividades
Problema/Problema	problema decompõe problema
	especificação do problema decompõe em objetivo do problema
	solução técnica atende objetivo do problema
	solução técnica é um procedimento de cálculo
	procedimento de cálculo recebe dados de entrada
	procedimento de cálculo resolve sistemas de equações
	procedimento de cálculo produz resultados
	dados de entrada é uma medida experimental que depende de um instrumento de medida
Usuário/Usuário	papel desempenhado pelo usuário define missão do usuário
	missão do usuário define solicitação do usuário
	solicitação do usuário produz uma resposta de usuário
Sistema/Sistema	sistema decompõe sistema
	missão do sistema decompõe objetivo sistema
	solução computacional atende a objetivos do sistema
	solução computacional é composta de repositório/memória, processos funcionais, estímulos, respostas/serviços
	estímulos e respostas/serviços especificam interface
	processo funcional recebe estímulos
	processo funcional executa algoritmo

	processo funcional produza respostas/serviços
	interface é uma interface de software, interface de hardware, interface de comunicação, interface de usuário

A linguagem de representação pode ter vários graus de formalidade. São exemplos de soluções propostas na literatura :

- especificação formal, como lógica, equações diferenciais, etc.;
- especificação semi formal, utilizando português restrito;
- especificação informal, usando português de forma irrestrita;
- uso de referências cruzadas e glossários.
- uso de linguagens de representação especializadas, em particular linguagens de domínio.

Uma linguagem de representação abrangente pode ser estabelecida por um conjunto de formulários fortemente interrelacionados. Esta é a forma utilizada por TALISMAN [STAA 92]. Para cada classe de objeto é definido um formulário. Este formulário é especificado através de um programa de formulários escrito na linguagem TALISMAN. Cada formulário admite diversos fragmentos de texto. Estes podem ser designados a receber informação de determinado significado semântico e grau de formalidade.

Cada formulário admite ainda um grande número de relacionamentos com outros objetos. Através de programas de formulários pode-se criar formulários especializados visando a exibição e edição contextualizada do conjunto de informações contidas no repositório.

4. PROCESSO DE GERAÇÃO

O processo de especificação de requisitos pode ser desmembrado em quatro atividades – *captura* da informação, *representação* dos requisitos, *validação* dos requisitos e *acesso* aos requisitos, organizadas em três fases – *elicitação de requisitos*, *análise de requisitos* e *gerenciamento dos requisitos* [REUBENSTEIN 91, GOLDIN 94].

A *elicitação de requisitos* tem por objetivo a obtenção de informações necessárias para representar de forma precisa e inequívoca o modelo conceitual do ambiente do usuário. Aqui são capturados e registrados os requisitos.

A *análise de requisitos* trata da representação e validação conceitual e sintática dos requisitos. É a fase onde há o confronto dos requisitos para assegurar consistência, completeza, não ambigüidade, e adaptação ao problema a ser resolvido. O produto desta fase é um conjunto de requisitos consistentes, precisos usados para a produção do software. No caso do TALISMAN o resultado é um repositório que pode ser explorado como um hipertexto, utilizando formulários capazes de exibir, editar e parcialmente validar o conteúdo do repositório. Estes formulários apresentam partes do conteúdo do repositório em diferentes formas de agregação e formatos, facilitando assim o entendimento do problema a partir de diferentes pontos de vista. Os fragmentos de texto que compõem um formulário podem ter diversos níveis de formalidade, desde irrestrito até, por exemplo VDM. Atualmente, no entanto, TALISMAN não é capaz de verificar a correção destes textos. Para tal necessita-se linearizar o conteúdo do repositório fornecendo o arquivo gerado a ferramentas de validação específicas.

O *gerenciamento de requisitos* tem por objetivo facilitar o acesso aos requisitos, garantir a manutenção de alterações de requisitos gerenciados por técnicas de gerência de configuração e manter o alinhamento entre os diferentes produtos do processo de desenvolvimento e a especificação de requisitos [PAULK 93]. No caso do TALISMAN os requisitos identificados podem ser explicita-

mente relacionados aos componentes das demais especificações, projetos e mesmo implementações. Consegue-se dessa forma a rastreabilidade dos requisitos.

O detalhamento do processo depende do conjunto de requisitos conceituais necessários para especificação do sistema de software, depende do cenário de uso do produto/serviço, ou seja, da natureza da especificação. Um sistema "*missão crítico*" não é especificado da mesma forma que um sistema necessário a automatização de um procedimento de cálculo de uma função específica do negócio; um sistema para um usuário em uma unidade de negócio não é especificado da mesma maneira que um sistema que interliga várias unidades de negócio da estrutura organizacional da empresa.

O conteúdo explicitado neste documento está focado para *especificação de software "não missão crítico"* para *aplicações técnico-científicas de pequeno e médio porte*, com múltiplos usuários restritos a uma unidade de negócio. É suposto que o domínio da aplicação e o domínio da solução sejam conhecidos Dessa forma o escopo dos requisitos conceituais necessários, abrange os *formulários de especificação do problema, do usuário e do sistema* (anexo 2). A especificação do *negócio* não se faz necessária uma vez que o ambiente de negócio é conhecido, se encontra estável e consolidado, e o software a desenvolver não possui relevância estratégica.

Para o caso de especificação de software *"missão crítico",* com múltiplos usuários, distribuídos em múltiplas áreas do negócio, essencial para atingir os objetivos estratégicos do negócio, apoiar a tomada de decisão e garantir a posição competitiva, há necessidade da especificação do domínio da aplicação e domínio da solução. Para tanto o processo de especificação de requisitos conceituais, além dos formulários já citados, precisará do *formulário de especificação do negócio*, também apresentado no anexo C.

4.1 Fase Captura da Informação

Fase inicial onde são coletadas todas as informações relativas ao problema a ser solucionado e seu contexto, e explicitadas todas as necessidades que o software deverá atender. Precisa-se definir com clareza :

- *o que* perguntar ?
- *porque* perguntar ?
- *como* perguntar ?
- *a quem* perguntar ?

As atividades subsequentes tentam responder a estas questões. A *atividade 1* focaliza a primeira questão, coletando todas os documentos existentes sobre o problema e sua solução. As *atividades 2, 3 e 4* tentam sistematizar o procedimento para responder as outras duas questões, utilizando-se de *técnicas de entrevistas*, *questionários* e as informações coletadas na atividade 1. No anexo B são apresentados alguns exemplos de questionários para levantamento.

Atividade 1 : Fontes de Informação

1. Coletar todas as *informações* disponíveis relativas ao problema tais como:

- notas técnicas;
- embasamento teórico da solução técnico-científica;
- atas de reuniões;
- procedimentos operacionais;
- manuais de operação de equipamentos;

2. Registrar todos os documentos consultados em uma *base de referências bibliográficas*, para serem referenciados durante o processo de desenvolvimento. O *formulário de referências bibliográficas* proposto no *TALISMAN* foi utilizado para isto e se encontra reproduzido no anexo C.

Atividade 2 : Caracterização do Problema

1. *Coletar informações* sobre o problema e sua solução técnica.

Responder os itens de 1 à 6 do *formulário de especificação do problema*. Sempre referenciar a fonte de informações do *formulário de referências bibliográficas*.

