

Uma Estratégia para Implantação de uma Gerencia de Requisitos Visando a Melhoria Dos Processos de Software

Ana Elizabete Souza de Carvalho, Helena Cristina Tavares, Jaelson Brelaz Castro
Centro de Informática, Univesidade Federal de Pernambuco
{ana-elizabete.carvalho, helena-cristina.tavares}@serpro.gov.br, jbc@cin.ufpe.br

Resumo. A indústria de software vem demonstrando crescente interesse em engenharia de requisitos, isto é, entender o que se deseja construir antes de começar a fazê-lo. Estão percebendo que o tempo utilizado no entendimento do problema é um excelente investimento. Os requisitos de software são a base a partir da qual a qualidade é medida. Desta forma, a falta de conformidade aos requisitos significa falta de qualidade. O modelo de avaliação de maturidade do processo de desenvolvimento CMM (*Capability Maturity Model*) considera o gerenciamento de requisitos como sendo uma das primeiras etapas para alcançar a maturidade organizacional, e para haver o gerenciamento é preciso que o processo de desenvolvimento de requisitos esteja implantado na empresa. Desta forma, para se alcançar a gerência de requisitos é necessário que os requisitos tenham sido definidos, e é importante que a empresa também possua seus processos de desenvolvimento de requisitos definidos.

Este artigo tem por objetivo definir uma estratégia para a implantação dos processos de desenvolvimento e gerenciamento de requisitos para os projetos de desenvolvimento e manutenção de software sob responsabilidade do SERPRO, estabelecendo um entendimento comum entre o cliente e a equipe de projeto quanto aos requisitos que serão atendidos no projeto de software.

Palavras chaves: requisitos, processos, gerenciamento.

1. Introdução

Ultimamente tem havido um grande interesse na comunidade de engenharia de software na melhoria do processo. As empresas precisam medir a sua capacidade de desenvolver software com qualidade. Para isto, estão utilizando o modelo CMM (*Capability Maturity Model*), que é um modelo gerencial que organiza as melhores práticas existentes, embora os padrões e as práticas que são aplicáveis não sejam completamente definidos.

Em geral, o desenvolvimento de software comercial responde rapidamente às mudanças tecnológicas [1]. Por isso, é importante investir no processo de melhoria contínua para o aumento da qualidade focalizando a engenharia de requisitos.

Encontram-se algumas tentativas de uso de requisitos nas organizações mas, infelizmente, as tentativas começam pela fase do gerenciamento do ciclo de vida e rastreabilidade dos requisitos, iniciada por um processo de avaliação de maturidade do nível organizacional SEI-CMM [2], sem antes ter o domínio da importante fase de descobrimento de requisitos, a partir do descobrimento dos fatos e fenômenos do

ambiente ou domínio da aplicação [3]. Por isso, é importante que a empresa também possua seus processos para o desenvolvimento de requisitos definidos.

Na próxima Seção, é descrita a importância da Qualidade de Software e o contexto do SERPRO. Na Seção 3, os processos a serem executados para a implantação da Gerência de Requisitos que foram definidos pelo SERPRO são descritos. Na Seção 4, as fases utilizadas para a implantação dos processos para a gerência de requisitos definidos são descritas. Na seção 5 é feito um mapeamento entre a proposta apresentada e as práticas do CMM para a Gerência de Requisitos. Uma descrição breve do estudo de caso é apresentada na Seção 6 e, a Seção 7 é composta das conclusões e trabalhos futuros.

2. Importância da Qualidade de Software

A cada dia, as empresas tornam-se mais dependentes dos seus sistemas de informações. Construir esses sistemas, em tempo hábil para serem úteis aos negócios, com a qualidade e custos adequados à sua importância para a organização, é o desafio que todos os desenvolvedores estão enfrentando.

Pressman define qualidade de software como *“conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido”* [4].

Diversos modelos, ferramentas e propostas tem sido projetadas, desenvolvidas e sugeridas nos últimos anos, visando permitir as empresas a se capacitarem evolutivamente para o projeto de software, agregando à cultura empresarial mecanismos de medições e controle, bem como de evoluir toda a técnica utilizada sempre que necessário.

Entre as abordagens para a melhoria da qualidade, destaca-se aquela que atua nos processos utilizados por uma organização de software. O modelo CMM (*Capability Maturity Model*) foi desenvolvido para guiar a melhoria da qualidade, por meio da maturidade da capacidade dos processos de software.

A capacitação do processo permite uma maior previsibilidade de desempenho. A maturidade de um processo de software de uma organização ajuda a prever a habilidade em atingir suas metas. Quanto maior for a capacidade do processo, mais benefícios podem ser alcançados, tanto para o cliente (interno ou externo) quanto para os desenvolvedores [5].

O CMM é um modelo utilizado para medir a maturidade de uma organização nos seus processos de desenvolvimento de software. É um modelo gerencial que organiza as melhores práticas existentes. Segundo o modelo CMM, quanto maior o controle sobre o processo de desenvolvimento, mais madura é a organização. É organizado em cinco níveis de maturidade, considerados como a principal característica do modelo. Nível de maturidade é um estágio evolutivo bem definido para alcançar um processo de software maduro. Cada nível de maturidade indica um nível de capacidade do processo, visando a melhoria contínua do processo de desenvolvimento de software [6]. Com exceção do nível 1, cada nível é composto por várias Áreas-chave de Processo (ACPs). Estas áreas conduzem ao alcance das metas de melhoria de um determinado nível [7].

A primeira ACP do CMM para atingir o Nível 2 (figura 3) é a *Gerência de Requisitos*, a qual estabelece as diretrizes para um comum entendimento entre o cliente e os requisitos de seu projeto de software, que serão conduzidos pela equipe do desenvolvimento. Este acordo com o cliente é a base para planejar e administrar o projeto de software. O controle das relações com o cliente depende de um efetivo controle do processo da mudança. Mas, para que a Gerência de Requisitos seja implantada na organização é necessário que um processo de Engenharia de Requisitos esteja implantado.

Como Empresa de Tecnologia da Informação, o SERPRO atua num mercado em permanente ebulição, por isso iniciou um processo para identificação e implantação das melhores práticas para elevar o estágio de maturidade do processo de desenvolvimento de *software*.

Impulsionando este processo está a meta de atingir o nível 2 de maturidade segundo o modelo CMM (*Capability Maturity Model*) do SEI até o final de 2002. E, os trabalhos iniciais verificaram que a falta de uma efetiva gerência de requisitos é um dos principais problemas a serem superados.

Dando início à estruturação interna de uma gerência de requisitos, o SERPRO participou do projeto *Plataforma Tecnológica em Engenharia de Requisitos - Estratégias para o Aumento da Qualidade no Desenvolvimento de Sistemas* [8], o qual teve como objetivo estabelecer bases para o aumento da qualidade dos processos de produção de software, por meio da cooperação entre universidades e empresas, com ênfase na utilização da Engenharia de Requisitos.

A plataforma foi extremamente importante para um melhor conhecimento dos problemas enfrentados pelas organizações que participaram do projeto e da possibilidade da cooperação com as instituições de pesquisa para minorar esses problemas.

Identificados os problemas, os processos a serem executados para a implantação da Gerência de Requisitos foram definidos pelo SERPRO e são descritos na próxima Seção.

3. O Processo para Implantação da Engenharia de Requisitos visando a melhoria da Qualidade de Software

A definição dos processos para a gerência de requisitos implantados no SERPRO foram baseadas nas boas práticas existentes na empresa, na cultura da organização e nas características de seus clientes. Pesquisas em bibliografia existente, o apoio de instituições de pesquisa e a utilização de *benchmarking* foram importantes para o desenvolvimento do projeto.

A figura 1 resume a estrutura adotada. Para alcançar as metas, algumas atividades precisam ser executadas por pessoas com papéis bem definidos. O responsável por analisar e selecionar os requisitos coleta os requisitos de software, os quais são revistos pelo Grupo de Engenharia de Software antes de sua incorporação no projeto. Uma vez incorporados, os requisitos formam uma *baseline*, que será a base para o Contrato Técnico entre o cliente e o desenvolvedor. Depois de validados, os requisitos são utilizados pelo Grupo de Engenharia de Software para planejar os artefatos e as atividades do desenvolvimento que irão compor o Plano de Desenvolvimento de

Software. Os desvios e as solicitações de alteração de requisitos são revistos antes de serem incorporados, seus impactos são avaliados, e os compromissos assumidos são renegociados com as partes interessadas (*stakeholders*).

