

# **Avaliação do processo de Gerenciamento e Engenharia de Requisitos em MPEs de Sistemas Embarcados: um estudo de caso**

Daniele Erica Damke  
UCB

[danidamke@yahoo.com.br](mailto:danidamke@yahoo.com.br)  
[claudia.melo.prof@gmail.com](mailto:claudia.melo.prof@gmail.com)

Patrícia Freitas de Moraes  
UCB

[pfmoraes@yahoo.com.br](mailto:pfmoraes@yahoo.com.br)

Claudia de Oliveira Melo  
UCB

## **RESUMO:**

*Aplicações de software embarcado estão cada dia mais presentes no cotidiano. No entanto, as técnicas de engenharia de software atuais não atendem às necessidades demandadas por este tipo de sistemas. Neste contexto, o presente artigo se propõe a avaliar o processo de engenharia de requisitos utilizados no desenvolvimento de Sistemas Embarcados por micro e pequenas empresas, tendo como objetivo sugerir um processo que se adapte de maneira adequada a este domínio de software.*

Palavras-Chave: Sistemas Embarcados. Engenharia de Requisitos, Micro e Pequenas Empresas.

## **1. Introdução**

Com a rápida evolução dos microprocessadores e a diminuição dos custos de produção, os Sistemas Embarcados tornaram-se universais, já que estão presentes em muitos dispositivos, desde os mais simples, como eletrodomésticos, até dispositivos complexos como naves espaciais. Entretanto, a maioria desses sistemas não é percebida pelas pessoas, pois como o próprio nome insinua, estão embutidos dentro de outros sistemas utilizados no dia-a-dia BARRETO (2006).

Os Sistemas Embarcados (SEs) são projetados sob restrições de recursos tais como energia consumida, utilização de memória, tamanho do circuito integrado, desempenho, resposta em tempo real, e, principalmente, custo BARBIERO (2006). À medida que aumenta a quantidade e variedade de SEs, há um acréscimo em sua complexidade, devido à integração de mais componentes de software e hardware ao sistema. A união de restrições e complexidade torna o projeto de Sistemas Embarcados um desafio constante, visto que todas estas características devem ser levadas em consideração durante o seu desenvolvimento.

Segundo CARRO E WAGNER (2002), os fatores que contribuem para o aumento do número de projetos fracassados em Sistemas Embarcados são a mudança na especificação, especificação incompleta, complexidade da aplicação, quantidade insuficiente de desenvolvedores e tecnologias específicas para a aplicação de softwares embarcados. Em parte porque as técnicas de Engenharia de Software existentes são recentes, em outra, porque não levam em consideração as características específicas de Sistemas Embarcados. As tecnologias de desenvolvimento existentes não avaliam adequadamente o impacto em Sistemas Embarcados e nem são flexíveis a ponto de servirem a tais propósitos GRAAF (2003).

O mercado mundial de software, estimado em mais de 21 bilhões de dólares, cresce rapidamente TAURION (2005). A indústria de SEs representa uma das melhores oportunidades para economias emergentes como o Brasil, pois está presente em diversas cadeias produtivas. No cenário nacional, 70% do setor de informática brasileiro é formado por micro e pequenas empresas (MPEs). Segundo MCT (2002), as MPEs são responsáveis pela

maioria das empresas e postos de trabalho em todo o mundo e, no Brasil, este segmento aumenta a cada ano com os movimentos de terceirização e de avanço técnico. Sendo assim, grande parte das empresas desenvolvedoras de software embarcado no Brasil são MPEs.

Neste contexto, este trabalho se propõe a identificar, entender e avaliar o processo de Engenharia de Requisitos de uma MPE desenvolvedora de software embarcado, a fim de elaborar um processo que melhor se adapte às necessidades deste domínio de software e de empresa.

Este artigo está organizado da seguinte maneira: na seção 2 são descritas as características dos Sistemas Embarcados. A seção 3 apresenta conceitos relacionados a processos de engenharia de requisitos tradicionais e específicos para Sistemas Embarcados. As seções 4 e 5 discorrerão sobre a avaliação e a modelagem do processo de engenharia de requisitos em micro e pequenas empresas desenvolvedoras de softwares embarcados. E, finalmente, a seção 6 corresponde às considerações finais do artigo, bem como trabalhos futuros.

## **2. Sistemas Embarcados**

Segundo MARWEDEL (2003), Sistemas Embarcados podem ser definidos como sistemas eletrônicos de processamento de informação embutidos em um produto de forma transparente para o usuário. Este tipo de software é composto de sistemas microprocessados que não aparecem como funcionalidade final de um produto, ou seja, estão completamente encapsulados ou dedicados ao dispositivo ou sistema que ele controla, portanto, são invisíveis do ponto de vista do usuário WOLF (2001).

Um estudo publicado por LEE (2002) ressalta que os Sistemas Embarcados não se resumem a software em um pequeno dispositivo. Ao contrário da computação tradicional, sua principal característica não é a transformação de dados, mas preferencialmente a interação com o mundo real. Enquanto, na computação tradicional, o objetivo principal é atender a funcionalidades com desempenho cada vez maior. Na computação embarcada o interesse está voltado para interfaces com o ambiente, de modo a satisfazer severas restrições, como, requisitos temporais, de consumo de energia e memória, mobilidade, tamanho, peso, segurança, confiabilidade, tempo de entrega dentre outras BARRETO (2006).

MARWEDEL (2003) afirma que para ser considerado “embarcado” o sistema deve apresentar algumas características, dentre elas: estar acoplado ao ambiente físico; ser confiável e eficiente (em termos de energia e memória consumida, tamanho do código, tempo de execução, peso e custo); ser dedicado à aplicação; possuir interface com o usuário dedicada (botões de pressão, volantes, pedais, etc); apresentar restrições de tempo real; ser híbrido (composto por componentes analógicos e digitais) e reativo (estar em interação contínua com seu ambiente, executando em um ritmo determinado por este ambiente).