2. *Avaliar a Solução Técnica.*

Nesta etapa do processo de geração, a avaliação do solução técnica diz respeito a *validação* das informações fornecidas no *formulário de especificação do problema*. Para o cenário de uso escolhido, a solução técnica já é uma solução conhecida e consolidada, não sendo analisada sua escolha (o item 4 não é verificado).

2.1 Verificar se os *critérios de qualidade* fornecidos (item 6) estão sendo atendidos. Caso negativo rever os critérios informados.

2.2 Analisar a *consistência técnica da solução técnica,* ou seja

- rever *descrição do problema* (item 1)
- rever *objetivos do problema* (item 3); verificar se todo objetivo possui *fator crítico de sucesso* e indicador *de desempenho*;
- analisar compatibilidade entre descrição e objetivos do problema
- rever solução técnica (item 5)
    - estabilidade numérica dos métodos matemáticos usados;
    - definição de todas as variáveis;
    - sistema de unidades consistente
    - ordem de grandeza do resultado;
    - faixa de variação de cada variável;
    - precisão numérica (implementação de fórmulas);
    - interpolação em tabelas (uso de funções analíticas);

Atividade 3 : Caracterização do Usuário

1. *Coletar informações* sobre os usuários e suas necessidades.

Responder os itens de 1 à 6 do formulário de especificação de usuário, para cada um dos usuários do sistema de software. Sempre referenciar a fonte de informação no formulário de referências bibliográficas.

2. *Avaliar as Necessidades.*

Nesta etapa do processo de geração, a avaliação das necessidades do usuário diz respeito a *validação* das informações fornecidas no *formulário de especificação do usuário* para cada usuário em separado e avaliação da *compatibilidade dos usuários* entre si e *coerência com o problema* a ser solucionado.

2.1 Para todo usuário do sistema:

- Verificar se para toda resposta de usuário existe pelo menos um *critério de qualidade* a ser satisfeito.. Caso negativo rever as informações fornecidas (itens 5 e 6).
- Analisar a compatibilidade entre os critérios de qualidade e respostas de usuário
- Verificar se para toda resposta de usuário existe pelo menos uma *solicitação* feita. Caso negativo rever as informações fornecidas (itens 4 e 5).
- Verificar se as solicitações informadas (item 4) estão compatíveis com as *atividades* do usuário informadas no item 2. Caso negativo verificar se as atividades do usuário estão compatíveis com missão *do usuário* (item 3). Caso negativo rever coleta de dados

2.2 Para todos os usuários fatorar as informações comuns quanto a:

- *terminologia* do usuário
- *solicitação* do usuário
- *resposta* do usuário
- *critérios de qualidade*.

2.3 Gerar uma lista de informações comuns a todos os usuários.

2.4 Gerar uma lista de informações personalizada para cada usuário.

2.5 Para as informações específicas de cada usuário em separado, verificar se não existem inconsistências quanto às solicitações, respostas e critérios de qualidade. Caso negativo resolver os conflitos e rever informações do formulário de especificação do usuário.

2.6 Revisar lista de informações personalizada para cada usuário.

2.7 Validadas as informações de especificação do usuário, verificar se existe compatibilidade entre a *solução técnica e as respostas esperadas* pelos usuários. Caso negativo resolver conflitos e atualizar informações fornecidas nos formulários de especificação do problema e do usuário.

Atividade 4 : Caracterização do Sistema

1. *Coletar informações* sobre o sistema de software e sua solução computacional.

Completar o formulário de especificação do sistema. Sempre referenciar a fonte de informação registrada no formulário de referências bibliográficas.

2. *Avaliar a Solução Computacional*

Nesta etapa do processo de geração, a avaliação do solução computacional diz respeito a *validação* das informações fornecidas no *formulário de especificação do sistema.* O item 4, soluções alternativas, não é verificado devido as características do cenário escolhido.

O sistema é avaliado quanto as suas características externas :

- sua utilidade, compatibilidade entre a solução e as necessidades do usuário,
- sua usabilidade, preferências dos usuários quanto a estilo de interação homem-máquina e

quanto as suas características internas:

- desempenho da solução
- potencialidade, compatibilidade entre a solução computacional e solução técnica

2.1 Verificar se os *critérios de qualidade* fornecidos (item 6) estão sendo atendidos. Caso negativo é preciso rever os critérios informados.

2.2 Analisar a consistência técnica da solução computacional, ou seja:

- rever *missão do sistema* (item 1)
- rever *objetivos do sistema* (item 3); verificar se todo objetivo possui fator crítico de sucesso e indicador de desempenho;
- analisar compatibilidade entre missão e objetivos do sistema
- rever *solução computacional* (item 5) , como por exemplo:
  - verificar os parâmetros de entrada e saída dos algoritmos existentes (condições de validade; unidades; estabilidade)
  - comparar os resultados calculados com os dados experimentais;
  - para cada fórmula usada garantir: definição de todos os termos; valor default para todas as variáveis; sistema de unidades consistente; faixa de variação de cada variável;

2.3 Avaliar o *desempenho* do sistema de software verificando (por exemplo):

- desempenho do algoritmo para grandes volumes de dados;
- velocidade de processamento;
- implementação de expressões matemáticas;
- desempenho do método de interpolação em tabelas;
- desempenho do método numérico

2.4 Avaliar compatibilidade da solução técnica e solução computacional

- verificar se as *premissas* do sistema é um subconjunto das premissas do problema. Caso negativo verificar a consistência entre ambas.
- verificar se os *objetivos* do sistema é um subconjunto dos objetivos do problema. Caso negativo verificar a consistência entre ambos
- verificar se existe consistência entre a descrição do problema e a missão do sistema. Caso negativo rever informações coletadas nos formulários de especificação

2.5 Avaliar a *utilidade* do sistema:

- verificar se a *lista de solicitação* de usuário está compatível com os *objetivos do sistema*. Caso negativo rever informações fornecidas (item 3 do formulário de especificação do sistema e item 4 do formulário de especificação do usuário)
- verificar se para todo evento da *lista de solicitação* existe um *processo funcional* que o atenda. Caso negativo rever informações fornecidas (item 5.5 do formulário de especificação do sistema e item 4 do formulário de especificação do usuário)
- verificar se para todo *processo funcional* que gera uma saída e/ou uma mensagem existe uma *resposta esperada pelo usuário*. Caso negativo rever funcionalidade do sistema e informações coletadas (item 5.5 do formulário de especificação do sistema e item 5 do formulário de especificação do usuário)

2.6 Avaliar a *utilizabilidade* do sistema:

- verificar se a *interface homem-máquina* está compatível com os *estilos de interação* de preferência dos usuários. Caso negativo verificar informações fornecidas (item 5.6 do formulário de especificação do sistema e item 2 do formulário de especificação do usuário).

4.2 Fase Representação

Fase onde as informações coletadas são reorganizadas de uma forma sistemática e o conhecimento adquirido é aplicado. Pela característica da linguagem de formulários do TALISMAN, a informação que vai sendo coletada já esta sendo modelada e armazenada como uma coletânea de *objetos* e *relações* entre estes objetos. Cada item do formulário é um objeto ou um atributo de objeto. As relações entre objetos são garantidas pela infra-estrutura de hipertexto que apoia o inter-relacionamento dos formulários.

4.3 Fase Validação

Os requisitos especificados são verificados para garantir que as necessidades dos usuários estão sendo efetivamente atendidas. O TALISMAN possui um validador que verifica as informações coletadas através do preenchimento dos formulários e a consistência entre informações através das relações estabelecidas, bastando para isto que sejam fornecidas as regras de validação desejadas para aquele contexto.