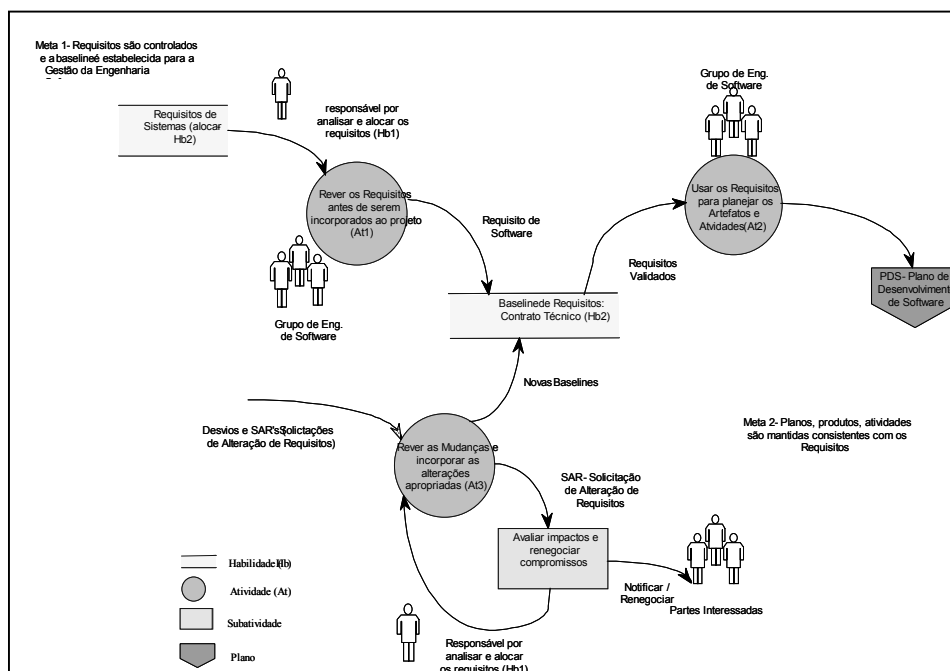


Figura 1. Área-Chave de Processo Gerência de Requisitos

Os processos devem definir as atividades que deverão ser executadas para que as metas sejam atingidas. Nesta proposta, os processos da Gerência de Requisitos foram divididos em: *Definição*, *Gerenciamento & Métricas*, *Validação* e *Verificação* (figura 2) e definem as atividades a serem executadas por uma empresa que desenvolve sistemas a partir das solicitações de um cliente.

Ao receber uma Solicitação de Serviço, o processo de *Definição* elabora o documento de *Visão*, o Glossário, o *Documento de Requisitos de Software (DRS)* e as *Matrizes de Rastreabilidade*. Caso uma *Solicitação de Alteração de Requisitos (SAR)* seja recebida, o processo para *Gerenciamento & Métricas* de requisitos, juntamente com as *Matrizes de Rastreabilidade* e o *Documento de Requisitos de Software (DRS)*, analisa os *requisitos impactados*, gera uma nova versão atualizada dos requisitos (*baseline*), atualiza o *Documento de Requisitos de Software (DRS)* e comunica as alterações às *áreas envolvidas*. Neste processo também são coletadas as métricas para controle. Durante o processo de *Validação*, o *Documento de Requisitos de Software (DRS)* e demais artefatos são avaliados para a elaboração dos *Casos de Testes*, além das comunicação das *áreas envolvidas*. O processo de *Verificação* recebe os artefatos dos modelos do processo de desenvolvimento e avalia se estão de acordo com os requisitos definidos.

As figuras 2 e 3 são apresentadas apenas com as entradas e saídas da técnica IDEF0. A Tabela 1 descreve as *principais* entradas e saídas dos processos descritos anteriormente.

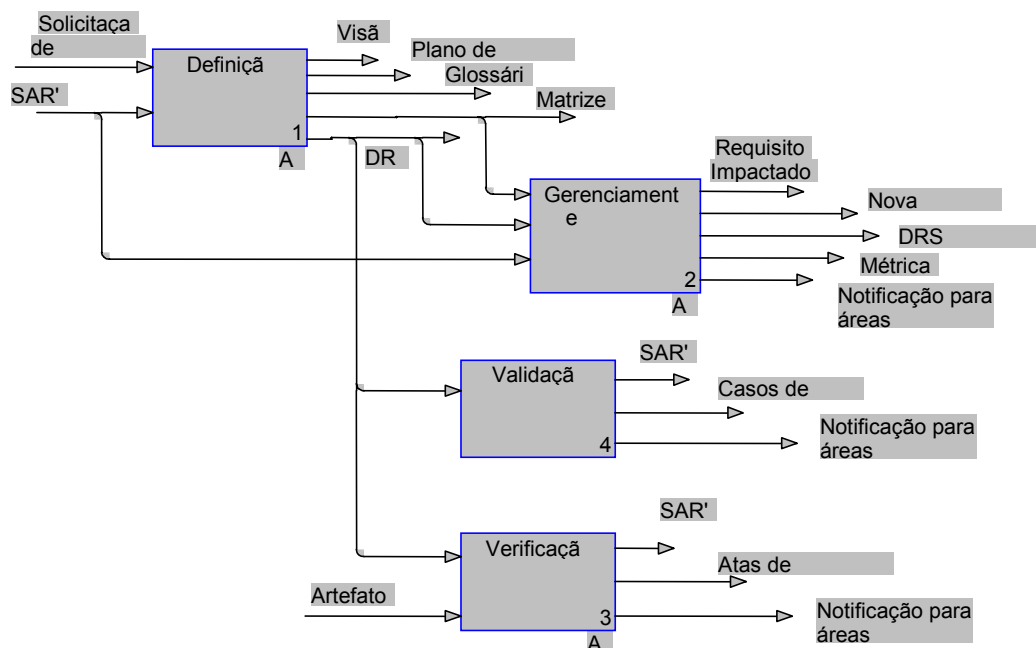


Figura 2. Grupos de Processos para a obtenção da Gerência de Requisitos

A figura 4 apresenta o *detalhamento* do processo para a definição de requisitos que foi exibido no macroprocesso da figura 2. É composto de *Elicitação*, *Análise & Documentação* e *Negociação & Priorização* (figura 4).

O processo de *Elicitação* dos Requisitos de Software recebe novos requisitos por meio de uma *Solicitação de Serviço* ou *Solicitação de Alteração de Requisitos* entre outras fontes, e uma vez documentados e analisados, são elaboradas as *Matrizes de Rastreabilidade* e o *Documento de Requisitos de Software (DRS)* entram num processo para negociação de prioridades, conforme detalhado a seguir. A *Elicitação* pode ser subdividida nas seguintes atividades, e seguem a ordem seqüencial e o paralelismo explicitado na figura 6 a qual utiliza o diagrama de atividades da UML [9]:

- *Efetuar a reunião inicial com o cliente:* Devem participar da reunião inicial com o cliente todos as partes interessadas (*stakeholders*) candidatos. Nesta reunião devem ser obtidos os principais assuntos que deverão estar presentes no Documento de Visão.

