

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Ana Flávia Campos Leão

RecMed: Uma Ferramenta de Apoio ao Uso de um Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos

Projeto de Graduação apresentado ao Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Monalessa Perini Barcellos

VITÓRIA
2012

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Ana Flávia Campos Leão

**RecMed: Uma Ferramenta de Apoio ao Uso
de um Conjunto de Recomendações para
Medição de Software Adequada ao Controle
Estatístico de Processos**

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.

Prof. Ricardo de Almeida Falbo, D. Sc.

Prof. Renata Silva Souza Guizzardi, D. Sc.

Vitória, 15 de março de 2012

Aos meus pais, Mário e Helenisse.

AGRADECIMENTOS

À Deus, por me conceder força e persistência para atingir os meus objetivos.

Aos meus pais, Mário e Helenisse, que são a minha base e exemplo de vida, pela compreensão, principalmente nessa reta final em que tive que abrir mão de grande parte das minhas férias, que há muito tempo não podia passar com eles, para conclusão desse trabalho.

À minha orientadora Monalessa, pelas horas dedicadas a esse projeto. Pela prestatividade e boa vontade sempre em responder as minhas dúvidas e pelo vasto conhecimento transmitido.

Aos meus amigos, pela presença, incentivo, compreensão e pelos momentos de distração, quando precisei.

Aos amigos do NEMO, Vitor, Renato, Julião e ao Alexandre pelas dúvidas esclarecidas nos momentos de desespero.

À Ludimila, que me ajudou não somente durante a execução desse trabalho, mas durante toda a minha graduação. Uma grande amiga que levarei para a vida.

Enfim, agradeço a todos que de alguma forma contribuíram para a realização desse trabalho.

RESUMO

Atualmente, as grandes exigências feitas pelo mercado em relação aos produtos de software têm levado as organizações a buscarem a melhoria de seus processos de software, buscando se adequarem à essa realidade. Medição de software é uma prática indispensável à melhoria de processos. Porém, para atingir a alta maturidade em processos de software, a medição tradicional, baseada na comparação entre valores reais e planejados, não é suficiente. É necessário aliar a ela o controle estatístico de processos, que permite analisar o comportamento dos processos, a fim de obter processos estáveis e capazes, o que permite que o desempenho dos processos possa ser previsto em projetos futuros e nas próximas etapas de projetos correntes. Nesse contexto, em (BARCELLOS, 2009a) foi proposto um conjunto de recomendações para medição de software adequada ao controle estatístico de processos, com o objetivo de apoiar as organizações que buscam a alta maturidade em seus processos. Esse conjunto de recomendações consistia, basicamente, de um corpo de conhecimento textual, não possuindo apoio computacional para acesso e utilização das recomendações. Visando prover apoio computacional ao conjunto de recomendações proposto, este trabalho propõe a ferramenta *RecMed*, que permite o registro, acesso, utilização, avaliação e evolução das recomendações de medição.

Palavras-chave: Medição de Software, Controle Estatístico de Processos, Alta Maturidade, Desenvolvimento Web.

SUMÁRIO

| | |
|--|-----------|
| Capítulo 1 - Introdução | 1 |
| 1.1 Introdução | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Histórico do Desenvolvimento do Trabalho | 3 |
| 1.4 Organização do Texto | 3 |
| Capítulo 2 - Medição de Software e Controle Estatístico de Processos | 5 |
| 2.1 Introdução | 5 |
| 2.2 Medição de Software | 5 |
| 2.3 Controle Estatístico de Processos | 7 |
| 2.4 Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos | 11 |
| 2.5 Considerações Finais | 14 |
| Capítulo 3 - Especificação de Requisitos e Análise da Ferramenta <i>RecMed</i> | 15 |
| 3.1 Introdução | 15 |
| 3.2 <i>RecMed</i> : Propósito do Sistema e Descrição do Minimundo | 15 |
| 3.3 Subsistemas da Ferramenta <i>RecMed</i> | 17 |
| 3.4 Casos de Uso da Ferramenta <i>RecMed</i> | 18 |
| 3.5 Diagramas de Classes | 21 |
| Capítulo 4 - Projeto e Implementação da Ferramenta <i>RecMed</i>..... | 25 |
| 4.1 Introdução | 25 |
| 4.2 Arquitetura de Software | 25 |
| 4.3 Detalhamento dos Componentes da Arquitetura..... | 27 |
| 4.4 A Ferramenta <i>RecMed</i> | 34 |
| Capítulo 5 - Considerações Finais | 47 |
| 5.1 Conclusões | 47 |
| 5.2 Trabalhos Futuros | 49 |
| Referências Bibliográficas | 51 |