4.4 Fase Apresentação

Atividade 5 : Geração da Especificação de Requisitos

O documento de especificação de requisitos deve conter um número mínimo de informações estabelecidos por normas. Um destes padrões usados são as normas do IEEE [IEEE 84] que estabelece o conteúdo de uma especificação de requisitos como sendo:

- Objetivos do software
- Critérios de Aceitação
- Critérios de Avaliação de Qualidade
- Requisitos Técnicos
  - Requisitos Conceituais
  - Requisitos Arquitetura
  - Requisitos Projeto
- Requisitos Não Técnicos (acordos, condições e / ou termos contratuais)
  - produtos a serem liberados
  - datas de liberação dos produtos
  - cronograma
- Hipóteses
- Restrições

A partir dos formulários de especificação do problema, usuário e sistema são extraídos os dados necessários para emitir uma especificação de requisitos com tal estrutura ou outra qualquer, já que os formulários permitem inserir e excluir objetos com muita facilidade.

Atividade 6: Verificação dos Critérios de Aceitação

Os requisitos gerados precisam ser analisados quanto a qualidade apresentada, ou seja, é necessário que. apoiam alguma atividade do usuário. É preciso responder à questões do tipo :

- estão explicitamente definidos ?
- são aceitáveis do ponto de vista do serviço a ser prestado ?

- são viáveis ?
- é possível perceber a satisfação ? medir ? avaliar ?
- são independentes da tecnologia a ser utilizada ?
- são necessários ?
- são suficientes ?
- são inteligíveis pelos diversos leitores ?
- são não ambíguos ?
- são consistentes entre si ?

Dessa forma a qualidade de uma especificação de requisitos deve ser verificada quanto ao conteúdo, forma e estilo. Uma proposta é apresentada por Davis em [DAVIS 90] e transcrita abaixo. Os seis primeiros critérios dizem respeito ao conteúdo e os três restantes abordam a questão de qualidade da forma e estilo da especificação de requisitos. Assim sendo uma especificação de requisitos com qualidade precisa ser :

Correta

Uma especificação de requisitos está correta se e somente se cada um dos requisitos que a compõe representa algo necessário à construção do produto/serviço de software e o conjunto for completo (necessário e suficiente).

Este é um atributo totalmente dependente do produto/serviço de software a desenvolver. É função do conhecimento do problema e seu domínio.

Não Ambígua

Uma especificação de requisitos é não ambígua se e somente se cada um dos requisitos que a compõe possui somente uma única interpretação.

O uso de um glossário minimiza o problema mas não resolve todos os casos de ambigüidade, inerentes ao processo de especificação, principalmente se a linguagem de representação escolhida for a linguagem natural.

Completa

Uma especificação de requisitos é completa se ela possui as quatro qualidades que se seguem:

a. Tudo que é suposto o software fazer se acha incluído na especificação de requisitos.

Este é o atributo mais difícil de ser verificado. A identificação de falta de informação, principalmente quando esta é ortogonal a informação já capturada e registrada, é extremamente difícil. É necessário uma parceria efetiva com os diversos usuários do software, e um entendimento preciso do problema e seu domínio, para que este problema seja minimizado. Uma técnica bastante eficaz neste caso é a prototipação.

b. Estão incluídas na especificação de requisitos a definição das respostas do software para todos os tipos de entrada realizáveis, para todos os tipos de situações realizáveis.

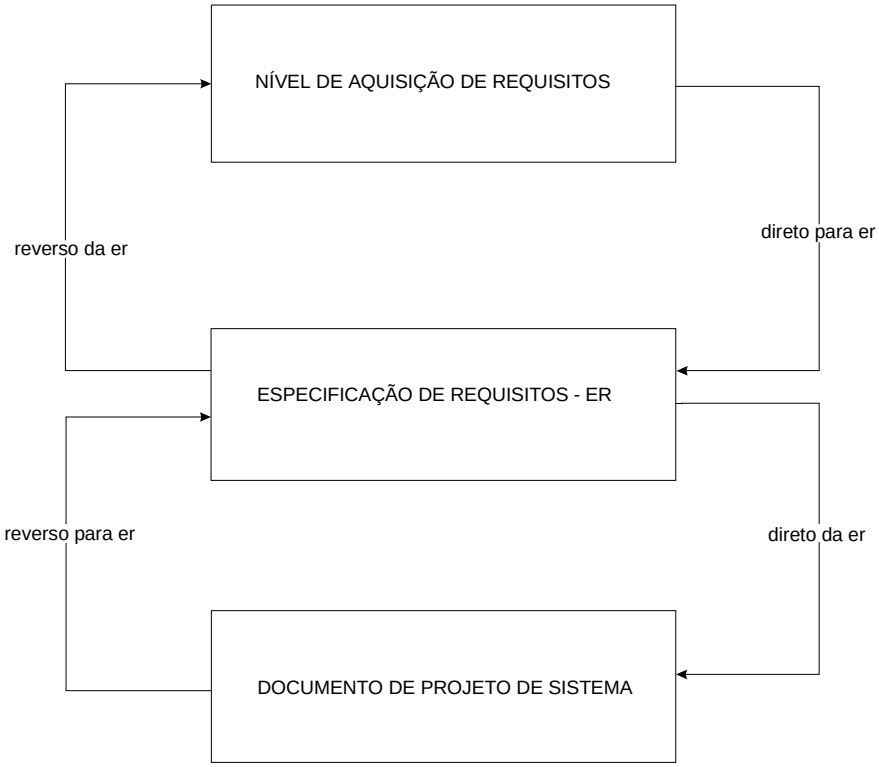


Figura 4 - Rastreamento de Requisitos

Isto significa que para toda a entrada do sistema mencionada, é especificada a saída adequada, que não precisa ser uma função da entrada, mas depender do estado atual em que se encontra o sistema. Podem ser utilizados usuários para resolver o problema do volume de dados.

c. Uma especificação completa, segundo a ótica de processamento do texto, deve ter:

- todas as páginas numeradas;
- todas as figuras e tabelas numeradas;
- todos os termos e unidades de medidas fornecidas e
- todo o material referenciado e seções mencionadas se encontram presentes na especificação.

d. O termo "a ser determinado" não é usada em nenhuma seção da especificação, e quando o for deve trazer associada a indicação de quem é o responsável pela informação faltante, e quando a informação será adicionada à especificação de requisitos.

Verificável

Uma especificação de requisitos é verificável somente se cada um dos requisitos que a compõe for verificável. Um requisito é verificável se e somente se existe um processo de verificação eficaz e de custo efetivo finito.

É importante ter consciência de que a verificabilidade depende da forma como a especificação de requisitos foi escrita, de sua linguagem de representação.

Existe um número grande de razões para que um requisito não possa ser verificável :

- qualquer ambigüidade conduz a não verificabilidade;

- o uso de quantidades não mensuráveis, implica na ausência de um processo de teste finito, o que conduz a não verificabilidade;
- qualquer requisito equivalente ao "problema da parada" é não verificável.