ENTRADA OU SAÍDA	TIPO	DESCRIÇÃO
Solicitação de Serviço	entrada	Refere-se ao documento formal por meio do qual o sistema é solicitado pelo cliente.
SAR (Solicitação de Alteração de Requisitos)	entrada	Documento que deve ser preenchido quando há necessidade de alteração ou inclusão de requisitos. Especifica e documenta a alteração de requisitos solicitada pelo <i>stakeholder</i> , estabelecendo as bases para sua execução, e condições de execução (prioridades, recursos envolvidos, recursos afetados, etc).
Visão	saída	Referem-se a informações gerais sobre o sistema. Documento onde é fornecido o objetivo do sistema a ser desenvolvido; suas principais características, funcionalidades e não-funcionalidades; os <i>stakeholders</i> . Seu objetivo é apresentar o sistema em linhas gerais, fornecendo uma visão inicial. Documenta o ambiente geral de processos a ser desenvolvido, fornecendo a todos os envolvidos uma descrição compreensível do sistema e suas macro-funcionalidades. O objetivo do documento Visão é definir as necessidades de alto-nível e características do sistema, focando nas potencialidades requeridas pelos <i>stakeholders</i> e como estes requisitos serão abordados no sistema.
Glossário	entrada e saída	O objetivo do glossário é fornecer uma terminologia comum sobre o projeto a todos os envolvidos. Este conjunto de termos e conceitos originais é usado para definir um mapeamento específico sobre o domínio do problema, explicando os itens mais relevantes e as descrições de outros componentes do projeto. Além de facilitar o entendimento da aplicação, o Glossário evita ambigüidades no entendimento das informações, ao longo do processo.
Rastreabilidade	entrada e saída	Neste contexto, as Matrizes de Rastreabilidade mostram os relacionamentos entre os requisitos elicitados.
DRS (Documento de Requisitos de Software)	entrada e saída	É composto pelo detalhamento dos requisitos. A DRS guarda a especificação dos requisitos de software, descritos num nível de detalhe que possibilite o entendimento de todos os envolvidos. A DRS serve como base de comunicação entre os <i>stakeholders</i> , e ajuda a manter um foco integrado no desenvolvimento.
Requisitos Impactados	saída	Referem-se aos requisitos que serão alterados em decorrência da inclusão ou alteração de requisitos.
Nova Baseline	saída	<i>Baseline</i> ou configuração base é um conjunto de requisitos aprovados e revisados que representa o embasamento de um acordo de evolução e desenvolvimento; atualizada por meio da inclusão ou alteração de requisitos.
DRS atualizado	saída	Corresponde ao Documento de Requisitos de Software alterado em função da inclusão ou alteração de requisitos.
Notificação as áreas envolvidas	saída	Refere-se à comunicação a todos os que serão impactados pela inclusão ou alteração de requisitos.
Métricas	saída	Informações coletadas ao longo do gerenciamento de requisitos.
Casos de Testes	saída	Devem estar incluídos no DRS (Documento de Requisitos de Software).
Atas de Verificação	saída	Será utilizada a Ata de Reunião para registrar as não-conformidades.
Plano de Revisão	saída	Será utilizado a Ata de Reunião para registrar os compromissos de revisão.

Tabela 1. Entradas e Saídas dos processos descritos

- *Entender o domínio da aplicação:* A reunião inicial com o cliente fornecerá a base para determinar quais os principais domínios serão necessários estudar mais profundamente para o desenvolvimento do sistema. Uma das principais dificuldades enfrentadas pelo desenvolvedor de *software* é a complexidade do domínio do problema [10]. Além disso, os usuários e desenvolvedores possuem perspectivas diferentes e fazem suposições diferentes sobre a natureza da solução [11]. Quando o desenvolvedor e o usuário compartilham a mesma linguagem, melhora a comunicação entre ambos. Em particular, se esta linguagem é própria do usuário esta comunicação melhora consideravelmente [12].
- *Identificar as partes interessadas (stakeholders):* Identifique as pessoas interessadas pelo produto, ou participarão do seu desenvolvimento. Para cada stakeholder é importante identificar além da função, o nome do *stakeholder*.

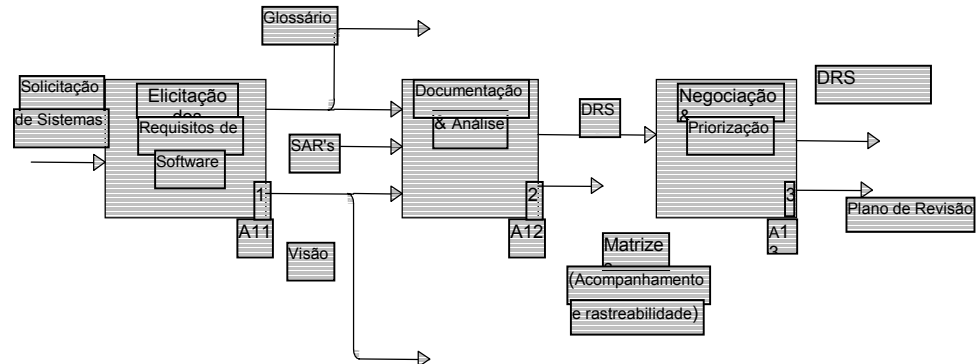


Figura 3. Processos para a Definição de Requisitos

- *Escolher a técnica de elicitação:* Ao escolher a técnica de elicitação deve-se considerar quais as fontes dos requisitos, a disponibilidade das pessoas e os tipos de requisitos a serem elicitados. A escolha da técnica apropriada para elicitar requisitos depende do tempo e recursos disponíveis, assim como do tipo de informação necessária. Algumas das classes de técnicas de elicitação são [13]:
 - Técnicas tradicionais – inclui o uso de questionários, entrevistas, análise de documentação existente [14];
 - Questionários: questões predefinidas são distribuídas para uma amostragem significativa e representativa de *stakeholders* e os resultados são avaliados. São úteis quando a quantidade de *stakeholders* é extremamente grande. Uma vez que todas as questões podem ser predeterminadas, é mais eficiente na avaliação de tendências de opiniões a respeito de requisitos específicos e bem definidos.
 - Entrevistas: Perguntar questões relacionadas aos *stakeholders* e/ou suas necessidades a respeito do problema a ser resolvido e

escutar suas respostas [15]. São úteis quando stakeholders possuem muitos conhecimentos subjetivos e estão dispostos a serem entrevistados. Para ser mais eficiente, o entrevistador deve ser experiente.

- Técnicas de elicitação de grupo – são técnicas de dinâmica de grupo com o objetivo de entender de forma mais detalhada as necessidades dos usuários, estão incluídas: brainstorming, sessões JAD e RAD [16];
 - *Brainstorming: stakeholders* são reunidos em um local, num ambiente que encoraje a participação, permitindo que todas as idéias sejam declaradas em voz alta (para que os demais sejam influenciados) e escritas (para que não sejam perdidas) [17]. É mais eficaz quando cada stakeholder possui uma parte do conhecimento de algum aspecto do problema.
- Prototipação – implementação parcial de um sistema de forma rápida para obter *feedback* para o processo de requisitos. O protótipo é descartado. É utilizada para elicitar requisitos quando há um alto grau de incerteza ou quando é necessário um rápido feedback dos usuários;
- Técnicas de modelagem – fornece um modelo específico das informações que serão adquiridas, e usa esse modelo para orientar o processo de elicitação. Uma técnica bastante utilizada é o uso de cenários para representar as tarefas que os usuários executam normalmente e aquelas que eles desejam executar [18]. Inclui métodos baseados em metas, tais como: KAOS [19] e I* [20] e métodos baseados em cenários como CREWS [21].
- Técnicas cognitivas – inclui uma série de técnicas originalmente desenvolvidas para aquisição de conhecimento para sistemas baseados em conhecimento, alguns exemplos são: Análise de protocolo, laddering, card sorting, repertory grids [22];
- Técnicas contextuais – surgiu como uma alternativa para as técnicas tradicionais e cognitivas, inclui técnicas de etnografia e análise social [23];
- Etnografia: observar potenciais usuários em seu ambiente natural. Resulta em uma percepção mais precisa do problema do que perguntar aos usuários o que eles fazem [24]. São úteis para suporte à automação de uma função pouco ou não automatizada, particularmente quando são disponíveis a observadores treinados sem noções preconcebidas do problema e de sua solução.

Condições específicas do projeto devem definir a técnica mais eficaz a ser utilizada [25], ou a combinação delas. No SERPRO, as técnicas mais utilizadas são a entrevista, o uso de cenários com *Use Cases* e a prototipação, com o acompanhamento do cliente não só na fase de requisitos, mas durante todo o processo de desenvolvimento do software. Um benefício é o comprometimento do cliente e a participação direta na definição dos

documentos. Um cuidado, porém, deve ser tomado para a tendência da não utilização dos processos para alteração de requisitos.