Capítulo 1

Introdução

1.1 Introdução

Atualmente, com a crescente exigência do mercado por produtos de software cada vez melhores, surgiu a necessidade das organizações melhorarem os seus processos de software. Existem vários *frameworks* de apoio à definição de programas com esse objetivo, nos quais a medição ocupa papel fundamental. Como exemplo, podem ser citados o MR MPS.BR – Modelo de Referência para Melhoria de Processo de Software Brasileiro (SOFTEX, 2011), o CMMI – *Capability Maturity Model Integration* (SEI,2010), a ISO/IEC 15504 – *Information Technology – Process Assessment* (ISO/IEC, 2003) e a ISO/IEC12207 – *Systems and Software Engineering – Software Life Cycle Process* (ISO/IEC, 2008). Alguns desses *frameworks* propõem a implementação da melhoria de processos em níveis, nos quais a maturidade e a capacidade dos processos evoluem gradativamente (BARCELLOS, 2009a).

Para as organizações que se encontram nos níveis iniciais de maturidade (caracterizados nos níveis G a C do MR MPS.BR e nos níveis 2 e 3 do CMMI), a coleta de dados dos projetos e comparação com o que foi planejado é uma prática suficiente. Porém, para as organizações que buscam a alta maturidade em seus processos (caracterizada nos níveis A e B do MR MPS.BR e nos níveis 4 e 5 do CMMI), essa prática sozinha é insuficiente. Na alta maturidade, é necessário realizar o controle estatístico de processos de software para conhecer o seu comportamento e, assim, prever o seu desempenho em projetos futuros, verificando se os objetivos iniciais foram atingidos e estabelecendo ações corretivas, quando necessário (BARCELLOS, 2008).

Para que uma organização esteja apta a realizar o controle estatístico, seus processos de software devem ter alcançado um considerável nível de maturidade, ou seja, ela deve possuir um conjunto de processos padrão definido e implementado e deve utilizar práticas de Engenharia de Software em seus projetos, incluindo a definição e coleta de dados de medidas ao longo dos projetos e armazenamento destes em uma base organizacional de medidas (SARGUT e DEMIRORS, 2006 *apud* BARCELLOS, 2009a).

Um dos maiores problemas percebidos nas organizações é a dificuldade em coletar dados adequados para que os resultados da medição sejam eficientes. Conseqüentemente, uma vez que a medição é a base para o controle estatístico de processos, pois fornece os dados necessários à análise

do comportamento dos processos, as práticas da alta maturidade acabam não sendo possíveis para grande parte das organizações de software (BARCELLOS, 2009a).

Nesse sentido, uma organização pode minimizar o esforço e o tempo despendidos na preparação para o controle estatístico de processos realizando, desde o início, com um programa de medição bem definido, que oriente a alimentação da base de medidas organizacional com medidas e dados aplicáveis ao controle estatístico de processos. Porém, analisando-se os estudos e experiências registrados, esse não tem sido um fato comum (WANG e LI, 2005 *apud* BARCELLOS, 2009).

Considerando esse cenário, Barcellos (2009a) propôs uma estratégia para auxiliar as organizações a realizarem o controle estatístico de seus processos. Um dos componentes dessa estratégia é um Conjunto de Recomendações de Medição de Software Adequada ao Controle Estatístico de Processos (CRMS), que apresenta orientações para apoiar a realização do processo de medição de software desde os níveis iniciais, visando ao controle estatístico de processos.

O conjunto de recomendações proposto em (BARCELLOS, 2009a) consiste, basicamente, de um corpo de conhecimento textual, não possuindo apoio computacional para acesso e utilização das recomendações. Acreditando que o desenvolvimento de um apoio computacional web auxilie as organizações a utilizarem e avaliarem as recomendações, bem como possa auxiliar a evolução das recomendações, este trabalho propõe a ferramenta *RecMed*.

1.2 Objetivos

O objetivo geral desse trabalho é desenvolver uma ferramenta computacional (*RecMed*) para prover apoio computacional ao conjunto de recomendações proposto em (BARCELLOS, 2009a), permitindo o registro, acesso, utilização, avaliação e evolução das recomendações de medição.