Consistente

Uma especificação de requisitos é consistente se e somente se nenhum subconjunto de requisitos que a compõe for conflitante. Isto pode ser manifestado, por exemplo, através de:

- ***termos múltiplos*** : termos diferentes são utilizados com o mesmo significado;
- ***termos conflitantes*** : um mesmo termo é utilizado com vários significados;
- ***características conflitantes*** : diferentes partes da especificação de requisitos exigem que o software apresente propriedades contraditórias;
- ***inconsistência temporal*** : diferentes partes da especificação de requisitos exigem que o software obedeça caraterísticas temporais contraditórias.

Inteligível

A especificação é inteligível se a linguagem adotada puder ser compreendida pelos diversos leitores e inspetores da especificação.

Quando for utilizado uma linguagem formal deve-se verificar se os leitores têm capacidade de ler e compreender textos redigidos nesta linguagem.

Modificável

Uma especificação de requisitos é modificável se permite que modificações dos requisitos, possam ser feitas fácil, completa e consistentemente, assegurando a satisfação dos demais critérios de aceitação de especificações de requisitos.

Modificabilidade implica controle de versão.

Rastreável

Uma especificação de requisitos é rastreável se a origem de cada um de seus requisitos é clara e se o acesso a cada requisito é facilitado para uso futuro no desenvolvimento e/ou manutenção, evolução do produto/serviço de software. evolução do produto/serviço de software. Existem quatro tipos de rastreamento conforme apresentado na figura 8.

4.5 Fase Acesso

Os requisitos consensuais estabelecidos são registrados, suas modificações são controladas e documentadas, para servir de base ao processo de desenvolvimento e manutenção do software.

Uma vez concluída a especificação de requisitos conceituais deve-se especificar os requisitos de arquitetura e requisitos de projeto para que o desenvolvimento do software possa ser concluído.

A linguagem de formulários utilizada permite que os requisitos sejam acessados com facilidade. Devido a estarem fortemente interligados, o rastreamento entre eles e as fases subsequentes do processo de desenvolvimento é garantido. Da mesma forma é mantida a integridade da base de

dados toda vez que sejam efetuadas atualizações em qualquer uma das fases do processo de desenvolvimento.

5. CONCLUSÕES

A *linguagem de representação* utilizada no presente trabalho fornece a flexibilidade tão necessária a um ambiente de requisitos, cuja caraterística principal é estar em constante mutação, como ao mesmo tempo é formal o suficiente para dar origem a uma especificação de requisitos que atenda aos critérios de qualidade estabelecidos.

Por ser formada de formulários fortemente interligados permite que o processo de geração seja um processo gradual de aprendizado, possibilitando que as atividades sejam executadas de forma não seqüencial, com ciclos de realimentação, sem que com isso haja a perda de sistematização e/ou a falta de consistência informações, em virtude de atualizações em qualquer um dos itens de seus formulários. Com o uso destes formulários as fases de modelagem, armazenamento e validação da informação, do *processo de geração da especificação*, ficam facilitadas, sendo executadas simultaneamente à aquisição das informações.

Um ambiente de requisitos sem o uso de *ferramentas* torna-se caótico, sendo impossível garantir a integridade e consistência das informações. Paralelamente a isto, boa parte da sistematização das atividades do processo de geração é perdida, já que muito esforço é gasto na tentativa de organizar e manter atualizadas as informações coletadas. O uso de ferramentas também vem a oferecer a validação automática dos requisitos e a geração da apresentação da especificação em sua forma final.

O processo e linguagem de representação utilizados permite que o problema, seu contexto e solução, possam ser entendidos e determinados gradualmente, convergindo para uma especificação de requisitos aceitável, ao invés de gerá-la em poucos passos. Permite ainda que a especificação de requisitos seja produzida junto com os usuários (clientes) do software, antes de iniciar o desenvolvimento.

BIBLIOGRAFIA

[ALFORD 77] Alford, M. W.; "A Requirements Engineering Methodology for Realtime Processing Requirements"; *IEEE Transactions on Software Engineering* SE-3(1), pp 60-69, 1977

[BERRY 94] Berry, D.; "The Requirement Iceberg and Various Icepicks Chipping at It"; *VIII Simpósio Brasileiro de Software*, Curitiba, Paraná, outubro 1994

[BORGIDA 85] Borgida, A.; Greenspan, S.; Mylopolous, J.; "Knowledge Representation as the Basis for Requirements Specifications"; *Computer,* 18(4), pp 82-91, April 1985

[CORDES 89] Cordes, D. W.; Carver, D. L.; "Evaluation Method for User Requirements Documents"; *Information and Software Technology* vol 31, nº 4, pp. 181-188, May 1989.

[DAVIS 90] Davis, A. M.; "Software Requirements - Analysis & Specification*"; Prentice Hall, Englewood Cliffs*, New Jersey, 1990

[DAVIS 91] Davis, A. M.; Freeman, P.; "A. Guest Editor- Introduction Requirements Engineering*"; IEEE Transactions on Software Engineering*, vol 17, nº 3, March 1991.

[GOLDIN 94] Goldin, L.; *A Method for Aiding Requirements Analysts in Requirements Elicitation for Large Software Systems*; PhD Thesis, Israel Institute of Technology, July 1994

[GOULD 91] Gould, J. D.; Boies, S. J.; Lewis, C.; "Making Usable, Useful, Productivity, Enhancing Computer Applications"; *Communications of the ACM* pp. 74-85, January 1991

[KRASNER 92] Krasner, H.; Terrel, J.; Linehan, A.; Arnold, P.; Ett, W. H.; "Lessons Learned from a Software Process Modeling System"; *Communications of the ACM*, pp. 91-100, September1992.

[LEITE 88] Leite, J.C.S.P; *Viewpoint Resolution in Requirements Elicitation*; PhD Thesis, University of California, Irvine, 1988

[LEITE 95] Leite, J.C.S.P.; Oliveira, A.P.A.; "A Client Oriented Requirements Baseline"; In *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, 1995

[PAULK 93] Paulk, M. C.; Weber, C. V.; Garcia, S. M.; Chrissis, M.; Bush, M.*; Key Practices of the Capability Maturity Model, Version 1.1*; Carnegie Mellon University, SEI-93-TR-25, February 1993.

[REUBENSTEIN 91] Reubenstein, H. B.; Waters, R. C.; "The Requirements Apprentice: Automated Assistance for Requirements Acquisition"; *IEEE Transactions on Software Engineering* vol 17, nº 3, March 1991

[ROSENTHAL 90] Rosenthal, S. R.; Salzman, H.; "Hard Choices about Software: The Pitfalls of Procurement"; *Sloan Management Review*, pp 81 - 91, Summer 1990.

[ROSS 77a] Ross, D.T.; Schoman, Jr, K.E.; "Structured Analysis for Requirements Definition*"; IEEE Transactions on Software Engineering* SE-3(1), pp 6-15, January 1977

[ROSS 77b] Ross, D.T.; "Structured Analysis (SA): A Language for Communicating Ideas*"; IEEE Transactions on Software Engineering* SE-3(1), pp 16-33, January 1977

[SALTON 83] Salton, G.; McGill, M.J.; "Introduction to a Modern Information Retrieval"; *McGraw-Hill*, New York, 1983

[SHEMER 87] Shemer, I.; "Systems Analysis: A Systemic Analysis of a Conceptual Model"; *Communications of ACM* vol 30, no 6, pp. 506 - 512, June 1987

[STAA 92] Staa Informática; *Ambiente de Engenharia de Software Assistido por Computador – TALISMAN, Versão 4.0*; Staa Informática Ltda, 1992.