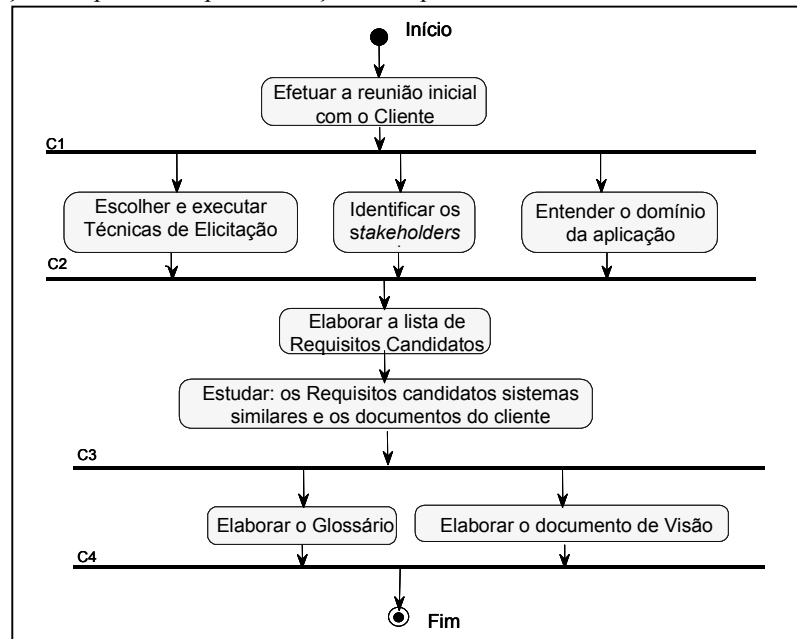


Figura 4. Processos para a Elicitação de Requisitos

- *Elaborar a lista de Requisitos Candidatos:* o objetivo desta atividade é descrever os principais requisitos do sistema, de forma a obter a essência do sistema, de forma que representará as principais necessidades e funcionalidades do sistema e deverá ser incluída no documento de Visão.
- *Estudar os Requisitos Candidatos, os sistemas similares e os documentos coletados com o cliente:* documentos devem ser inspecionados, arquivos e sistemas similares, para melhor compreender o contexto do sistema, e avaliar a possibilidade de reusabilidade de requisitos e de soluções.
- *Elaborar o Glossário:* para evitar ambigüidades e facilitar a leitura de documentos do sistema, é importante a definição de um glossário, onde os principais termos utilizados pelo domínio da aplicação e pelos desenvolvedores e usuários devem ser definidos.
- *Elaborar o documento de Visão:* o objetivo desta atividade é descobrir os principais requisitos do sistema, de forma a obter a essência do sistema. Deve-se avaliar o conjunto de requisitos essenciais para a definição do Documento de Visão do software e este deve incluir o escopo do projeto e suas limitações, bem como as principais características do software a ser desenvolvido.

A *Documentação & Análise* (figura 7) deve possuir as seguintes atividades:

- *Descrever os requisitos e os seus atributos*: depois que os requisitos do sistema são elicitados, eles devem ser documentados com um nível apropriado de detalhes. Na maior parte das organizações os requisitos são escritos em linguagem natural, pois é uma notação que é de fácil entendimento para uma grande variedade de *stakeholders*. Entretanto, o nível de abstração dos requisitos varia de acordo com a organização e deve ser definido de acordo com o projeto ou até mesmo com o tipo de requisito. O principal foco da pesquisa em documentação de requisitos é prover notações e linguagens de especificação. Desde linguagem natural [26] à lógica [27], diferentes linguagens têm sido propostas para expressar e descrever requisitos. Pesquisas atuais têm reconhecido que o gerenciamento de requisitos é uma atividade crucial no processo de engenharia de requisitos, ou seja, é necessário não somente escrever os requisitos de forma entendível mas também permitir que eles possam ser rastreados e gerenciados ao longo da evolução do sistema [28].

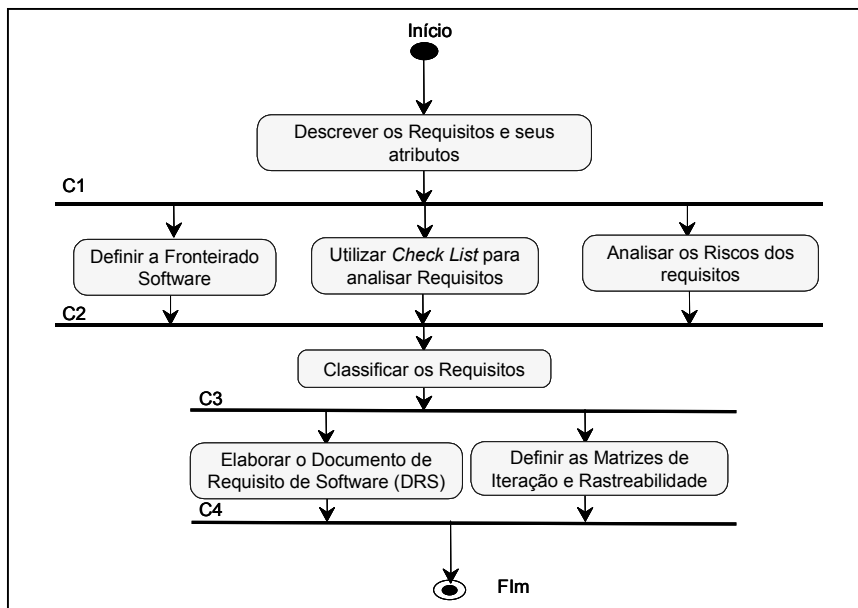


Figura 5. Processo Documentação & Análise

- *Definir a Fronteira do Software*: freqüentemente os *stakeholders* não têm certeza do que deve ou não estar no software a ser desenvolvido, o que faz com que requisitos desnecessários sejam incluídos durante a fase de Elicitação de requisitos. Por isso, deve-se avaliar o conjunto de requisitos essenciais para a definição do *Documento de Visão* do software. O *Documento de Visão* deve incluir o escopo do projeto e suas limitações, bem como as principais características do software a ser desenvolvido [29]. É importante que seja documentado os argumentos técnicos e/ou econômicos que justificam a exclusão dos requisitos do escopo do software.

- *Utilizar um Checklist para analisar os requisitos:* para facilitar, otimizar e tornar mais completa a análise de requisitos, deve-se definir um *checklist*, por meio do qual cada requisito deve ser analisado. Além de reduzir a probabilidade de erros, a utilização de um *checklist* é uma forma de reutilizar o conhecimento em análise de requisitos entre diferentes projetos. Caso fique muito grande deve-se particioná-lo, de forma a criar vários *checklists* que possam ser distribuídos para diferentes analistas. O *checklist* deve ser periodicamente revisado e validado por analistas experientes para verificar a necessidade de alterações. É importante conscientizar os analistas de que o *checklist* deve ser apenas um guia e que podem existir outros problemas não cobertos pelo *checklist*.
- *Analisar os Riscos dos requisitos:* é uma forma de identificar os requisitos que poderão causar mais problemas aos desenvolvedores. Deve-se identificar os problemas que poderão surgir, a probabilidade destes problemas e os efeitos decorrentes desses problemas para cada requisito ou para grupos de requisitos. Explicitar os riscos associados aos requisitos nesta fase é uma forma de avaliar se os requisitos estão incompletos, se precisam ser modificados ou se é necessária a definição de procedimentos para redução da probabilidade do problema ocorrer e do impacto, caso ocorra.
- *Classificar os requisitos:* o objetivo da classificação dos requisitos é agrupá-los de forma a identificar requisitos semelhantes. É importante que o número de classes definido não seja muito grande, pois ficam muito poucos requisitos por classe. Geralmente, fica em cinco ou seis tipos de requisitos. Uma vez classificados, os requisitos de uma mesma classe devem ser comparados e analisados, pois conflitos e sobreposições são mais frequentes em requisitos de um mesmo tipo. Requisitos similares também podem ser comparados e pode-se visualizar melhor a falta de algum requisito. A classificação dos requisitos deve ser definida no *Guia de Classificação de Requisitos*.
- *Definir as Matrizes de Iteração e Rastreabilidade:* numa matriz de interação, cada requisito deve ser comparado com os demais de forma a identificar se são conflitantes, sobrepostos ou independentes. Se a quantidade de requisitos for muito grande, deve-se definir a matriz para uma classe específica de requisito ou para os requisitos de um subsistema.
- *Elaborar o Documento de Requisito de Software (DRS):* o documento de requisitos é o meio através do qual é possível descrever as restrições quanto às características do produto e quanto ao processo de desenvolvimento, a interface com outras aplicações, a descrição sobre o domínio e as informações de suporte ao conhecimento do problema [30]. Em geral, é necessário que o documento de requisitos seja entendível por todos os envolvidos no processo de engenharia de requisitos pois ele servirá como um contrato entre usuários e desenvolvedores.