Para alcançar esse objetivo geral, foram definidos como objetivos específicos desse trabalho:

- (i) Identificar e documentar os requisitos da ferramenta;
- (ii) Realizar a modelagem comportamental e estrutural da ferramenta e documentar na Especificação de Requisitos da ferramenta;
- (iii) Definir a arquitetura da ferramenta e detalhá-la em um Documento de Projeto;
- (iv) Implementar a ferramenta.

1.3 Histórico de Desenvolvimento do Trabalho

A metodologia utilizada no desenvolvimento deste trabalho foi composta pelas seguintes atividades:

- (i) *Revisão Bibliográfica:* O trabalho teve início com uma revisão bibliográfica sobre alguns assuntos, principalmente sobre Medição de Software e Controle Estatístico de Processos. Foram analisados artigos científicos e trabalhos acadêmicos, sendo o principal deles, (BARCELLOS, 2009a), tese de doutorado que é a base deste trabalho.
- (ii) *Estudo de Tecnologias:* Nessa etapa, foi necessário o estudo de tecnologias utilizadas para o desenvolvimento da ferramenta, tais como: Linguagem de Programação JAVA; Ambiente de Desenvolvimento *Eclipse EE IDE for Web Developers*; Servidor Web *Apache Tomcat*; Banco de Dados *Postgree SQL*; *Hibernate* (*framework* que realiza o mapeamento Objeto/Relacional); *Spring* (*framework* que provê uma fábrica de *beans* com injeção automática de dependências); *Zkoss* (*framework* AJAX responsável pela criação das interfaces gráficas); e JPA (*Java Persistence API*).
- (iii) *Elaboração da Documentação da Ferramenta:* Nessa etapa, foi definida a documentação da ferramenta. A princípio foi elaborado o Documento de Requisitos, apresentando uma descrição geral do minimundo do sistema e definição dos requisitos e regras de negócio. Em seguida, foi elaborada a Especificação de Requisitos, apresentando a identificação dos subsistemas, casos de uso, modelo estrutural, modelo dinâmico e glossário do projeto. Por último, foi elaborado o Documento de Projeto, contendo a arquitetura do software e projeto detalhado de cada um de seus componentes.
- (iv) *Implementação e Testes da Ferramenta:* Nessa etapa, a ferramenta foi implementada e testada, tendo sido realizados testes ao longo do desenvolvimento, bem como após a conclusão de funcionalidades presentes na ferramenta.
- (v) *Redação da Monografia:* Nessa etapa, foi realizada a escrita desta monografia. Vale ressaltar que em paralelo com a escrita da monografia houve a finalização da implementação da ferramenta.

1.4 Organização do Texto

Neste capítulo, foi apresentada uma introdução, contendo uma breve descrição do domínio do problema, o contexto em que ele está inserido e a motivação que levou a esse trabalho. Além deste capítulo, a monografia possui mais 4 capítulos, a saber:

Capítulo 2 – Medição de Software e Controle Estatístico de Processos: Apresenta uma fundamentação teórica sobre Medição de Software e Controle Estatístico de Processos. Apresenta, ainda, o Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos (CRMS), definido em (BARCELLOS,2009a).

Capítulo 3 – Especificação de Requisitos e Análise da Ferramenta RecMed: apresenta a descrição do minimundo contemplado pela ferramenta, seus casos de uso e modelos de classe.

Capítulo 4 – Projeto e Implementação da Ferramenta RecMed: apresenta a arquitetura definida para a ferramenta, detalha alguns dos componentes dessa arquitetura e apresenta algumas telas da ferramenta.

Capítulo 5 – Considerações Finais: Apresenta as considerações finais do trabalho, incluindo algumas dificuldades encontradas, contribuições e experiências adquiridas no desenvolvimento dessa ferramenta. Além disso, são identificados alguns possíveis trabalhos futuros que poderão surgir a partir deste.

Capítulo 2

Medição de Software e Controle Estatístico de Processos

2.1 Introdução

Este capítulo apresenta os principais aspectos teóricos que fundamentaram este trabalho e está organizado em 4 seções, além desta introdução. A seção 2.2 aborda a Medição de Software, destacando os principais conceitos e o processo de medição. A seção 2.3 aborda o Controle Estatístico de Processos, mostrando os principais conceitos e o processo de melhoria baseado no Controle Estatístico de Processos. A seção 2.4 apresenta o Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos (CRMS), proposto em (BARCELLOS, 2009a). Por fim, a seção 2.5 apresenta as considerações finais do capítulo.