[STAA 93] Staa, A. v.; "Projeto TOTEM - Meta-Ambiente de Desenvolvimento de Software"; *Departamento de Informática*, PUC/RJ. 1993

[TEICHROEW 77] Teichroew, D.; Hershey III, E. A.; "PSL/PSA : A Computer-Aided Technique for Structure Documentation and Analysis of Information Processing Systems"; *IEEE Transactions on Software Engieering* SE-3(1), pp 41-48, January 1977.

[WALZ 93] Walz, D. B.; Elam, J. J.; Curtis, B.; "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration"; *Communication of ACM* vol 36, nº 10, pp. 63-77, October 1993

[WINCHESTER 82] Winchester, J.; Estrin, G.; "Requirements Definition and Its Interface to the SARA Design Methodology for Computer-Based Systems*";AFIPS Conference Proceedings* 51, pp 369-379, June 1982

[ZAVE 82] Zave, P.; "An Operational Approach to Requirements Specification for Embedded Systems*"; IEEE Transactions on Software Engineering* SE-8(3), May 1982.

ANEXO A : GLOSSÁRIO

Ambiente de Engenharia de Software

É um conjunto de procedimentos (tarefas manuais), técnicas, métodos, heurísticas, ferramentas (programas), repositórios e modelos de desenvolvimento, cuidadosamente organizados visando o desenvolvimento, a manutenção e a gerência do processo de desenvolvimento e manutenção de sistemas de software[STAA 93]

Ambiente de Negócio

Tudo que afeta o negócio e pode representar risco ou oportunidade. Ele pode ser ambiente-fonte, se fornecer alguma entrada para o negócio, ou ambiente-usuário se receber uma saída do negócio.

Atividade

Estabelece o conjunto de tarefas necessárias para que o serviço seja executado dentro de uma unidade organizacional dinâmica (departamento, setor, seção, etc.) e como pertencente a uma determinada função. Indica também um trabalho administrativo.

Atividade - Fim

São atividades envolvidas diretamente no tratamento de recursos humanos e materiais, transformando-os em produtos e colocando-os ao alcance da comunidade (mercado).

Atividade - Meio

São atividades que proporcionam os meios de apoio ao processo de transformação dos recursos humanos e materiais em produtos e serviços e a sua posterior colocação no mercado.

Atribuição

Estabelece o conjunto de responsabilidades e a autoridade de um determinado cargo.

Autoridade

Permissão concedida de forma legal ou legalizada a determinados níveis da estrutura hierárquica e representada no organograma por diferentes meios, para que tomem decisões sobre assuntos e objetos do trabalho, com vistas a atender ao planejamento nos seus objetivos, metas e projetos, proporcionando o equilíbrio organizacional, a satisfação dos envolvidos e a auto-afirmação. Deve ser reconhecida e aceita pelos membros da empresa, caso contrário, não subsistirá a não ser por meios coercitivos.

Cargo

É utilizado normalmente para indicar a posição hierárquica que uma pessoa ocupa na empresa e o conjunto de atribuições a ela conferidas.

Competência Distintiva

Como a Empresa compete para liberar seus produtos/serviços; escolhas que permitem a diferenciação de seus produtos : custo, qualidade, condições de venda.

Comunicação

É uma atividade constante e ininterrupta entre pessoas, que, utilizando símbolos e/ou códigos num processo formal ou informal, procuram influenciar mutuamente os pensamentos, palavras, ações e omissões.

Crença

São estruturadas através de conceitos emitidos pelos empregados, conforme a sua percepção da realidade ou do meio ambiente que os cerca. O clima organizacional influencia o comportamento individual e a formação destas crenças.

Cultura

Conjunto de maneiras de ser, valores, padrões de comportamento, estilos típicos de ação, etc.

Dado

É a base de toda a informação necessária para a tomada de decisão em determinado momento;

Departamento

Divisão administrativa da empresa; pode ser considerada a maior fração da estrutura empresarial. Em algumas empresas troca a posição hierárquica (no organograma) pelo título Divisão.

Divisão

Terminologia utilizada para indicar a fração maior da empresa que engloba várias áreas ou subáreas de forma abrangente e total.

Empresa

Um sistema composto por variáveis mutuamente dependentes, ou, um sistema composto por conjuntos de elementos unidos por ações integrativas e d inter-relacionamento, para atingirem objetivos, metas e projetos de interesse comum, e para os quais estão dirigidas todas as forças, na busca do resultado sinérgico.

Entrevista

É uma forma de diálogo, onde se procura obter informações de uma ou ambas as partes. É um sistema face a face e que se divide em entrevistas estruturadas e não-estruturadas.

Entrevista Estruturada

É aquela que possui uma listagem ou questionário preestabelecidos, com perguntas padronizadas ou questões que permitem a reformulação no momento em que são feitas, conforme a situação ou contingência.

Entrevista Não-Estruturada

É a entrevista que ocorre com pouco ou nenhum planejamento, o que leva os participantes a divagarem sobre assuntos e à superficialidade das respostas.

Escopo do Negócio

Decisões que determinam onde o negócio vai atuar; muitas vezes o escopo do negócio é visto como: segmentação do mercado, escolhas que definem tipo de produtos (serviços, nichos de usuários, incluindo as forças competitivas estabelecidas por Porter - compradores, fornecedores, substitutos, entrantes)

Estratégia

É um movimento organizacional possível somente com a cooperação de todos os sistemas e subsistemas (inclusive com a participação da alta administração) na elaboração de uma proposta ou plano filosófico e estratégico de trabalho.

Estratégia do Negócio

Determina o balanceamento entre : Escopo do Negócio, Competência Distintiva e Política do Negócio;

Estrutura Organizacional

(1) Estrutura organizacional = Organização

É formada pelos papéis a desempenhar; responsabilidades e estrutura de autoridade dentro do negócio. A estrutura organizacional pode ser organizada, por exemplo, em linhas de produtos ou departamentos funcionais. Determina o número de camadas gerenciais; especifica e mantém o sistema de gerenciamento;

(2) É o resultado do planejamento filosófico e estratégico da empresa; seus limites ou fronteiras são delineados de acordo com os objetivos, metas e projetos propostos nos níveis estratégico, tático e operacional.

Função

Estabelece o conjunto de tarefas desempenhado por pessoas, servindo muitas vezes como base para a departamentalização.

Grupo

São duas ou mais pessoas que compartilham um conjunto de leis, normas, crenças, éticas e valores, estabelecendo relações comportamentais entre as pessoas que influenciam e trazem conseqüências, implícita ou explicitamente, para cada um em particular e para o coletivo.

Hipóteses

São suposições de uma causa ou de uma norma destinadas a explicitar provisoriamente um problema até que os fatos venham contradizê-la ou afirmá-la.

Informação

É o conhecimento adquirido pela interpretação/entendimento de um dado.

Infra Estrutura Organizacional

Determina o balanceamento entre : estrutura administrativa (organização; organograma), processos (organização das atividades) e pessoas (capacitação).

Limite

É o momento, espaço, época, data que marca o começo ou o fim de uma determinada tarefa ou atividade.

Norma

(1) Tudo que é estabelecido como medida para acompanhamento e avaliação de um trabalho que esta sendo realizado ou desenvolvido.

(2) É o conjunto de regras ou procedimentos escritos para avaliação e realização das atividades necessárias ao alcance de objetivos específicos e gerais, propostos pelo negócio e por seus executivos.