Durante a *Negociação & Priorização* (figura 8) as seguintes atividades são feitas:

- *Escolher a técnica de negociação:* o processo de negociação de requisitos tenta resolver conflitos entre usuários sem necessariamente comprometer a

satisfação dos objetivos de cada usuário. Em geral, os modelos de negociação identificam as principais necessidades de cada usuário, ou seja, atribuem prioridades aos requisitos, em seguida analisam esses resultados para tentar garantir que os requisitos mais críticos sejam atendidos. Em [31], Karlsson descreve um estudo de caso realizado num projeto Ericsson em que duas importantes técnicas para priorização e negociação de requisitos foram comparadas. As técnicas avaliadas foram AHP [32] e QFD [33], os resultados obtidos indicaram que a técnica AHP apesar de ter sido considerada complexa, apresentou melhores resultados.

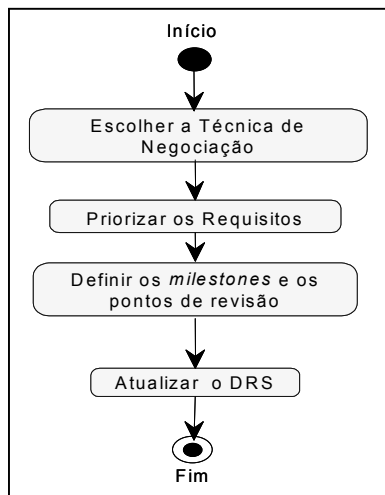


Figura 6. Processo Negociação & Priorização

- *Priorizar os Requisitos*: uma vez escolhida a técnica de negociação, ela deve ser usada para não apenas para resolver conflitos, mas também para priorizar os requisitos. Ao priorizar os requisitos, é importante que os riscos e a complexidade dos requisitos sejam observados de forma mitigar possíveis atrasos nos milestones a serem definidos.
- *Definir os milestones e os pontos de revisão*: de acordo com a priorização, os requisitos são agrupados para estabelecer baselines de implementação, facilitando a definição dos milestones e pontos de revisão.
- *Atualizar o Documento de Requisitos de Software*: o resultado obtido a partir da técnica de priorização deve ser incorporado ao Documento de Requisitos de Software, enriquecendo a documentação sobre os atributos dos requisitos.

Durante processo de *Gerenciamento & Métricas* (figura 9) as seguintes atividades são executadas:

- *Receber as Solicitações de Alteração de Requisitos (SAR)*: O grupo de engenharia de requisitos recebe as solicitações de alteração de requisitos ou

por formulário padronizado ou por meio de sistema de solicitação de demandas.

- *Registrar novos requisitos:* Os novos requisitos também devem ser recebidos formalmente ou por formulário padronizado ou por meio de sistema de solicitação de demandas.
- *Verificar requisitos relacionados:* uma vez preenchidas as matrizes de rastreabilidade, de forma que tenham sido estabelecidos relacionamentos de rastreamento entre os requisitos, é necessário verificar os requisitos relacionados para que seja avaliado o impacto desta inclusão ou alteração posteriormente.

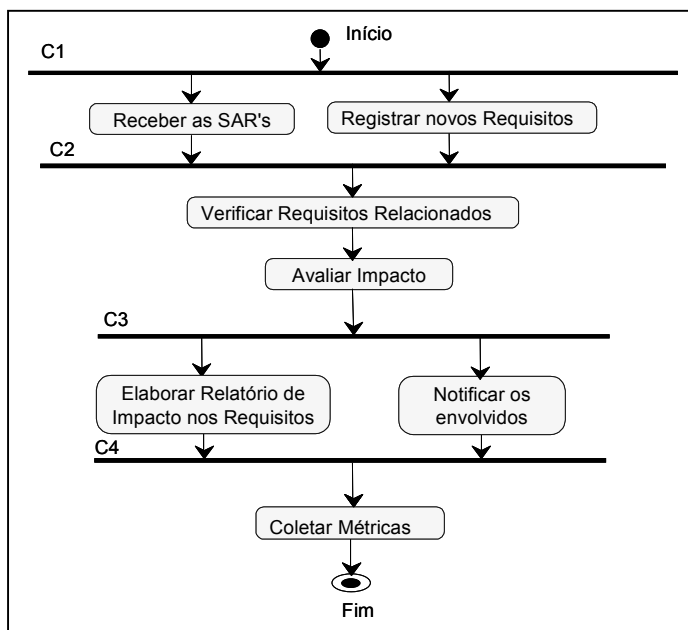


Figura 7. Processo de Gerenciamento & Métricas

- *Elaborar relatórios de Impacto nos Requisitos:* uma vez decidida a alteração, é mantido um histórico de alterações para cada requisito, permitindo uma visão cronológica das principais mudanças nos requisitos, inclusive em seus atributos. O autor, a data-hora, a descrição e a justificativa da alteração também são documentados.
- *Notificar os envolvidos:* todos os envolvidos devido à alteração dos requisitos devem ser notificados.
- *Coletar Métricas:* as métricas desempenham um papel essencial na detecção dos desvios, fornecendo meios para a visualização de discrepâncias e identificação de pontos fora de uma situação projetada. As métricas devem ser utilizadas e coletadas periodicamente para o devido acompanhamento das atividades da Gerência de Requisitos, fornecendo também uma indicação da

volatilidade dos requisitos. A equipe de gerência de requisitos deve definir quais métricas serão utilizadas no projeto, de acordo com suas características peculiares.

Durante processo de *Validação* (figura 10) as seguintes atividades são executadas:

- *Usar Checklist de Validação*: *checklists* estruturam o processo de validação de forma que aspectos importantes não sejam esquecidos. Embora sejam semelhantes aos *checklists* de análise, os *checklists* de validação devem focalizar aspectos de qualidade do documento de requisito como um todo, bem como os relacionamentos entre os requisitos. Algumas questões podem ser relacionadas para checar se algum requisito está ausente; se os requisitos estão consistentes e não contraditórios; se o documento está estruturado de forma a facilitar o entendimento; se foi feito o rastreamento dos requisitos; ou se o documento está de acordo com os padrões.
- *Definir os casos de teste dos requisitos*: definir um ou mais casos de testes que posteriormente permitam verificar se o sistema satisfaz o requisito, além de uma forma de validar os requisitos, pode servir como base para os testes finais do sistema. Ao escrever o caso de teste, o requisito estará sendo escrito de um ângulo diferente. Cada caso de teste deve possuir, no mínimo, um identificador, os requisitos a ele relacionados e a descrição do teste. Se utilizados em uma ferramenta, os casos de testes podem estar diretamente associados aos requisitos, além de permitir que os testes possam ser feitos automaticamente.

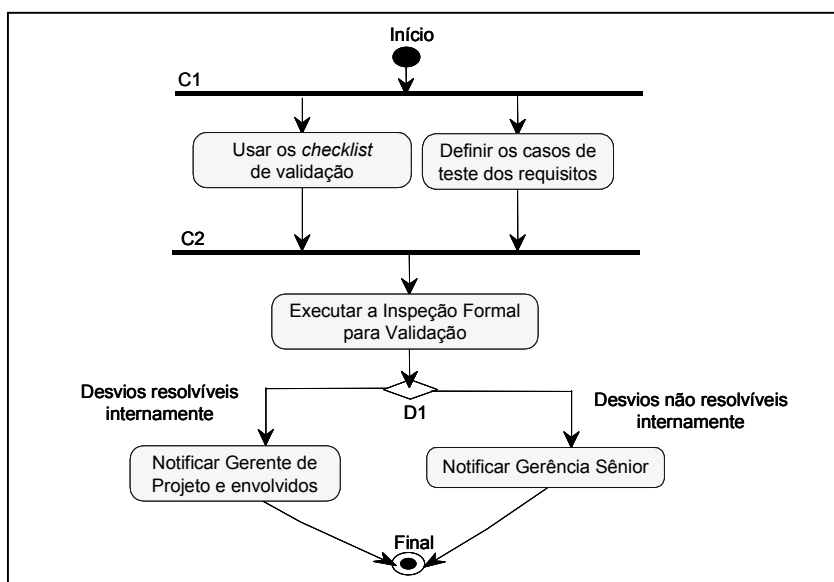


Figura 8. Processo de Validação

- *Executar as Inspeções Formais*: nas inspeções formais um grupo deve validar os requisitos de forma sistemática, descobrir os problemas relacionados a eles e entrar em um acordo a respeito de como solucioná-los.