Apesar de se mostrar um assunto extremamente interessante e com grandes possibilidades de estudo, os Sistemas Embarcados são pouco representados academicamente e em discussões públicas MARWEDEL (2003). Do ponto de vista do software, eles eram muito pequenos e retrógrados, conseqüentemente, o projeto de Sistemas Embarcados não foi beneficiado pela valiosa abstração do desenvolvimento do século XX. As melhores tecnologias com seu uso desregrado de memória, camadas de abstração, algoritmos elaborados e otimizações estatísticas não se mostravam aplicáveis ao contexto dos Sistemas Embarcados. Além disso, como os resultados das pesquisas não resolviam os problemas dos Sistemas Embarcados, o problema não era, portanto, interessante aos olhos dos pesquisadores LEE (2000).

No entanto, isto mudou, devido a inúmeras razões os pesquisadores estão começando a reconstruir suas pesquisas de maneira que possam adequá-las aos problemas reais e diferentes apresentados pelos Sistemas Embarcados. A maior dificuldade é justamente, adaptar as técnicas de software existentes para enfrentar os desafios do mundo físico LEE (2000).

Os desafios da pesquisa são enormes, uma vez que os prazos de entrega dos sistemas é cada vez menor, o custo e a complexidade desse tipo de sistemas é extremamente alta, não sendo confiável a utilização de metodologias de desenvolvimento com soluções *ad hoc*. Contudo, as tecnologias de engenharia de software atuais ainda não possuem flexibilidade suficiente para lidar com as características específicas dos SEs.

### 3. Engenharia de Requisitos

Entender os requisitos de um problema está entre as tarefas mais difíceis enfrentadas por um Engenheiro de Software PRESSMAN (2006). Por mais bem definido e otimizado que seja o sistema, entregar ao cliente o que ele deseja e, realmente, precisa, exige esforço, atenção e experiência por parte da equipe de projeto.

O ponto de partida na concepção de sistemas é a definição de seus requisitos funcionais e não-funcionais. No entanto, pesquisas realizadas com centenas de organizações revelaram que a Engenharia de Requisitos é deficiente em mais de 75% das empresas HOFMANN (2001). O trabalho de GASTALDO (2003) demonstrou que cerca de 50% do retrabalho referente ao processo de desenvolvimento de software ocorre nas fases iniciais de elicitação, análise e documentação de requisitos.

Diante deste contexto, a Engenharia de Requisitos tem se tornado cada vez mais necessária para resolver os problemas encontrados nas organizações com relação à definição de sistemas. Elaborar um processo que preencha as lacunas existentes nas fases de elicitação, análise, documentação e validação de requisitos, especialmente os não-funcionais, é uma tarefa de grande importância para o sucesso dos projetos de Sistemas Embarcados.

#### REQUISITOS EM SISTEMAS EMBARCADOS

Um dos maiores desafios no desenvolvimento de Sistemas Embarcados é o atendimento de seus requisitos não-funcionais. SEs devem obedecer restrições severas no que diz respeito a requisitos temporais, de peso, tamanho, mobilidade, segurança, confiabilidade, consumo de energia e memória. Estes requisitos colocam restrições sobre o produto que está sendo desenvolvido, sobre o processo de desenvolvimento, e também, especificam restrições externas ao produto.

De uma maneira geral, requisitos não-funcionais são descobertos ou tratados tardiamente nos projetos. Um dos motivos é o fato de não serem cobertos pela maioria dos métodos de Engenharia de Requisitos SOMMERVILLE (1998). Além disso, algumas restrições são descobertas somente na fase de projeto; certas restrições são altamente subjetivas, e portanto, determinadas, apenas, por avaliações empíricas; os requisitos não-funcionais tendem a estar relacionados com mais de um requisito funcional, dificultando a explicitação desta dependência; os requisitos não-funcionais tendem a conflitar e contradizer uns aos outros; não existem regras para expressar os requisitos não-funcionais.

Agregado a todas as características já citadas, existe a demanda do mercado, por produtos de baixo custo e curto prazo de desenvolvimento YAMAURA ET AL (2003). Ou seja, estes softwares, além de possuírem características extremamente complexas, devem garantir uma rápida entrega e produtos de altíssima qualidade LEE ET AL (2007).

## PROCESSO DE ENGENHARIA DE REQUISITOS PARA SISTEMAS EMBARCADOS

A maioria das companhias que desenvolve softwares embarcados não os vende. Eles vendem, na realidade, telefones celulares, CD players e outros produtos. O software nestes produtos constitui apenas uma parte, certamente importante, do produto como um todo. Consequentemente, grande parte das empresas possui um desenvolvimento dirigido ao hardware, onde o software começa a ser desenvolvido depois que o hardware já está em um estágio avançado de desenvolvimento GRAAF (2003).

Segundo SOMMERVILLE (1998), poucas organizações têm um processo de requisitos explicitamente definido e padronizado. Processos de engenharia de requisitos específicos para Sistemas Embarcados são ainda mais difíceis de encontrar. De acordo com GRAAF (2003), o modelo mais comum de processo de desenvolvimento de sistemas embarcados utiliza uma abordagem *top-down*. Ou seja, nestes modelos o refinamento de requisitos e arquitetura é realizado em várias iterações subsequentes, partindo do nível de sistemas até chegar ao nível de componente. A Figura 1 apresenta este modelo.