2.2 Medição de Software

Segundo Bass (1999), medição de software é uma avaliação quantitativa de qualquer aspecto dos processos e produtos da Engenharia de Software, que permite seu melhor entendimento e, com isso, auxilia o planejamento, controle e melhoria do que se produz e de como é produzido. Para realizar a avaliação quantitativa, são utilizadas medidas.

De acordo com a norma ISO/IEC 15939:2001 (ISO/IEC, 2001), medida é a variável à qual um valor é atribuído como resultado de uma medição. As medidas caracterizam, em termos quantitativos, alguma propriedade de um objeto da Engenharia de Software (BASILI e ROMBACH, 1994 *apud* BARCELLOS, 2009a), provendo, assim, informações úteis que podem ser analisadas para dar suporte à tomada de decisões e identificação de ações corretivas apropriadas, se necessário. Como exemplos de medida, podem-se citar: quantidade de defeitos de um produto, prazo estimado do projeto, tamanho do produto, quantidade de horas estimadas para a execução de uma atividade, índice de produtividade, densidade de defeitos nos testes de aceite, etc.

A norma ISO/IEC 15939:2001 (ISO/IEC, 2001) define medição como o conjunto de operações que têm por objetivo determinar o valor de uma medida.

Medição de software é uma prática relativamente recente. Teve início na década de 70, mas por um longo período, seus objetivos não eram claros e seus resultados não eram efetivamente úteis. Somente nos anos 90, impulsionados por algumas aplicações bem sucedidas, foram desenvolvidos

modelos para o processo de medição baseados na melhoria de processos e nos princípios da qualidade total, fornecendo as diretrizes e a infraestrutura básicas para definir, coletar, validar e analisar medidas (BASS *et al* , 1999; FENTON e NEIL, 1999 *apud* BARCELLOS, 2009a).

Atualmente, a medição é considerada uma atividade fundamental entre as práticas de Engenharia de Software. Enquanto, no passado, muitas organizações de software não reconheciam a importância das atividades de medição, hoje ela é considerada uma prática básica da Engenharia de Software, sendo evidenciada por sua inclusão nos requisitos dos níveis iniciais dos modelos que tratam da melhoria de processos em níveis. No CMMI (SEI, 2010), a medição encontra-se no nível 2 (área de processo Medição e Análise) e no MR MPS (SOFTEX, 2011) encontra-se no nível F (processo Medição) (BARCELLOS, 2010).

O processo de medição pode ser definido como um conjunto de passos que deve orientar a realização da medição em uma organização. Um processo de medição eficiente é fator crítico ao sucesso da medição na organização, pois é ele que direciona as atividades a serem realizadas para que, com os resultados da análise dos dados coletados, seja possível a identificação de tendências e antecipação aos problemas, a fim de prover melhor controle dos custos, redução dos riscos, melhoria da qualidade e, conseqüentemente, alcance dos objetivos técnicos e de negócio (WANG e LI, 2005 *apud* BARCELLOS, 2009a).

Atualmente, existem na literatura algumas abordagens que tratam o processo de medição de software. Apesar dessas abordagens apresentarem diferenças entre si, de um modo geral, elas consistem basicamente de quatro passos: (i) definição de medidas; (ii) coleta de medidas; (iii) análise das medidas coletadas e (iv) utilização dos resultados da análise em ações (BARCELLOS, 2009a).

As principais abordagens são apresentadas a seguir (BARCELLOS, 2010):

- **ISO/IEC 15939 (E) Software Engineering – Software Measurement Process:** De acordo com a norma ISO/IEC 15939 (ISO/IEC, 2007), um processo de medição é descrito como um modelo que seja capaz de identificar as atividades que serão requeridas para obter as informações de medição necessárias. Além disso, ele deve especificar como as medidas serão realizadas, como os resultados obtidos serão analisados e como avaliar a qualidade desses resultados. Esse processo consiste de quatro atividades sequenciais que compõem um ciclo iterativo, permitindo um *feedback* de melhoria contínua de processos. Essas atividades são: (i) estabelecer e manter comprometimento com a medição; (ii) planejar o processo de medição; (iii) executar o processo de medição; e (iv) avaliar a medição.