Durante o encontro, que deve ser coordenado por alguém que não esteja envolvido na definição dos requisitos a serem validados, o engenheiro de requisitos deve apresentar os requisitos e os problemas encontrados devem ser anotados para posterior discussão. Diferentemente de inspeções de programas [34], onde os erros são simplesmente informados ao autor do programa, as inspeções de requisitos devem definir as ações a serem tomadas para a devida correção.

Durante processo de *Verificação* (figura 12) as seguintes atividades são executadas:

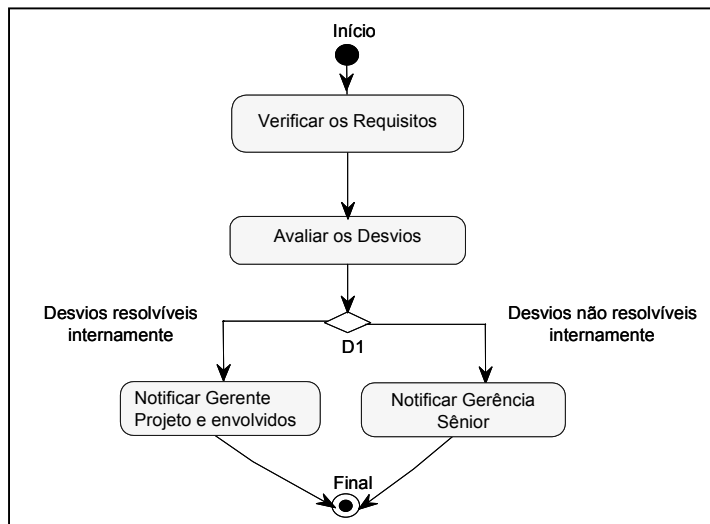


Figura 9. Processo para Verificação

- *Verificar os Requisitos:* é importante verificar se os requisitos encontram-se implementados corretamente nos modelos do processo de desenvolvimento de *software*. É interessante que esta verificação seja feita por pessoas que conheçam os modelos e a padronização adotada pela empresa, mas não tenham se envolvido na especificação dos requisitos. Uma vez sendo encontradas inconsistências, estas devem ser avaliadas. Neste processo, podem ser utilizadas matrizes de rastreabilidade com os modelos do processo de desenvolvimento de software definidos.
- *Avaliar os Desvios:* o objetivo é avaliar junto aos grupos envolvidos o impacto das inconsistências ou desvios encontrados. É importante que seja avaliado, pois muitas vezes os desvios podem originar alterações nos requisitos.
- *Notificar os Gerentes e as Partes Interessadas:* caso os desvios possam ser resolvidos internamente, a gerência de projeto e os envolvidos são notificados. Caso contrário, a gerência sênior é comunicada para tomada de ações corretivas e acompanhamento.

4. As Fases para Implantação

Inicialmente, foram definidos os processos (vide Seção 3) e os padrões documentados que deverão ser utilizados, conforme a política de Gerência de Requisitos aprovada para o SERPRO. Também foi definido um plano de treinamento para ser utilizado pelas equipes.

Em cada regional foi escolhido um projeto para ser desenvolvido utilizando os procedimentos para a Gerência de Requisitos definidos. A equipe deste projeto deve estar preparada para disseminar o conhecimento para as demais equipes do Pólo, aproveitando os conhecimentos adquiridos.

Primeiramente algumas atividades foram executadas para a avaliação da situação da Gerência de Requisitos e definição da política, dos procedimentos, dos padrões documentados e do plano de treinamento que deverão ser utilizados pelas equipes na implantação da Gerência de Requisitos.

Para a implantação do processo de melhoria, nove fases (figura 14) devem ser seguidas até que se consiga o resultado esperado. Ao longo de toda a implantação os processos devem ser avaliados e continuamente ajustados de acordo com o *feedback* obtido. O processos podem selecionados para serem implantados ou omitidos de acordo com as características e necessidades da organização. Por exemplo, uma organização que já possui profundo conhecimento do negócio do cliente pode omitir a atividade *entender o domínio da aplicação*.

A seguir, são descritas as nove fases que foram seguidas para implantação dos processos:

- *Fase 1 - Conscientização*: nesta fase, deve-se identificar a situação atual, definir o *status* desejado (estabelecendo metas), e tomar ações para reduzir a distância entre o status atual e o almejado. O *status* atual pode ser obtido por meio de observações e de questionários e entrevistas aos desenvolvedores em relação ao conhecimento e utilização dos Processos de Engenharia de Requisitos. Uma vez obtidas as informações, estas devem ser consolidadas e apresentadas aos gerentes e desenvolvedores, o que pode ser feito por meio de uma *workshop*, de forma a conscientizar e mobilizar os gerentes e desenvolvedores para a promoção das práticas e mudanças requeridas pelo programa. Neste evento podem ser organizadas palestras sobre o tema. É fundamental o apoio gerencial para as ações e iniciativas de aprimoramento de processos de Engenharia de Requisitos. Manter esse comprometimento e repassá-lo para os analistas é parte fundamental do processo de melhoria.
- *Fase 2 - Preparação*: durante a preparação devem ser escolhidos os grupos de trabalho. Estes grupos devem estar profundamente familiarizados com a filosofia e os termos utilizados nos processos de Engenharia de Requisitos. O grupo de trabalho *GT-Requisitos* é definido para planejar e acompanhar todas as tarefas do projeto de melhoria da capacidade. Este grupo visa garantir que o processo de implantação do modelo como um todo flua de maneira adequada. Este grupo deve aprovar, com a participação da alta gerência, um *cronograma* com as macro-atividades e deve modificá-lo sempre que se fizer necessário. É da responsabilidade do *GT-Requisitos* acompanhar e atualizar este *cronograma*. Dependendo do tamanho da

organização podem ser definidos subgrupos de trabalho, onde o coordenador deve ser um membro do GT-Requisitos. O grupo de trabalho *GT-Pessoas-chave* deve ser composto por pessoas que detêm o conhecimento de como é o processo de desenvolvimento e manutenção de software na empresa. Devem ser entrevistadas e treinadas primeiro de acordo com a necessidade e a disponibilidade. Este grupo pode mudar à medida que aumenta a compreensão dos processos de software. É importante que nesta fase esteja circulando na empresa informações a respeito do trabalho sendo desenvolvido, o que pode ser feito por meio de jornais internos, circulares, e-mail ou circulação de revistas e livros que tratem sobre o tema. Pode-se também disponibilizar informações na Intranet da empresa.

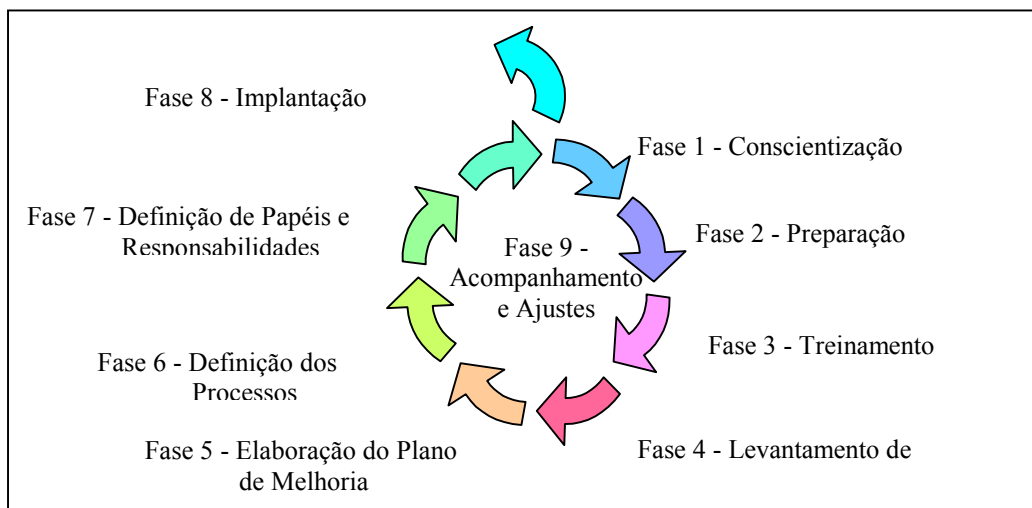


Figura 10. Fases para a Implantação

- *Fase 3 - Treinamento:* o GT-Requisitos também deve definir um *Plano de Treinamento*, priorizando a participação das pessoas que compõe os grupos de trabalho. O treinamento deve preferencialmente ser preparado e aplicado por integrantes do GT-Requisitos, pois deve ser dada uma atenção especial para a adaptação dos termos do modelo de melhoria para a cultura da empresa. Todos os termos devem ser definidos e explicados fazendo, quando possível, um mapeamento para os termos equivalentes utilizados na empresa. A importância do treinamento deve-se também à oportunidade de trocar experiência entre diferentes equipes de desenvolvimento, especialmente em empresas com diferentes pólos de desenvolvimento, cada qual trabalhando com um ambiente e características de projeto próprias. Durante o treinamento, é importante que um componente do GT-Requisitos anote as sugestões e considerações que porventura apareçam, pois podem ser de grande utilidade tanto para os próximos treinamentos como para utilização ao longo do processo de melhoria.