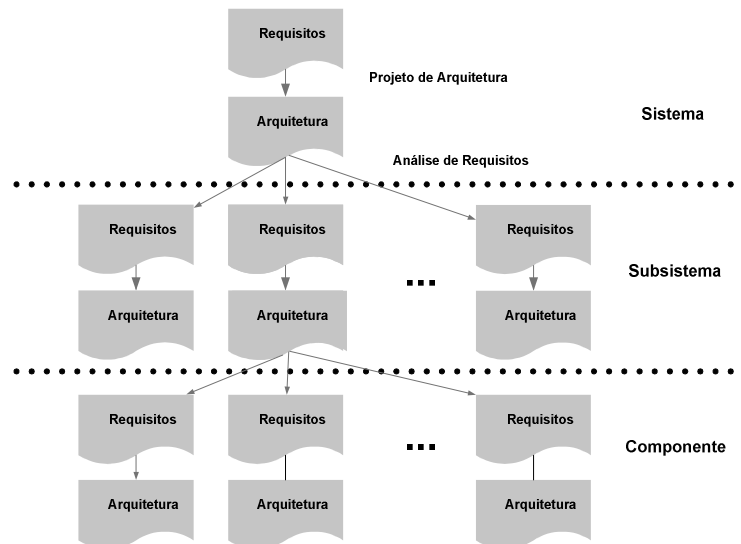


Figura 1 - Decomposição do processo de desenvolvimento de sistemas embarcados (Graaf 2003)

Este processo genérico não se distancia muito do elaborado por LATTEMANN (1997), representado na Figura 2. Em ambos o sistema é dividido em sistemas menores e a especificação é refinada em etapas. LATTEMANN (1997) defende que os requisitos de todo o sistema devem ser descritos e só depois os do controlador, elaborando as interfaces entre hardware e software.

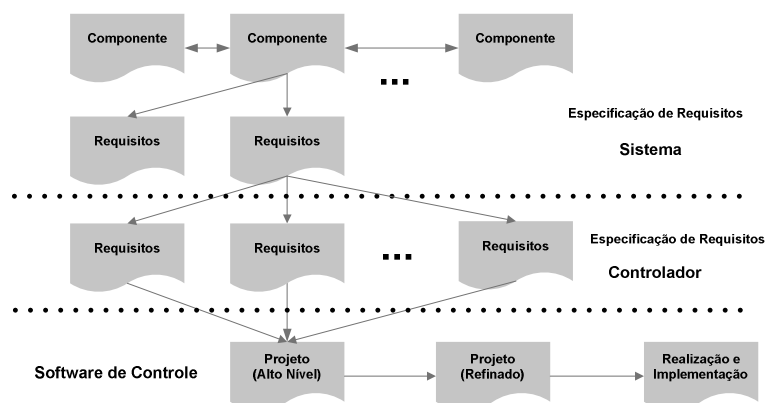


Figura 2 - Processo de Requisitos (Lattemann 1997)

O processo de requisitos utilizado por NARS (2002) toma como base a técnica de caso de uso e possui 7 etapas principais, conforme ilustrado na Figura 3. Segundo esta abordagem,

o primeiro passo consiste na definição do sistema, seguido da especificação dos requisitos de qualidade, onde serão explicitamente definidas as restrições do sistema. Daqui em diante o processo de NARS (2002) muito se assemelha aos processos tradicionais de ER: são encontrados os atores, os casos de uso, é construído o modelo de caso de uso e são elaborados os documentos do sistema.

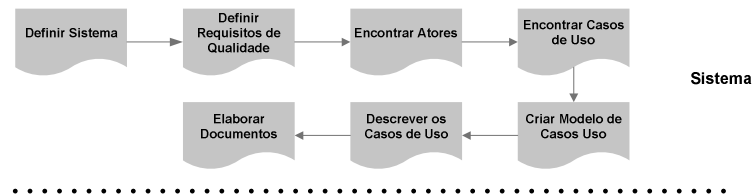


Figura 3 - Processo de Requisitos (Nars 2002)

Diferentemente dos processos elaborados por GRAAF (2003) e LATTEMANN (1997), o processo de requisitos utilizado por NARS (2002) não divide a especificação em subsistemas ou separa as especificações de hardware e software. NARS (2002) defende a especificação do sistema como todo, dada a difícil separação de hardware e software no contexto de SEs. Como ponto positivo desta abordagem pode-se destacar a importância dada à especificação dos requisitos de qualidade. Identificar, explicitamente, os requisitos não funcionais do sistema é de fundamental importância no domínio de SEs.

No que diz respeito às metodologias adotadas no desenvolvimento de softwares embarcados, de acordo com GRAAF (2003) a UML ainda não é uma prática comum, mas muitas empresas já têm considerado sua possibilidade de aplicação na ER PUSCHNIG (2004) defende a utilização da técnica de caso de uso, pois de acordo com suas pesquisas, a modelagem por casos de uso foi de grande valia para o entendimento inicial do sistema. NARS (2002) por sua vez, ressalta que a aplicação desta técnica em um caso real da indústria de sistemas embarcados revelou que os casos de uso são carentes de definições apropriadas para a adaptação a SEs.

Segundo GRAAF (2003), o produto da primeira fase de Engenharia de Requisitos em todas as empresas era o acordo de especificação de requisitos, uma descrição do sistema que todos os stakeholders pudessem compreender. Vale destacar que um estudo realizado por PUSCHNIG (2004) identificou que, para o sucesso da Engenharia de Requisitos, além do núcleo da equipe de desenvolvimento formada por engenheiros de requisitos, analista de processo, programadores, etc., é necessário o envolvimento de especialistas de diferentes domínios. Isto se faz necessário em virtude do pouco conhecimento que os stakeholders possuem acerca do sistema a ser desenvolvido.

GRAAF (2003) ressalta, ainda, que as companhias normalmente especificam seus requisitos em linguagem natural, registrando as informações em processadores de texto. Com frequência utilizam modelos para estruturar os documentos, no entanto, nem todos os projetos de uma mesma companhia os utilizavam. Ainda em relação à documentação dos requisitos, GRAAF (2003) afirmou que apesar da importância dos requisitos funcionais no contexto de SEs, de maneira geral, requisitos como consumo de energia e memória não são explicitamente especificados e projetados.

Outro complicador destacado por GRAAF (2003) é o fato de muitas companhias reutilizarem as especificações de requisitos de projetos anteriores, mesmo para o desenvolvimento de uma nova linha de produto. Isto dificulta a tarefa de deixar todos os produtos e documentos do desenvolvimento consistentes, visto que é preciso analisar precisamente o impacto de novas características. No entanto, os projetos estudados frequentemente não explicitavam as relações entre requisitos, então a análise de impactos era

muito difícil. Rastrear os requisitos era difícil por que as relações (por exemplo, entre requisitos e componentes arquiteturais) eram muito complexas para serem especificadas manualmente e as ferramentas de gerenciamento de requisitos disponíveis não pareciam resolver este problema, apesar de algumas versões customizadas funcionarem em alguns casos.