- **IEEE Std 1061-1998 – IEEE Standard for a Software Quality Metrics Methodology:** O IEEE Std 1061-1998 (IEEE, 1998) é uma metodologia para determinar a qualidade dos requisitos e identificar, implementar, analisar e validar como as medidas de qualidade do processo e do produto de software são definidas. Esse processo de medição é composto por cinco atividades: (i) estabelecer os requisitos da qualidade de software; (ii) identificar as medidas de qualidade de software; (iii) implementar as medidas; (iv) analisar os resultados das medidas; e (v) validar as medidas.
- **Practical Software Measurement – PSM:** PSM (McGARRY *et al* , 2002) é uma abordagem para medição de software orientada às necessidades de informação organizacionais aderente à ISO/IEC 15939. Ela possui dois componentes: um Modelo de Informação de Medição e um Processo de Medição. Essa abordagem propõe um processo de medição composto por quatro fases: (i) planejamento; (ii) execução; (iii) avaliação; e (iv) comprometimento.

Conforme já dito anteriormente, a medição de software constitui a base para a melhoria de processos. Nos níveis iniciais de maturidade, é realizada a medição tradicional. Na alta maturidade, a medição passa a incluir a análise do comportamento de processos, por meio do controle estatístico de processos, que é brevemente discutido na próxima seção.

2.3 Controle Estatístico de Processos

O controle estatístico de processos foi originalmente desenvolvido para implementar um processo de melhoria contínua em linhas de produção na área de manufatura, envolvendo o uso de ferramentas estatísticas e técnicas de resolução de problemas com o objetivo de detectar padrões de variação no processo de produção para garantir que os padrões de qualidade estabelecidos para os produtos fossem alcançados. É utilizado para determinar se um processo está sob controle sob o ponto de vista estatístico (BARCELLOS, 2009b).

Com o sucesso da sua utilização na área de manufatura, ele passou a ser utilizado em diversas outras áreas, como por exemplo, no desenvolvimento de software.

No contexto do controle estatístico de processos, dois conceitos são importantes: *estabilidade* e *capacidade*. Um processo é considerado *estável* se seu comportamento for previsível e repetível. Por

outro lado, um processo é considerado *capaz*, se, além de ser estável, ele consegue alcançar os objetivos propostos (BARCELLOS, 2009b).

Em relação à estabilidade, é importante destacar que é intrínseco aos processos apresentar variações em seu comportamento. Sendo assim, um processo estável não é um processo que não apresenta variações e, sim, um processo que apresenta variações aceitáveis, que ocorrem dentro de limites previsíveis, que caracterizam a repetitividade de seu comportamento (BARCELLOS, 2009b).

Essas variações aceitáveis são provocadas pelas *causas comuns* (SHEWART, 1980), que são variações que pertencem ao próprio processo. Considerando processos de software, um exemplo de causa comum pode ser a diferença de produtividade entre os membros de uma equipe relativamente homogênea. Cada membro executa um certo processo com uma determinada produtividade, sendo assim, as variações causadas no comportamento desse processo, por essa diferença de produtividade entre seus executores, são previsíveis (BARCELLOS, 2009a).

Por outro lado, as chamadas *causas especiais* (SHEWART, 1980) provocam desvios que excedem os limites de variação aceitável para o comportamento do processo, revelando um processo instável. Considerando processos de software, a inclusão de um membro inexperiente na equipe pode alterar o comportamento do processo além do esperado, caracterizando uma causa especial (BARCELLOS, 2009a).

Quando todas as variações no comportamento de um processo são aceitáveis, ou seja, quando não há causas especiais, o processo é dito estar *sob controle estatístico*. Um processo sob controle estatístico é um processo estável (FLORAC e CARLETON, 1999 *apud* BARCELLOS, 2009a).

Uma vez estabilizado, a capacidade do processo deve ser analisada. A capacidade descreve os limites de resultados que se espera que o processo alcance para atingir os objetivos estabelecidos. Caso o processo não seja capaz, ele deve ser alterado através da realização de ações de melhoria que busquem o alcance da capacidade desejada. Melhorar a capacidade de um processo significa diminuir os limites de variação que são considerados aceitáveis para seu comportamento, ou seja, consiste em tratar as causas comuns (BARCELLOS, 2009b).

Quando um processo se torna estável e capaz, novos objetivos devem ser traçados, de modo que o processo possa melhorar continuamente. Segundo Barcellos (2009b), muitas organizações conhecidas pela excelência de seus programas de qualidade estabelecem limites de variação bastante “estritos” para os processos e, uma vez que estes são alcançados, a organização obtém processos estáveis, capazes e com um grau de variabilidade consideravelmente baixo.