- *Fase 4 - Levantamento de Processos Engenharia de Requisitos existentes:* o primeiro passo para a melhoria é o conhecimento de como está o seu processo de desenvolvimento. Uma empresa pode ter várias maneiras de executar uma mesma tarefa como desenvolver e manter software. É importante que o GT-Requisitos tome conhecimento de todos os processos de Engenharia de Requisitos existentes na empresa. Este conhecimento deverá ser provido por reuniões, entrevistas e coletas de documentos com o GT-Pessoas-chave. Deve-se avaliar as políticas, padrões e processos atuais de Engenharia de Requisitos existentes na organização, identificando os pontos fortes e fracos.
- *Fase 5 - Elaboração do Plano de Melhoria:* o *Plano de Melhoria* do Processo de Engenharia de Requisitos é a base para o futuro esforço de melhoria e deve estar de acordo com o cronograma definido anteriormente. O GT-Requisitos deve preparar o *Plano de Melhoria* contendo os processos e documentos existentes de modo que fique claro os pontos em que há necessidade de melhoria e o impacto benéfico que pode causar ao processo. Pode recomendar ações para solucionar as carências identificadas e até estimar o tempo e recursos necessários para cada atividade de melhoria.
- *Fase 6 - Definição dos Processos:* a definição dos Processos a serem implantados deve ser elaborada tendo como base com as boas práticas existentes na empresa, o Plano de Melhoria definido, o tipo, as restrições e a cultura da organização. São importantes pesquisas em bibliografia existente, instituições de pesquisa e, se possível, a utilização de *benchmarking*.
- *Fase 7 - Definição de Papéis e Responsabilidades:* nesta fase devem ser especificados os papéis e as responsabilidades das pessoas que executarão as atividades de desenvolvimento e gerenciamento de requisitos. Deve-se definir um mapeamento para as funções existentes na Empresa.
- *Fase 8 - Implantação:* a fase de *Implantação* deve possuir o *planejamento* e a *execução*, inicialmente para o projeto-piloto e posteriormente, para os demais projetos. Para iniciar a implantação é importante que o GT-Requisitos defina um *Plano para Projeto-piloto*, onde seja especificado o projeto escolhido, as pessoas envolvidas, os treinamentos necessários, além os prazos esperados. A implantação do projeto-piloto deve ser acompanhada pelo GT-Requisitos e as dificuldades encontradas devem ser documentadas para posterior análise e avaliação dos Processos definidos.
- *Fase 9 – Acompanhamento e Ajustes:* esta fase deve acompanhar a aplicação das práticas de Gerência de Requisitos implantadas, avaliando-as e promovendo os refinamentos necessários. Nesta fase, deverão ser executadas ações para atender a prática de Verificação definida pelo CMM para a Gerência de Requisitos.

5. Estratégia x CMM

A tabela 2 descreve as práticas-chave para Gerência de Requisitos do CMM relacionando em que parte desta proposta as principais ações a serem executadas são definidas.

Práticas do CMM		Estratégia
Co	1 O projeto segue uma política organizacional para gerenciar os requisitos de software.	Fase 6 - Definição dos Processos
Hb	1 Para cada projeto, está estabelecida a responsabilidade para analisar os requisitos de sistema e alocá-los ao hardware, software ou outros componentes do sistema.	Fase 7 - Definição de Papéis e Responsabilidades
	2 Os requisitos de software estão documentados	Processo de Definição
	3 Estão disponíveis os recursos e fundos necessários para gerenciar os requisitos de software.	Fase 1 - Conscientização Fase 2 - Preparação Processo de Definição Processo de Elicitação de Requisitos
	4 Os membros do grupo de engenharia de software e de outros grupos relacionados a software estão treinados para realizar suas atividades de gerência de requisitos.	Fase 3 - Treinamento
At	1 O grupo de engenharia de software revisa os requisitos de software, antes de serem incorporados ao projeto.	Processo de Gerenciamento & Métricas Processo de Documentação & Análise Processo de Negociação & Priorização Processo de Validação Processo de Verificação
	2 O grupo de engenharia de software utiliza os requisitos de software como base para confeccionar os planos de desenvolvimento do software, desenvolver produtos de trabalho e realizar atividades.	Processo de Verificação
	3 Revisar as alterações nos requisitos de software e incorporá-las ao projeto de software.	Processo de Gerenciamento & Métricas
Ve	1 A gerência sênior revisa, periodicamente, todas as atividades de gerência dos requisitos de software.	Fase 9 - Acompanhamento e Ajustes
	2 O gerente de projeto revisa periodicamente ou por evento, todas as atividades de gerência dos requisitos de software.	Fase 9 - Acompanhamento e Ajustes
	3 O grupo de garantia de qualidade de <i>software</i> revisa e/ou audita as atividades e produtos de trabalho utilizados para gerenciar os requisitos de software, reportando seus resultados.	Processo de Validação

Tabela 2. Práticas-chave e estratégia

6. Estudo de Caso

Tendo definido as diretrizes para a implantação da gerência de requisitos, um projeto-piloto foi conduzido para validar a estratégia estabelecida. A decisão sobre qual sistema adotar foi tomada com base num perfil específico. O sistema deveria atender às seguintes características:

- a. possuir um cliente disposto a ser o pioneiro no processo de implantação da gerência e requisitos;

- b. possuir uma técnica devidamente treinada ou disponível para treinamento nos conceitos de engenharia de requisitos, em especial sobre processos para a melhoria da qualidade;
- c. possuir um escopo mínimo definido e controlável;
- d. ser de importância para o contexto das organizações participantes (SERPRO e Cliente);
- e. ser de fácil inserção no contexto da gerência de requisitos do SERPRO.

O sistema escolhido foi o SAD (Sistema de Apoio à Decisão), um sistema de apoio à decisão, possibilitando aos usuários de alto escalão do Cliente realizar consultas analíticas sobre uma base de dados que agrega, para várias áreas de atuação do usuário, inúmeras informações sobre as suas atividades. Dando suporte ao projeto, um *data warehouse* congrega dados provenientes dos principais sistemas do cliente, obtidas por meio de um complexo processo de extração, agregação e carga. Uma interface Web provê a acessibilidade às consultas para usuários habilitados em todo país. Isto é possível por meio das funcionalidades OLAP (*On-line Analytical Process*) [35]. Dentre estas funcionalidades, o SAD possibilita ao usuário trabalhar a informação, manipulando a sua apresentação para permitir análises de diversas maneiras, segundo diferentes critérios.

É um projeto de grande porte, envolvendo o processamento de 21 bases de dados de assuntos e extração distintas, o que implica na especificação de requisitos para 21 equipes de desenvolvimento. Vale ressaltar que as equipes estão localizadas nas diversas regiões do país.

Tendo escolhido o projeto piloto, foi inicializado o processo para a implantação da gerência de requisitos. Foram utilizadas ferramentas automatizadas para gerenciamento de mudanças, gerenciamento de requisitos, gerenciamento de configuração e testes.

Em processos de desenvolvimentos iterativos como os aplicados no SERPRO, uma das chaves para a qualidade do produto final é o acompanhamento eficiente dos requisitos, como forma de garantir que todos os requisitos fluam corretamente entre todos os envolvidos, sem que requisito algum seja perdido, mal interpretado ou simplesmente adicionado sem a anuência do cliente.

Durante o processo de *Definição*, foram utilizadas como técnicas de elicitação prototipação e entrevistas ao usuário. A prototipação auxiliou o usuário na descoberta de requisitos não imaginados previamente. Durante as entrevistas, foram elaborados os documentos *Glossário* e *Visão*, cuja importância foi bastante evidenciada tanto pelos desenvolvedores como pelo cliente.