Esta rastreabilidade é um aspecto essencial do gerenciamento de requisitos, visto que entender os riscos associados ao impacto do software em seu ambiente é fundamental para assegurar a dependabilidade e a segurança no funcionamento de Sistemas Embarcados. Infelizmente, a análise de risco de Sistemas Embarcados tem sido negligenciada e apenas recentemente tem recebido atenção da comunidade de engenharia de software HEWETT (2005).

Embora processos tenham sido propostos, eles não possuem o detalhamento suficiente de atividades, tarefas e métodos para auxiliar o levantamento e especificação de requisitos de um SE. Além disso, eles mostraram-se muitas vezes inadequados para o contexto deste tipo de sistema. Isso contribui para que as empresas não consigam institucionalizar um processo de desenvolvimento que ajude a reduzir as falhas em projetos. Neste contexto, fica clara a necessidade da elaboração de um processo que melhor se adapte ao domínio de SEs, objetivo final deste estudo.

#### 4. Avaliação do Processo de Engenharia de Requisitos em Micro e Pequenas Empresas Desenvolvedoras de SEs

Para propor um processo de Engenharia de Requisitos mais aderente à realidade das empresas desenvolvedoras de SEs, particularmente às MPes, foi desenvolvido um modelo de avaliação baseado na abordagem Goal Question Metric (GQM).

O principal objetivo da metodologia GQM é fornecer e caracterizar um melhor entendimento dos processos, produtos, recursos e ambiente para estabelecer bases para comparações com trabalhos futuros ou até mesmo com metas (índices) de mercado ANQUETIL (2005). Portanto, podemos defini-la como um mecanismo utilizado para planejar e definir objetivos da medição e avaliar produtos e processos de software, pois é uma abordagem de mensuração que é orientada a objetivos GLADCHEFF (2001).

De acordo com o GQM, para que se possa medir de maneira eficaz é necessário, em primeiro lugar, traçar os objetivos da medição a fim de que sirvam como um guia para a criação das questões e, posteriormente, das métricas. O modelo GQM parte do princípio que a implementação de metas operacionais e mensuráveis para melhoria de software deve seguir uma abordagem *top-down*, no entanto a interpretação dos dados coletados durante a avaliação deve seguir a abordagem *bottom-up* WANGENHEIM (1999). O modelo GQM é dividido em quatro fases, conforme ilustrado na Figura 4.

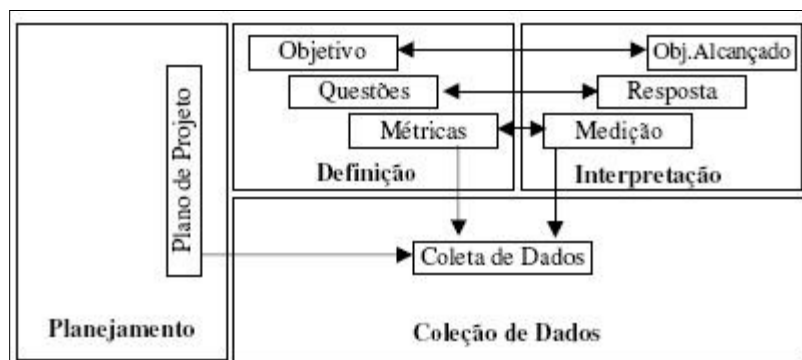


Figura 4 - Processo de Requisitos (Lattemann 1997)

Sendo assim, tomando-se como base a metodologia GQM, o modelo de avaliação proposto foi elaborado em 4 (quatro) etapas – planejamento, definição, coleta e interpretação.

#### PLANEJAMENTO

Nesta fase foram definidos o objetivo, o foco e a equipe participante da avaliação. O objetivo principal da avaliação foi o de analisar as dificuldades encontradas no processo de engenharia de requisitos de SEs. Com relação ao foco, as MPEs foram selecionadas como público alvo da avaliação. Quanto aos participantes, a equipe responsável por definir e coordenar a avaliação é formada pelas pesquisadoras relacionadas ao trabalho. Já a equipe avaliada é formada por funcionários de uma pequena empresa com foco comercial em desenvolvimento de sistemas agrários.

#### DEFINIÇÃO

Nesta fase o objetivo de medição foi formalizado, conforme descrito na Tabela 1.

Tabela 1 - Objetivo do Negócio

|  |
|--|
| Adequar o processo de gerenciamento de requisitos de micro e pequenas empresas desenvolvedoras de sistemas embarcados, levando em consideração características específicas deste tipo de software. |
|--|

Também foram elaboradas as questões, subquestões e métricas relacionadas ao objetivo definido. Nesta tarefa, além das características e dificuldades específicas do domínio escolhido, foram levados em consideração os resultados esperados do processo de Gerência de Requisitos do nível G do programa de Melhoria de Processo do Software Brasileiro – MPS.BR SOFTEX(2007).

#### COLETA E INTERPRETAÇÃO

A coleta dos dados da pesquisa foi efetuada manualmente, por meio do preenchimento de planilhas Excel. Após o recebimento do questionário respondido pela equipe avaliada foi realizada uma análise para verificar a coerência entre as respostas.

#### INTERPRETAÇÃO DOS ÍNDICES CALCULADOS

A partir dos índices coletados em cada questão foram calculados os dados quantitativos por subquestão o que permitiu a interpretação das informações obtidas com a avaliação.

A empresa participante da avaliação é uma MPE com 6 funcionários que atua no desenvolvimento de hardware criptográfico há 3 anos. Os softwares embarcados desenvolvidos são dedicados a uma plataforma própria e compõem um produto da empresa. No entanto, possuem integrações não somente com sistemas da empresa, mas também com sistemas de outros fabricantes. Como o software desenvolvido é parte de um produto da empresa avaliada, os requisitos do software são levantamentos/criados internamente.