Negócio

É o conjunto de tarefas (atribuições) desempenhadas por uma pessoa, grupo de pessoas, empresas, no intuito de alcançar resultados (objetivos) traçados para estas pessoas, grupos ou empresas.

Objetivo

É o fim ou alvo que se deseja atingir depois de um certo tempo e em determinada quantidade ou intensidade. Normalmente é subdividido em partes menores que recebem os nomes de metas e projetos (nesta ordem de grandeza : objetivos, metas, projetos)

Operação

Estabelece a menor parte, ou a parte indivisível , da execução de uma tarefa. Pode ser manual ou instrumental.

Política

Conjunto de objetivos que formam determinado programa gerencial ou empresarial e que condicionam a sua execução. Na empresa, define as diferentes responsabilidades e autoridades conforme os níveis hierárquicos, construindo sistemas de relacionamento e comunicação. Serve de parâmetro, padrão e modelos para a tomada de decisão gerencial. Recebe o nome de planejamento filosófico quando emanados da Diretoria ou Presidência da empresa.

Procedimento

Contém as normas que definem as políticas em vigor na empresa e a descrição detalhada dos processos de trabalho, tanto em nível operacional como administrativo.

Processo

(1) É a visão das etapas de operações limitadas destinadas à obtenção de um ou mais objetivos bem definidos. Atua com elementos concretos, objetivos racionais e práticos, adaptados para cada situação.

(2) É a atividade que possibilita a transformação do insumo (input-entrada) em um produto/serviço (output - saída).

Produtividade

É a medida obtida da relação entre o que é produzido (seja em nível administrativo ou produtivo, isto é, quer serviço quer produto) e a quantidade de recursos consumidos (humanos e materiais empregados).

Produto

É o resultado de qualquer atividade humana ou de um sistema; é o resultado da produção; na empresa é aquilo que é colocado à disposição do consumidor, satisfazendo à sua necessidade. É medido pela produtividade, isto é, a relação entre o que foi colocado como matéria-prima no produto e o resultado da transformação pela participação do homem, envolvendo o que foi aceito pelo consumidor final.

Projeto

É o empreendimento a ser realizado dentro de um tempo finito, utilizando recursos finitos e visando alcançar objetivos definidos.

Questionário

É o sistema de obter respostas às questões por um formulário que o envolvido preenche.

Resultado

É o produto de uma operação, trabalho, pesquisa ou levantamento. Pode ter aspecto de uma deliberação, decisão ou resolução; a conseqüência, efeito e/ou seguimento de um trabalho.

Requisito de Software

(1) Ótica do Usuário
Condição ou capacidade necessária ao usuário para resolver um problema ou alcançar um objetivo.
(2) Ótica do Sistema
Condição ou capacidade que precisa ser alcançada ou possuída pelo sistema para satisfazer um contexto, um padrão, uma especificação, ou qualquer outro documento formal imposto.

Risco

Perigo ou possibilidade de falha e de perda com responsabilidade de ocorrência de um determinado resultado (positivo, neutro ou negativo) conforme a ação ou decisão tomada.

Serviço

Produto da atividade humana e de sistemas de que sem assumir a forma de um bem material satisfaz uma necessidade.

Sinergia

É o ato ou esforço concentrado e numa mesma direção de vários órgãos/departamentos da empresa para a realização de uma tarefa ou consecução de um objetivo. A soma das partes é maior (sinergismo positivo) ou menor (sinergismo negativo) do que obtido isoladamente através de cada elemento do grupo.

Sistema

Conjunto de elementos, materiais ou ideais, chamados de partes interdependentes e integrantes ou subsistemas, entre os quais se possa encontrar ou definir alguma relação ou interesse, formando desta maneira um todo unitário (chamado de sistema integrado). Qualquer sistema tem um objetivo ou uma função recebendo nomes característicos, como sistema de comunicação, sistema financeiro, etc. É considerado sistema aberto quando troca informações, matéria, energia ou semelhantes com o ambiente externo; será sistema fechado, quando não houver esta troca.

Tarefa

É o agrupamento de operações que possuem uma determinada ordem seqüencial que permitem uma interligação, seja entre pessoas ou áreas da empresa. É considerada como aquele trabalho desenvolvido no nível mais detalhado e baixo da produção. Estabelecida a partir da subdivisão do trabalho (Taylor).

Tecnologia

Empregada pelas indústrias, molda os diferentes tipos de grupos de trabalho e a sua constante evolução.

Unidade de Negócio

É qualquer uma das unidades da estrutura organizacional do negócio. Pode ser : um departamento, uma divisão ou até mesmo um grupo provisório formado para executar uma determinada tarefa.

Valores

São objetivos, metas e projetos que a maioria dos membros da empresa procura alcançar, consciente ou inconscientemente. Estes valores traçam o perfil de comportamento que influencia no desenvolvimento das características individuais

ANEXO B QUESTIONÁRIOS DE LEVANTAMENTO

AMBIENTE DE NEGÓCIO

Visão do Negócio

1. Qual a área de atuação do negócio ?
2. Qual o cenário corrente ?
3. Para onde o negócio esta evoluindo ?
4. Qual a direção estratégica?
5. Quais as ações estratégicas ?
6. Qual a implicação do produto/serviço neste contexto ?

Processo Empresarial

1. Qual o processo empresarial que o produto/serviço ira apoiar ?
2. Como vai apoiar ?
3. Qual a posição deste processo na cadeia de valor do negócio ?
4. Quais as entradas e saídas do processo empresarial apoiado pelo software?
5. Como este processo se relaciona no ambiente de negócio ?
6. Quais são os fornecedores e clientes do processo empresarial apoiado pelo software?
7. Qual é o valor agregado pelo processo empresarial ao negócio sem o apoio do software a desenvolver?
8. Qual a modificação neste valor provocada pelo software ?

Plataforma Tecnológica

1. Qual o grau de informatização do ambiente ao qual se destina o produto/serviço de software?
2. Quais as interfaces que o produto/serviço deve respeitar ?

Estrutura Organizacional

1. Qual a estrutura organizacional da empresa ?
2. Quem é o órgão cliente do produto/serviço do software ?
3. Quais são as suas atribuições ?
4. Qual o seu relacionamento dentro da empresa?
5. Quais os objetivos da missão do órgão que serão atendidos pelo produto/serviço de software?
6. Quais as atividades/tarefas do órgão que serão apoiadas pelo produto/serviço de software?

SOFTWARE A DESENVOLVER

Quem

- Quem necessita do produto/serviço ?
- Como os usuários se diferenciam ?
- Qual a visão que cada usuário tem do produto/serviço ?
- Como satisfazer a cada um dos usuários ?

O que

- Quais as necessidades que o software vai atender ?
- O que será feito com o produto/serviço ?
- Quais os resultados esperados ? Como são avaliados ?
- Quais são as solicitações externas que o produto/serviço deverá atender ?
- Quais são as respostas esperadas ?
- Como atender estas expectativas ?
- Poderia ser resolvido de outra forma ?
- Quais as alternativas disponíveis ?
- Quais as que obtiveram sucesso ?
- Por que ?
- Quais as que fracassaram ?
- Por que ?
- Qual o ciclo de vida esperado para o produto/serviço ?
- Quais são os critérios para avaliação do ciclo evolutivo do produto/serviço de software?