No passo seguinte de *Análise & Documentação*, os requisitos foram documentados utilizando os documentos *Guia de Classificação de Requisitos*, *Especificação de Casos de Uso* e *Especificação Suplementar*. Estes dois últimos juntos compõem o DRS (Documento de Requisitos de Software do sistema). A escrita dos requisitos utilizou *Use Cases* para requisitos funcionais e descrições textuais para os requisitos não-funcionais e restrições.

Uma vez definidos e instanciados os documentos, os processos de *Negociação & Priorização* se seguiram, onde o Analista de Requisitos apresentou os documentos de requisitos por meio de um *Workshop* envolvendo o cliente com o intuito de validar o

seu conteúdo, identificando possíveis erros e priorizando os requisitos os requisitos dentro da seqüência de iterações, além da definição dos *milestones* e pontos de revisões.

Durante todo o processo de desenvolvimento, o processo de *Gerenciamento & Métricas*, por meio das *Solicitações de Alterações de Requisitos (SARs)*, o usuário pôde formalizar as alterações necessárias nos requisitos à equipe de projeto, as quais eram recebidas por meio da ferramenta de correio eletrônico da Empresa para que o Analista de Requisitos, utilizando a Matriz de Rastreamento, pudesse gerar o *Relatório de Impacto de Requisitos* e negociar então as alterações. Após a aprovação das alterações, todos os envolvidos eram notificados formalmente (por meio de mensagem de correio eletrônico), os documentos, alterados e uma nova *baseline* gerada.

Por fim, durante a execução dos processos de *Validação e Verificação*, coube ao grupo de garantia de qualidade de software verificar a conformidade dos documentos com os padrões definidos. *Checklists* de validação fornecem a base para esta atividade.

O estudo de caso completo encontra-se disponível em [36].

7. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma estratégia para a implantação de uma gerência de requisitos baseada em três pilares básicos: qualidade, requisitos e processos. Para que a qualidade seja alcançada é primordial que os requisitos tenham sido bem definidos e controlados e, para isto, devem haver processos estabelecidos e implantados.

Gerenciamento de requisitos é reconhecido como um importante pré-requisito para desenvolver software de alta qualidade e definidos como a habilidade de descrever e seguir a vida de um requisito, em ambas as direções.

As dificuldades mencionadas pela equipe de engenharia de requisitos foram a mudança da cultura para desenvolvimento de *software*, a definição de requisitos utilizando um paradigma novo para a equipe, as características peculiares dos requisitos para um sistema de *data warehouse* e a grande quantidade de documentos gerados.

Entretanto, foi ressaltada a importância da utilização dos processos definidos, que direcionaram as atividades e utilização dos padrões e da integração com outros grupos, como o de garantia de qualidade de software e o de gerência de configuração, que responsabilizou-se pelo versionamento e pelo relacionamento entre os documentos.

Como trabalhos futuros, deve-se ter o acompanhamento da execução dos processos em toda a empresa com a extração de métricas para que o processo evolua de acordo com as mudanças organizacionais, visando a melhoria contínua.

Bibliografia

- [1] Sommerville, I., "Software Engineering", 6th Edition, Addison-Wesley, 2001.

- [2] SEI-Software Engineering Institute, "Capability Model for Software", 1^a ed. USA: Carnegie Mellon University, Pittsburgh, Pennsylvania, 1997.
- [3] Zanolenci, E. P., "Descrição e Qualificação de Requisitos: Um Modelo Aplicável à Análise e Validação da Informação", Tese de Mestrado, PUC-PR, 1999.
- [4] Pressman, R., "Software Engineering: A Practioner's Approach", McGraw-Hill, 2000.
- [5] Zahran, S., "Software Process Improvement", Addison Wesley, 1998.
- [6] Paulk M. C. et all., "The Capability Maturity Model", version 1.1 (CMU/SEI-93-TR-24), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA(USA),1993.
- [7] Fiorini, S. T., Staa A., Baptista R. M. , "Engenharia de Software com CMM", Brasport, 1998.
- [8] Leite, J., Castro, J., Pinheiro, F., "Plataforma Tecnológica em Engenharia de Requisitos – Estratégias para o Aumento da Qualidade no Desenvolvimento de Sistemas", <http://www.cic.unb.br/~facp/per/perhome.html>
- [9] Booch, G., Rumbaugh, J E, Jacobson, I., "The Unified Modeling Language User Guide", Addison Wesley, 1999.
- [10] Booch, G., "Object Oriented Design with Applications", The Benjamin/Cumming Publishing Company, Inc., Redwood City, 1991.
- [11] Gil, G., Figueroa, D., Oliveros, A., "Producción Del LEL em um Dominio Técnico. Informe de um caso", III Workshop de Engenharia de Requisitos. Rio de Janeiro-Brasil, 2000.
- [12] Leite, J. "Applications Language: A product of Requirements Analysis", Departamento de Informática, PUC-RJ, 1989.
- [13] Nuseibeh, B. E, Easterbrook, S., "Requirements Engineering: A Roadmap", Proceedings of the 22nd International Conference on Software Engineering. Limerick, Ireland. Jun. 2000.
- [15] Gause, D. E, Weinberg, G.M., "Exploring Requirements, Quality before Design", 1989.
- [16] Damian, A., Hong, D. E, Li, H., "Joint Application Development and Participatory Design", disponível em www.cpsc.ucalgary.ca/~pan/seng/613/report.html
- [17] Couger, D., "Creative and innovation in information systems", Lomdon: International Thomson, 1995.
- [18] Schneider, G. E, Winters, J., "Applying Use Cases: a practical guide", Addison-Wesley. 1998.
- [19] Lamsweerde, A., Darimont, R. E., Letier, E., "Managing Conflicts in Goal-Driven Requirements Engineering", IEEE Transactions on Software Engineering, Special Issue on Managing Inconsistency in Software Development. Nov. 1998.
- [20] Yu, E., "Modeling Strategic Relationship for Process Reengineering", PhD thesis, Computer Science Department, University of Toronto, Toronto, Canada, 1995.

- [21] Maiden, N., "CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements", Automated Software Engineering, 1998.
- [22] Shaw, M. E, Gaines, B., "Requirements Acquisition", Software Engineering Journal, vol 11, 1995.
- [23] Viller, S. E., Sommerville, I., "Social Analysis in the Requirements Engineering Process: from ethnography to method", 4th International Symposium on Requirements Engineering, Limerick, Ireland, Jun, 1999.
- [24] Gogen, J. E, Jirotko, M. "Requirement Engineering", Boston, Mass.: Academic Press, 1994.
- [25] Davis, A. M., "Achieving Quality in Software Requirements" SQP 1, NO. 3. 1999.
- [26] Ambriola, V. E., Gervasi, V., "Processing Natural Language Requirements", 12th International Conference on Automated Software Engineering. Lake Tahoe, EUA. Nov. 1997.
- [27] Antoniou, G., "The Role of Non-monotonic Representations in Requirements Engineering", International Journal of Software Engineering and Knowledge Engineering, vol 8. 1998.
- [28] Gotel, O. E., Finkelstein, A., "An Analysis of the Requirements Traceability Problem", 1st International Conference on Requirements Engineering, Colorado Springs, EUA. Apr. 1994.
- [29] Wiegers K. E. , "When Telepathy Won't Do: Requirements Engineering Key Practices", Cutter IT Journal, May, 2000.
- [31] Karlsson, K. E., Ryan, K., "Software Requirements Prioritizing", 2nd International Conference on Requirements Engineering. Apr. 1996. pp. 110 – 116.
- [32] Saaty, T., "The Analytic Hierarchy Process", New York: McGraw-Hill, 1990.
- [33] Hauser, J. E., Clausing, D., "The House of Quality", The Harvard Business Review, vol 3. 1988.
- [35] Codd, E. F., Codd, F. B., Salley, C. T., "Providing OLAP (Online Analytical Processing) to User Analyst: an IT Mandate", Arbor Software White Paper, <http://www.arborsoft.com/OLAP.htm>, 1993.
- [36] Carvalho, A. E., "Uma Estratégia para Implantação de uma Gerência de Requisitos visando a Melhoria dos Processos de Software", Dissertação de Mestrado, UFPE, Brasil, 2001.