As métricas obtidas na subquestão que avalia a contextualização do software revelaram que seu nível de portabilidade não é alto, no entanto seu índice de integração com outros sistemas, inclusive de outros fabricantes, é crítico. Logo, o contexto no qual o software está inserido é de média criticidade.

Já os índices de contextualização da equipe de requisitos mostraram que não existe na empresa uma equipe específica para o levantamento de requisitos. No entanto, apesar do software ser apenas um componente de um produto, seu levantamento de requisitos é realizado por um processo a parte. Embora a formação dos responsáveis pelo levantamento seja em computação ou áreas afins, estes não possuem conhecimentos aprofundados sobre a Engenharia de Requisitos. Em compensação, possuem conhecimentos arraigados acerca do negócio para o qual o software está sendo desenvolvido.

A empresa apresentou um excelente nível de entendimento dos requisitos funcionais e não-funcionais. No entanto, o estudo revelou que o levantamento dos requisitos é realizado pelos arquitetos e projetistas do sistema, de maneira que a equipe de desenvolvimento recebe os requisitos prontos para implementação. E este levantamento é realizado sem a utilização de nenhum método, modelo ou linguagem específica.

Apesar de terem mostrado um bom índice na identificação e registro das necessidades, expectativas e restrições do cliente, a pesquisa expôs que o registro destas informações não é realizado em um documento específico para esse fim. No entanto, essas informações são acordadas e registradas em documentos estruturados, por exemplo, atas de reunião. Já o índice de aprovação dos requisitos propriamente ditos revelou que os mesmos são acordados e registrados formalmente em documentos como o de especificação de requisitos.

Com relação ao item que analisa o nível de refinamento dos requisitos funcionais, o índice encontrado revelou que as necessidades do cliente não são refinadas gradualmente. Dessa maneira, as necessidades do cliente são levantadas e refinadas no nível de requisitos sem a preocupação de identificar as características do sistema.

O cálculo dos dados quantitativos revelou que os requisitos não-funcionais são identificados a partir dos requisitos-funcionais, no entanto, não são mantidos ao longo do tempo. Além disso, são realizados somente estudos de viabilidade dos requisitos funcionais.

O índice que mede a dependência entre os requisitos mostrou que esta é avaliada somente entre os requisitos funcionais, desconsiderando os não-funcionais. Já a rastreabilidade dos requisitos apesar de ser elaborada, não é mantida ao longo do projeto.

No que diz respeito à correteza dos requisitos foi identificado que estes são testados pela equipe de desenvolvimento sem a utilização de roteiros de testes mas não são validados pelo cliente.

Já os dados quantitativos referentes às revisões dos produtos de trabalho denunciaram que a sua revisão e correção é realizada durante o processo de desenvolvimento do software mas não durante sua manutenção. Por outro lado, as mudanças dos requisitos funcionais e não-funcionais são mantidas ao longo do tempo. É importante ressaltar que essa manutenção dos requisitos não-funcionais não leva em consideração os requisitos funcionais relacionados. Vale destacar também que as manutenções mais frequentes na empresa consistem na geração de versões especializadas do produto que não substituem os requisitos originais.

Por fim, pode-se identificar, por meio dos valores calculados, que as interfaces do sistema são representadas com a utilização de atores, no entanto, isto é registrado em documentos sem uma estrutura específica para tal, como atas de reuniões.

## **5. Modelagem do Processo de Engenharia de Requisitos em Micro e Pequenas Empresas Desenvolvedoras de SEs**

Com base no estudo de caso apresentado na seção anterior, foi possível propor um modelo de processo de Gerenciamento e Engenharia de Requisitos para Micro e Pequenas empresas desenvolvedoras de software (ProReqSE).



Além da utilização dos resultados obtidos no estudo de caso, o ProReqSE utilizou como base para sua elaboração o Open UP, uma simplificação do RUP que se adapta à realidade de pequenas equipes. Da mesma maneira que o OpenUP, o ProReqSE é um processo elaborado para aplicação em equipes pequenas e co-localizadas que buscam uma abordagem simplificada para a especificação de seus produtos. Esta abordagem foi mantida devido ao curto prazo normalmente exigido para o desenvolvimento de sistemas embarcados e ao contexto das MPEs.

Outros dois pontos foram considerados na definição das principais características do ProReqSE: a informalidade dos processos de desenvolvimento das micro e pequenas empresas brasileiras e a formação multi - disciplinar das equipes de desenvolvimento de sistemas embarcados. Dessa forma, o ProReqSE foi modelado para ser personalizável - de maneira que as empresas possam adequá-lo às suas necessidades de acordo com sua cultura e nível de formalização; completo - permitindo que o processo de engenharia de requisitos seja efetivamente utilizado para atingir seus objetivos sem a necessidade de complementações e mínimo - de forma que seu foco é atingido com a utilização de poucos elementos.

O ProReqSE define um conjunto de atividades cujo objetivo é a compreensão, o refinamento, a documentação e a verificação das necessidades do cliente de forma a construir um software embarcado adequado. Para, isso foram definidas as seguintes atividades: definir visão, encontrar e resumir requisitos e detalhar requisitos.

Estas atividades possuem artefatos de entrada e saída os quais foram denominados produtos de trabalho. Além disso, elas devem ser executadas pelos papéis definidos no processo.

Embora os papéis do ProReqSE exijam conhecimento específico de Engenharia de Requisitos e grande parte das empresas avaliada não possuam equipes com este conhecimento, tal aspecto não será de grande relevância, visto que o ProReqSE foi elaborado de forma a apoiar todo o processo de engenharia de requisitos. Para isso foram definidos modelos de documentos (templates), diretrizes de trabalho, listas de verificações e conceitos específicos da engenharia de requisitos.