Onde

- Onde os usuários necessitam do produto/serviço ?
- Qual o ambiente?
- Como oferecer o produto/serviço no ambiente onde ele é necessário?

Quando

- Quando os usuários precisam do produto/serviço ?
- Como fornecer o produto/serviço no momento necessário ?
- Qual o prazo máximo para entrega do produto/serviço, a partir do qual o produto/serviço deixa de ser estratégico para o negócio?

Por que

- Por que os usuários necessitam do produto/serviço ?
- O produto/serviço é um meio ou um fim ou algo intermediário para o negócio?
- Quais as conseqüências da falha do produto/serviço ?
- Como o produto/serviço se insere no ambiente de negócio?
- Qual a sua relevância para a missão do negócio?

Como

- De que forma os usuários necessitam do produto/serviço ?

- O que fazer para entregar da forma que os usuários necessitam?
- Quais as premissas do produto/serviço ?
- Como os premissas do produto/serviço se relacionam com as premissas do negócio?

Quanto

- Qual a equipe necessária ?
- Ha necessidade de ferramentas de apoio ?
- Quais ?
- As ferramentas precisam ser desenvolvidas ?
- Já existem ?
- Serão adquiridas ?
- Ha necessidade de treinamento ?
- Qual a duração ?
- Qual o tempo previsto para o desenvolvimento ?
- Quais os recursos necessários ?
- Qual é o custo total do desenvolvimento ?

ANEXO C FORMULÁRIOS DE ESPECIFICAÇÃO

FORMULÁRIO DE REFERÊNCIAS BIBLIOGRÁFICAS

1. ***Ficha cadastral*** do documento

Fornecer os dados necessários para a identificação precisa da referência bibliográfica.

Título	
Autores **Editoras** **Volume/Número do periódico** **Data de Publicação** **Páginas do Documento** **Identificação de Referência** **Nome/Número de identificação única do documento** **Código de catalogação/localização do documento**	

2. ***Definição*** do documento

2.1 ***Descrição informal*** do documento

Descrever de forma resumida o conteúdo da referência bibliográfica

2.2 ***Palavras Chave*** do documento

Lista de palavras que classificam a referência bibliográfica quanto ao assunto abordado.

FORMULÁRIO DE ESPECIFICAÇÃO DE NEGÓCIO

1. Informar a ***identificação*** do negócio.

   Registrar o *nome* (e/ou *código*) de identificação inequívoca do negócio.

2. Definir a ***missão do negócio***

   Descrever a missão do negócio significa conhecer :

   - *o que* produz
   - *porquê* produz
   - *para quem* produz
   - *como* vai produzir
   - *quais* são suas *premissas*
   - qual o ambiente *onde* atua

   É preciso especificar também como o sistema de software se relaciona neste contexto, qual seu valor estratégico direto ou indireto.

   Estas informações devem ser fornecidas pelo usuário do ambiente de negócio por intermédio de referências bibliográficas e/ou pelo uso de técnicas de entrevistas e preenchimento de questionários

3. Definir a ***terminologia*** do negócio.

   Registrar e definir no glossário os termos específicos do ambiente de negócio

4. Especificar os ***objetivos estratégicos*** do negócio.

   Listar os objetivos estratégicos do negócio que precisam ser alcançados pela solução técnica do problema. Para cada um dos objetivos especificar:

   4.1 ***Fatores críticos de sucesso***

   4.2 ***Indicadores de desempenho***

   4.3 ***Objetivos Operacionais***

   Descrever os objetivos operacionais em que se decompõe o objetivo estratégico.

5. Definir as ***políticas*** do negócio

   Descrever as políticas do negócio que influenciam o problema a ser resolvido.

   5.1 Definir as ***normas*** do negócio**.**

   Descrever as normas geradas a partir das políticas do negócio, selecionadas, e que têm influência sobre o problema a resolver. Para cada uma das normas informar:

   5.1.1 v***igência***

   5.1.2 c***ritérios de desempenho***

   5.2 Definir os ***procedimentos*** do negócio.

   Descrever os procedimentos gerados a partir das políticas do negócio, selecionadas, e que têm influência sobre o problema a resolver.

6. Descrever a ***estrutura organizacional*** utilizada pelo negócio**.**

Descrever quais as áreas de negócio afetadas pelo problema e como elas se relacionam. Para cada área afetada especificar:

6.1 ***Pessoal***

Descrever os usuários envolvidos e afetados pelo problema a resolver. Informar sua formação e capacitação.

6.2 ***Processo Empresarial***

Descrever os processos empresariais, atuais ou futuros, que são afetados pelo problema. Para cada um dos processos informar :

- ***descrição do processo***
- ***objetivos operacionais***
- ***fornecedores***
- ***clientes***
- ***fluxo de atividades***
- ***recursos***
- ***resultados***
- ***custos***
- ***valor***

6.3 **Papéis**

Descrever os papéis definidos pelos processos selecionados. Para cada um dos papéis estabelecer:

- **usuários** que os desempenham, e
- **atividades** decorrentes**.**

6.4 P***lataforma Tecnológica***

Especificar a plataforma tecnológica (plataforma computacional e plataforma de telecomunicação) da unidade de negócio composta de:

6.4.1 **Plataforma de Hardware**

6.4.2 **Plataforma de Software**

6.4.3 **Plataforma de Comunicação**

7. Definir o ***projeto.***

Especificar o projeto responsável pelo planejamento da solução do problema. descrevendo :

7.1 ***atividades do projeto***

7.2 ***recursos de projeto***

7.3 ***resultados de projeto***

8. Especificar os ***critérios de sucesso*** usados para avaliação do valor do serviço prestado pelos usuários afetados e valor estratégico do sistema de software que apoia a solução do problema.

FORMULÁRIO DE ESPECIFICAÇÃO DE PROBLEMA

1. Fornecer a ***descrição do problema*** a resolver.
   Descrever a estrutura de conhecimento envolvida e/ou referenciar a fonte de informação registrada no formulário de referências bibliográficas (veja item a seguir).

2. Definir a ***terminologia*** do problema.

   Registrar e definir no glossário os termos técnicos específicos do domínio de aplicação e/ou do problema a resolver.

3. Especificar os ***objetivos do problema*** a resolver.

   Listar os objetivos que precisam ser alcançados pela solução técnica do problema. Para cada um dos objetivos especificar:

   3.1 ***Fatores críticos de sucesso***

   3.2 ***Indicadores de desempenho***

4. Descrever as ***soluções alternativas do problema***.

   Caso ainda não seja uma solução completamente consolidada, registrar :

   - Quem esta trabalhando no mesmo assunto ?
   - Quais as soluções dadas ?
   - O que se encontra disponível ?
   - Quais as falhas e/ou riscos ?
   - Quais as vantagens destas soluções ?

   Para isto usar as informações constantes do formulário de referências bibliográficas.

   Caso contrário, para soluções estáveis, não evolutivas, dispensar este tipo de informação.

5. ***Solução técnica*** do problema e o embasamento teórico que a apoia.

   Descrever a solução técnica dada ao problema e qual o embasamento teórico da mesma, ou referenciar a fonte da informação registrada no formulário de referências bibliográficas.