Um dos pontos a ser destacado em relação ao processo definido é a abrangência de sua aplicação. Apesar da dificuldade na separação dos requisitos de hardware e software no domínio de sistemas embarcados, o ProReqSE optou por manter a distinção entre estes dois componentes do sistema. Dessa maneira, o ProReqSE abordará o processo de engenharia de requisitos de sistemas embarcados visando a construção de seu software. Os motivos que levaram a esta decisão foram: a falta de conhecimento, por parte da equipe desenvolvedora do processo, em relação à engenharia de hardware; e as diferentes necessidades demandadas pelos dois processos de desenvolvimento – dificilmente uma técnica ou modelo para levantamento e especificação de requisitos de software seria aplicável ao hardware. No entanto, o ProReqSE não deixou de levar em consideração esta intrínseca ligação entre os dois componentes do sistema ao definir os elementos do processo.

## PAPÉIS

Os papéis definidos no ProReqSE levam em consideração as habilidades dos profissionais que devem assumi-los, além destas habilidades estarem diretamente relacionadas as atividades que cada papel executa. Os papéis definidos no ProReqSE são: Analista, Arquiteto, Desenvolvedor, Envolvidos e Gerente de Projetos.

## ATIVIDADES

O ProReqSE foi dividido em quatro atividades, onde cada uma delas possui uma seqüência de tarefas a serem seguidas para atingir um objetivo.

- **Definir Visão:** definir a visão para o futuro sistema, ou seja, descrever o problema a ser resolvido e as características do sistema de acordo com as solicitações dos interessados. Vale ressaltar que neste momento não há separação entre hardware e software. A visão deve ser elaborada para o sistema como um todo.

O fluxo de execução da atividade “Definir Visão”, os papéis envolvidos, os produtos de trabalho tanto de saída quanto de entrada, as listas de verificação, os conceitos e as diretrizes são ilustrados na Figura 5.

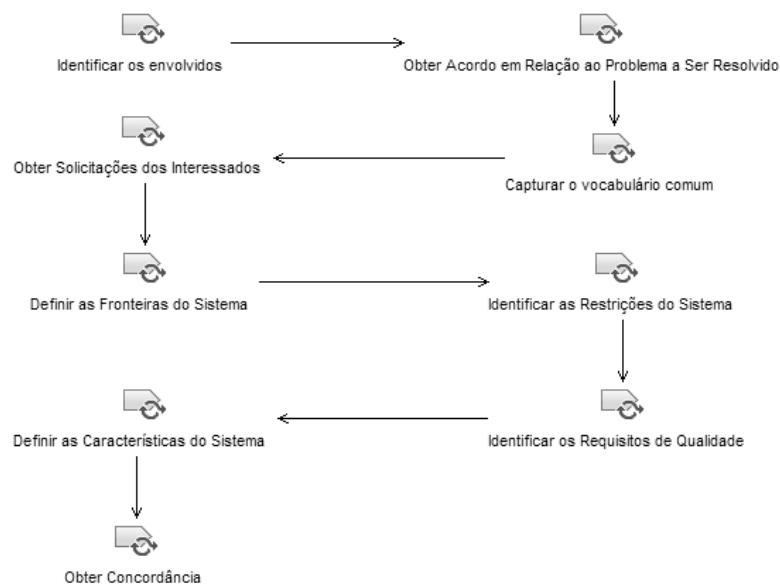


Figura 5 – Fluxo da Atividade Definir Visão

- **Encontrar e Resumir Requisitos:** O propósito desta atividade é identificar e capturar requisitos funcionais e não-funcionais para o sistema. Estes requisitos formam a base da comunicação e concordância, entre os interessados e a equipe de desenvolvimento, a respeito do que o sistema deve fazer para atender as necessidades dos envolvidos. O seu objetivo é entender os requisitos em um alto nível de abstração para que seja possível determinar o escopo do sistema. Esta atividade pode ser realizada dividindo-se o sistema em componentes menores e construindo-se conexões entre eles. Posteriormente serão realizadas análises para detalhar os requisitos prioritários para implementação.

As tarefas da atividade “Encontrar e Resumir Requisitos” são ilustradas na Figura 6:

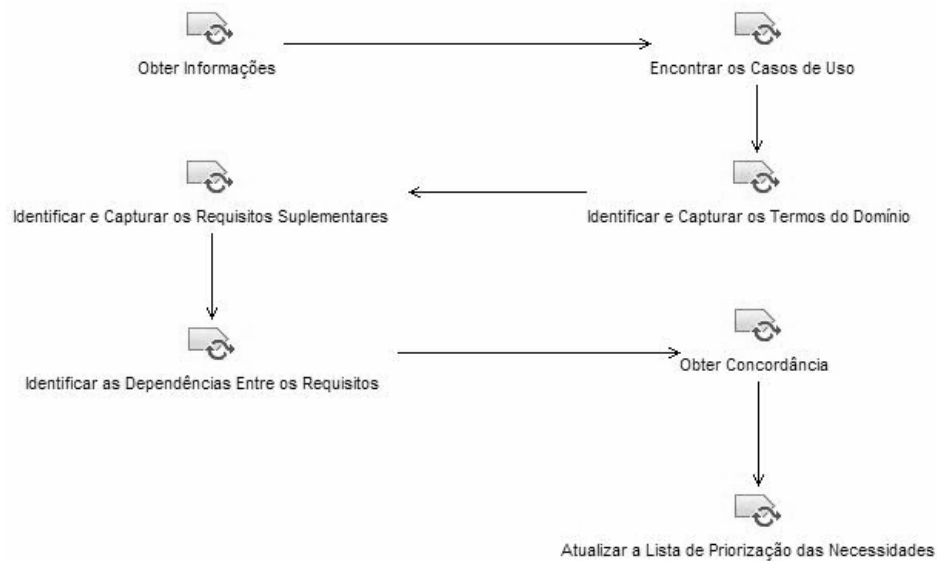


Figura 6 – Fluxo da Atividade Encontrar e Resumir Requisitos

- **Detalhar Requisitos:** Esta atividade visa descrever os requisitos em detalhes, de acordo com sua priorização, para permitir a iniciação do desenvolvimento.

O fluxo da atividade “Detalhar Requisitos” é ilustrado na Figura 7:



Figura 7 – Fluxo da Atividade Detalhar Requisitos

- **Revisar Requisitos:** Esta atividade visa revisar todos os produtos de trabalho elaborados nas atividades e tarefas anteriores com o objetivo de validar o entendimento dos requisitos e garantir sua concordância com as expectativas dos envolvidos.

As tarefas da atividade “Revisar Requisitos” são ilustrados na Figura 8:

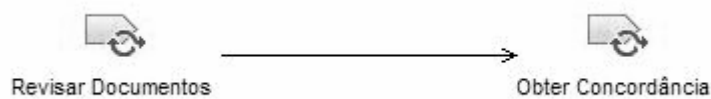


Figura 8 – Fluxo da Atividade Revisar Requisitos

## PRODUTOS DE TRABALHO

Produtos de Trabalho são informações produzidas, modificadas ou utilizadas pelas atividades do processo. Para facilitar o entendimento, a elaboração dos produtos de trabalho e a melhoria da qualidade da especificação foram definidos modelos (templates) destes documentos.

O ProReqSE sugere a utilização de poucos artefatos devido a seu direcionamento às MPEs e a cultura das equipes desenvolvedoras de software embarcado. No entanto, estes artefatos podem ser acrescidos, suprimidos ou customizados de acordo com a necessidade de cada projeto.

O ProReqSE fornece os seguintes produtos de trabalho:

- Documento de Visão: Este artefato fornece uma base de alto nível, às vezes contratual, para os requisitos técnicos mais detalhados que são visíveis para os envolvidos. Captura a essência do sistema descrevendo as restrições do projeto e os requisitos de alto nível que dão ao leitor uma visão geral do sistema de uma perspectiva dos requisitos comportamentais.
- Especificação de Caso de Uso: Este artefato captura a seqüência das ações executadas por um sistema que tenham um resultado de valor observável para aqueles que interagem com o sistema. O principal diferencial da especificação de caso de uso utilizada pelo ProReqSE é a definição de tipos de atores específicos do domínio de sistemas embarcados, descritos na tarefa Definir as Fronteiras do Sistema.
- Especificação Suplementar: Este artefato captura e detalha todos os requisitos do sistema que não são especificados nos casos de uso, incluindo requisitos de atributos de qualidade e requisitos funcionais globais. O maior diferencial da Especificação Suplementar do ProReqSE é a utilização da tabela de especificação da PLanguage.
- Glossário: O objetivo do glossário é fornecer um vocabulário comum acordado por todos os envolvidos. Ajuda pessoas de diferentes grupos funcionais a alcançar uma compreensão mútua do sistema. O objetivo não é registrar todos os termos possíveis, mas somente aqueles que não estão claros, são ambíguos ou pertencem a domínios específicos.
- Lista de Priorização das Necessidades: O objetivo deste artefato é priorizar as necessidades dos envolvidos a fim de guiar a seqüência do detalhamento, projeto e implementação do sistema.
- Modelo de Caso de Uso: Este artefato captura um modelo das funções desejadas do sistema e de seu ambiente, e serve como um contrato entre o cliente e os desenvolvedores.
- Documento de Referência Cruzada: A finalidade deste documento é manter a rastreabilidade e a concorrência (dependência, prejuízo ou colaboração) entre os elementos do projeto. O ProReqSE sugere que seja feita a rastreabilidade entre necessidades, características, requisitos funcionais, requisitos não-funcionais. Contudo, outros elementos do projeto podem ser rastreados.
- Diagrama de Seqüência de Cenários Operacionais: Este documento visa representar a seqüência temporal da execução de cada um dos cenários dos casos de uso. De maneira que a concorrência entre os fluxos possa ser tratada a fim de definir a correta seqüência a ser obedecida durante a execução. Por exemplo: identificar qual exceção deve ser priorizada caso duas exceções sejam lançadas no mesmo momento.

## DIRETRIZES

As diretrizes são guias que buscam facilitar a execução das tarefas que compõe as atividades definidas do ProReqSE e conseqüente a elaboração de seus produtos de trabalho. Estes itens são de vital importância para o processo dado o contexto das equipes desenvolvedoras de sistemas embarcados.

As diretrizes disponibilizadas no ProReqSE são: Armadilhas de Requisitos; Detalhar Casos de Uso; Encontrar Caso de Uso; Encontrar e Resumir Atores e Caso de Uso; Escrever Requisitos Adequadamente; Formato dos Casos de Uso; Modelo de Caso de Uso; Realizar a Revisão dos Requisitos; Requisitos Suplementares; Técnicas para Obtenção de Requisitos.

## LISTAS DE VERIFICAÇÃO

As listas de verificação visam garantir a qualidade e a completude dos produtos gerados em cada uma das atividades do processo de engenharia de requisitos para sistemas embarcados, bem como, a alta satisfação do cliente. O ProReqSE utiliza as Listas de Verificação definidas a seguir na atividade Revisar Requisitos: Lista de Verificação de Ator; Lista de Verificação de Bons Requisitos; Lista de Verificação de Casos de Uso; Lista de Verificação de Modelos de Caso de Uso; Lista de Verificação de Requisitos Suplementares; Lista de Verificação de Visão;

## CONCEITOS

O objetivo dos conceitos é definir de forma detalhada termos e técnicas freqüentemente utilizados na Engenharia de Requisitos. Da mesma forma que as diretrizes, este elemento do processo auxilia as equipes de desenvolvimento de sistemas embarcados, uma vez que estas não possuem conhecimentos específicos de Engenharia de Requisitos.

Os conceitos fornecidos pelo ProReqSE são: Ator; Atributos de Requisitos; PLanguage; Característica; Caso de Uso; Modelo de Caso de Uso; Rastreabilidade; Requisitos; Requisitos Arquiteturalmente Significantes; Requisitos Suplementares.