   5.1 Registrar as ***premissas*** assumidas para a solução dada.

   5.2 Descrever os ***fatores críticos de sucesso*** (facilitadores da solução)

   5.3 Descrever os ***riscos*** potenciais associados à solução.

   5.4 Descrever o ***procedimento de cálculo*** (referente a fonte registrada no *formulário de referências bibliográficas*) associado mostrando:

      5.4.1 que ***sistema de equações*** é resolvido

      Registrar o sistema de equações a serem implementadas com a descrição de cada um dos termos e suas unidades de medida;

      5.4.2. quais os ***dados de entrada*** recebidos

      Descrever cada uma das variáveis de entrada, sua unidade e forma de aquisição e/ou geração. Em casos de dados experimentais registrar:

Medida Experimental:

Além de descrever a *parâmetro físico medido* e sua unidade no sistema de equações, informar o ***instrumento de medida*** usado, registrando:

- calibração do equipamento
- precisão dos medidores
- banda morta
- unidade de medida
- fonte de informação usada e registrada no formulário de referências bibliográficas

5.4.3. quais os ***resultados de cálculo*** produzidos

Registrar os resultados esperados e sua ordem de grandeza (faixa de variação válida).

6. Especificar os ***critérios de aceitação de resultados*** usados para avaliação da solução técnica.

FORMULÁRIO DE ESPECIFICAÇÃO DE USUÁRIO

1. Informar a ***identificação*** do usuário.

   Registrar o *nome* (e/ou *código* de identificação) de cada um dos usuários do software. A *localização* não será necessária, uma vez que a natureza de especificação contempla múltiplos usuários em uma unidade de negócio.

2. Descrever os ***usuários.***

   O objetivo é obter subsídios para determinar o comportamento externo do software e configurar a interface homem-máquina. Para isto é preciso :

   - estabelecer os diferentes "*cenários de uso*" do produto/serviço de software e
   - identificar as *necessidades* e *preferências* dos diversos usuários.

   Assim para cada um dos usuários do software descrever:

   - a ***missão do usuário*** ,ou seja , o papel que o usuário desempenha;
   - a ***atividade do usuário***, ou seja, as tarefas que executa e que possuem intercessão com o problema a resolver;
   - o ***estilo de interação***, quais as técnicas e métodos de interação homem-máquina de sua preferência.

3. Definir a ***terminologia*** do usuário.

   Registrar e definir no glossário os termos específicos de uso e domínio de cada um dos usuários envolvidos, relacionados ao domínio do problema a resolver.

4. Definir a ***solicitação*** do usuário.

   Para cada um dos usuários identificar as solicitações a serem feitas ao sistema. A lista de solicitações é a definição dos eventos externos do sistema. As solicitações :

   - definem a ***missão do software*** e
   - podem alterar o ***repositório*** do software e/ou
   - podem disparar um ***processo funcional*** do software e/ou
   - podem definir a ***entrada*** dos processos funcionais do software

5. Definir a ***resposta*** do usuário

   Para cada um dos usuários identificar as respostas esperadas. As repostas podem ser:

   - ***mensagens*** geradas por um processo funcional do software,
   - ***saída*** de um processo funcional que executa um procedimento e produz um *resultado de cálculo*.

6. Especificar os ***critérios de aceitação das respostas*** usados por cada um dos usuários para avaliação das respostas fornecidas pelo software.

FORMULÁRIO DE ESPECIFICAÇÃO DE SISTEMA

1. Definir ***missão do sistema***.

   Descrever a missão do software significa conhecer:

   - *o que* o software vai produzir,
   - *porque* vai produzir,
   - *como* vai produzir,
   - *qual* o perfil de seu *usuário* e
   - *onde* ele irá atuar (meio ambiente).

   Extrair estas informações das solicitações do usuário definidas no *formulário de especificação de usuário*.

2. Definir a ***terminologia*** do sistema.

   Registrar e definir no glossário os termos técnicos específicos da solução computacional do problema.

3. Especificar os ***objetivos do sistema***.

   Extrair de objetivos do problema, *formulário de especificação do problema*, o conjunto de objetivos que deverão ser atendidos pelo software. Para cada um dos objetivos selecionados transcrever:

   3.1 ***Fatores Críticos de Sucesso***
   3.2 ***Indicadores de Desempenho***

   Os objetivos específicos da solução computacional são especificados nos *Requisitos de Arquitetura e Requisitos de Projeto*.

4. Descrever as ***soluções alternativas*** *do sistema* de software.

   Explicitar as soluções computacionais alternativas para o problema:

   - Quem esta trabalhando no mesmo assunto ?
   - Quais as soluções dadas ?
   - O que se encontra disponível ?
   - Quais as vantagens e desvantagens das soluções ?

   justificando a escolha feita. Para isto usar as informações constantes do formulário de referências bibliográficas (ver item a seguir).

   Caso a solução computacional seja óbvia, não evolutiva e já consolidada estas informações tornam-se dispensáveis.

5. ***Solução computacional*** do sistema de software.

   Descrever a solução computacional dada ao problema, executando os itens a seguir:

   5.1 Registrar as ***premissas do sistema.***

Extrair das *premissas do problema*, formulário de especificação do problema, aquelas que são pertinentes à solução automatizada do problema.

Especificar as premissas específicas da solução computacional adotada.

5.2 Descrever o ***desempenho funcional*** esperado para o software.

5.3 Descrever as ***restrições computacionais*** da solução automatizada.

5.4 Especificar os dados a serem armazenados no ***repositório/memória*** do sistema.

Extrair das ***solicitações*** do usuário, *formulário de especificação do usuário*, e ***dados de entrada*** do procedimento de cálculo, *formulário de especificação do problema.*

Especificar os dados internos do sistema a serem definidos nos *Requisitos de Arquitetura e Requisitos de Projeto.*

5.5 Especificar a funcionalidade do solução computacional, explicitando os ***processos funcionais***.

Os *processos funcionais* são definidos sob diferentes pontos de vista, ou seja:

- ótica dos usuários,
- ótica da equipe de operação,
- ótica da equipe de suporte,
- ótica de integração do sistema.

Para cada um dos processos fornecer:

- uma ***descrição*** sumária;
- as ***estímulos*** que recebe;

  Extrair das solicitações dos usuários, formulário de especificação de usuário, e/ou, dados de entrada do procedimento de cálculo, formulário de especificação do problema, quando for o caso.

- o ***algoritmo*** que executa;

  Extrair do *procedimento de cálculo* (*sistema de equações*) da solução técnica, *formulário de especificação* do *problema,* e/ou procedimento do negócio, quando for o caso. Indicar fonte de informação do *formulário de referências bibliográficas*.

  Para o *algoritmo* usado explicitar sua:

  - *precisão*
  - *convergência*
  - *precisão de entrada*
  - *complexidade*
  - *faixa de aplicabilidade*

- as ***respostas/serviços*** que produz;

As saídas produzidas geram uma resposta do usuário, formulário de especificação de usuário, e/ou um resultado de cálculo, do procedimento de cálculo da solução técnica, formulário de especificação do problema.

- as ***mensagens*** que emite;

As mensagens emitidas geram uma resposta do usuário, formulário de especificação de usuário.

5.6 Especificar as ***interfaces***

- do usuário,
- de software,
- de hardware,
- de comunicação,

com subsídios da ***plataforma tecnológica*** do ambiente de negócio (já especificada e definida) e a serem detalhadas nos *Requisitos de Arquitetura.*

6. Especificar os ***critérios de aceitação*** usados para avaliação da solução computacional