## 6. Considerações Finais

Com base na evolução da computação embarcada, nas dificuldades de desenvolvimento deste tipo de sistemas e no crescente aumento da MPEs, o presente trabalho apresentou um modelo processo de engenharia de requisitos para sistemas embarcados, o ProReqSE. Esta proposta foi elaborada com base nos resultados da avaliação do processo de desenvolvimento de software de uma MPE desenvolvedora de SEs e estudos sobre processos de desenvolvimento, processo de requisitos e software embarcado.

Dessa forma, realizada a revisão bibliográfica, a elaboração e aplicação da avaliação dos processos de gerenciamento e engenharia de requisitos das MPEs desenvolvedoras de SEs e a modelagem do ProReqSE, verificou-se, dentro do escopo delimitado para o estudo de caso que:

- a combinação dos resultados obtidos na avaliação aliado as características do desenvolvimento de software embarcado identificadas nas pesquisas e o conhecimento empírico dos pesquisadores serviram como subsídio à elaboração do processo apresentado. O resultado foi o ProReqSE, um processo flexível, abrangente e facilmente utilizável pelas equipes desenvolvedoras de softwares embarcados.

O trabalho possui algumas limitações, dentre elas destacam-se:

- a carência de MPEs desenvolvedoras de software embarcado dispostas a participar da pesquisa.
- a limitação do escopo do projeto a modelagem do processo de Gerenciamento e Engenharia de Requisitos.
- além disso, o fato dos pesquisadores envolvidos neste estudo não vivenciarem o contexto do desenvolvimento de sistemas embarcados dificultou a elaboração do programa de medições utilizado na avaliação. O nível de detalhe das questões, os termos a serem utilizados no questionário e a definição das métricas de cada subquestão possivelmente não tenham sido definidos de forma que pudessem avaliar precisamente as deficiências e os pontos fortes do processo de requisitos das empresas participantes.

Diante do exposto, é importante ressaltar que o desenvolvimento de novas pesquisas que resolvam estas limitações seria importante para estabelecer novos processos de desenvolvimento de software para MPEs desenvolvedoras de SEs, além de aprimorar e aplicar o processo desenvolvido.

## 7. Referências

ANQUETIL, N.; OLIVEIRA, K.M.; SOUZA, K.D. Uso do GQM para Avaliar a Implantação de Processo de Manutenção de Software. IV Simpósio Brasileiro de Qualidade de Software, 2005.

BARBIERO, A.A. Ambiente de Suporte ao Projeto de Sistemas Embarcados. Curitiba: Universidade Federal do Paraná, 2006.

BARRETO, R.S. Sistemas Embarcados: o novo boom da Informática? Sociedade Brasileira de Computação, 2006.

CARRO, L.; WAGNER, F.R. Sistemas Computacionais Embarcados. Vol. 56. Academic Press, 2002.

GASTALDO, D.L.; MIDORIKAWA, E.T. Processo de Engenharia de Requisitos Aplicado a Requisitos Não-Funcionais de Desempenho – Um Estudo de Caso, 2003.

GLADCHEFF, A. P.; SANCHES, R.; DA SILVA, D.M. Um Instrumento de Avaliação de Qualidade de Software Educacional: como elaborá-lo. Simpósio Brasileiro de Engenharia de Software - VIII Workshop de Qualidade de Software, 2001.

GRAAF, B.; LORMANS, M.; TOETENEL, H. Embedded Software Engineering: The State of the Practice. IEEE Software, 2003: p.61-69.

HEWETT, R.; SEKER, R. A Risk Assessment Model of Embedded Software Systems. IEEE International Conference on Man and Cybernetics Systems 4, 2005: p.3238- 3243.

HOFMANN, H.F.; LEHNER, F. Requirements Engineering as a Success Factor in Software Projects. Vol. 18. 4 vols. 2001.

LATTEMANN, F. E LEHMANN, E. A. “Methodological Approach to the Requirement Specification of Embedded Systems.” *Proceedings of the 1st International Conference on Formal Engineering Methods (ICFEM'97)*, 1997: 183.

LEE, E.A. Embedded Software. Vol. Vol 56. Academic Press, 2002.

LEE, E.A. What's Ahead for Embedded Software? IEEE Computer, 2000: 19-26.

LEE, S., KO, H., JO, D., JEONG, J., KIM, K. “Reusable SW Requirements Development Process: Embedded SW Industry Experiences.” *Proceedings of the 2007 Australian Software Engineering Conference (ASWEC'07)*, 2007: 147-158.

MARWEDEL, P. Embedded System Design. Springer, 2003.

MCT, Ministério da Ciência e Tecnologia. Qualidade e Produtividade no Setor de Software Brasileiro. Comunicação Pessoal, 2002.

NARS, E.; MCDERMID, J.; BERNAT, G. Eliciting and Specifying with Use Cases for Embedded Systems. Proceedings of the Seventh International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002), 2002: 350-357.

PRESSMAN, R. S. Engenharia de Software. 5ª edição. Mc Graw Hill, 2006.

PUSCHNIG, A. E KOLAGARI, R.T. Requirements Engineering in the Development of Innovative Automotive Embedded Systems. Proceedings of the 12th International Requirements Engineering Conference (RE'04), 2004: 328-333.

SOFTEX. Melhoria de Processo de Software Brasileiro - Guia Geral Versão 1.2. <http://www.softex.br/mpsbr/> (in portuguese) (acesso em 15 de 11 de 2007).

SOMMERVILLE, I.; KOTONYA, G. Requirements Engineering. Wiley, 1998.

TAURION, CEZAR. Software Embarcado – A nova onda da Informática. Brasport, 2005.

WANGENHEIM, C.G.V.; RUHE E. G. Análise de Custo Benefício de Mensuração Baseada em GQM – Um Estudo de Caso Replicado. Proceedings of X Conferência Internacional de Qualidade de Software, 1999.

WOLF, W. Computers as Components: Principles of Embedded Computing System Design. McGraw-Hill, 2001.

YAMAURA, T.; MIYAZAKI, H.; ONOMA, A.K. A New Defining Approach for Software Requirement Specifications. Proceedings of the IEEE Workshop on Software Technologies for Future Embedded Systems (WSTFES'03), 2003: 13-16.