Quanto menor for o limite de variação dos processos, menores são as chances de desvios entre os valores planejados e realizados nos projetos, ou seja, maior é a aderência dos projetos aos cronogramas, orçamentos e demais planejamentos estabelecidos. Conseqüentemente, melhor será a gerência dos projetos e processos (BARCELLOS, 2009b).

2.3.1 Métodos Estatísticos

Existe uma grande variedade de métodos estatísticos que podem ser utilizados como ferramentas analíticas para representar e analisar os dados coletados para as medidas. Como exemplos, têm-se gráficos de barras, diagramas de tendências, histogramas e gráficos de controle, entre outros.

Os gráficos de controle, muito utilizados no controle estatístico de processos, são capazes de medir a variação dos processos e avaliar sua estabilidade. Associam métodos de controle estatístico e representação gráfica para quantificar o comportamento de processos, auxiliando a detectar os *sinais* de variação no comportamento dos processos e a diferenciá-los dos *ruídos*. Os ruídos dizem respeito às variações que são aceitáveis e são intrínsecas aos processos (causas comuns). Já os sinais indicam variações que precisam ser analisadas em busca de uma melhoria dos processos (causas especiais) (BARCELLOS, 2009b).

Existem diversos tipos de gráficos de controle e cada um deles é aplicável a determinadas situações. O *layout* básico de um gráfico de controle é ilustrado na Figura 2.1. Tanto a linha central quanto os limites superior e inferior representam estimativas que são calculadas a partir de um conjunto de medidas coletadas. Os limites superior e inferior ficam a uma distância de três desvios padrão em relação à linha central. A linha central e os limites não podem ser arbitrários, uma vez que são eles que refletem o comportamento atual do processo. Seus valores são obtidos aplicando-se as expressões e constantes definidas pelo tipo de gráfico de controle a ser utilizado (BARCELLOS, 2009b).

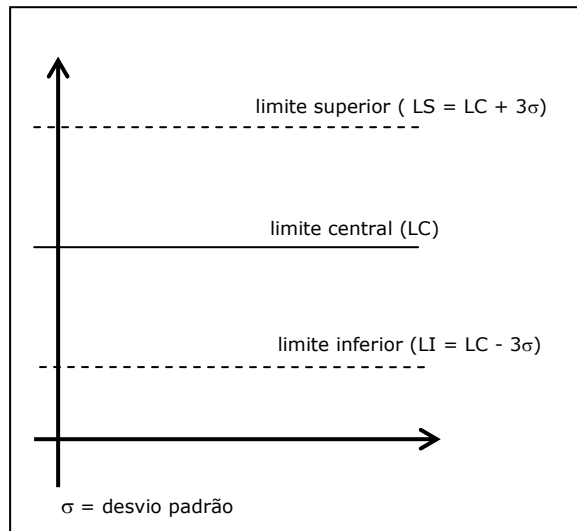


Figura 2.1 – Layout básico de um gráfico de controle.

A Figura 2.2 ilustra um exemplo de aplicação de um gráfico de controle para representar as medidas coletadas em um processo estável, ou seja, onde não há causas especiais. O gráfico representa a média diária de horas dedicadas a atividades de suporte por semana em uma determinada empresa.

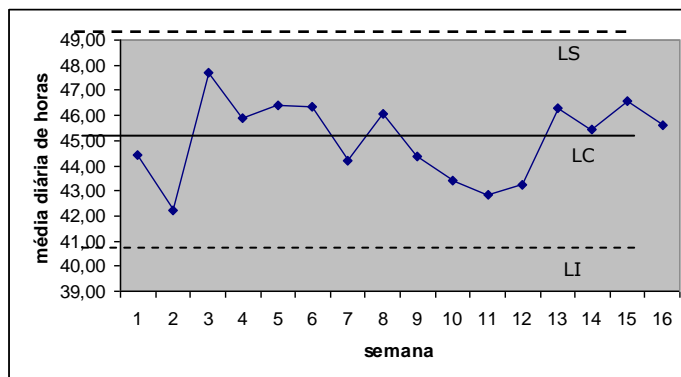


Figura 2.2 – Processo estável.

Na Figura 2.3, é apresentado um gráfico que ilustra um processo cujo comportamento extrapolou os limites de variação aceitáveis, sendo identificados pontos cujas causas de variação (causa especial) devem ser investigadas. O gráfico representa o número de problemas relatados pelos clientes diariamente à área de suporte de uma organização que não foram resolvidos (PNR).

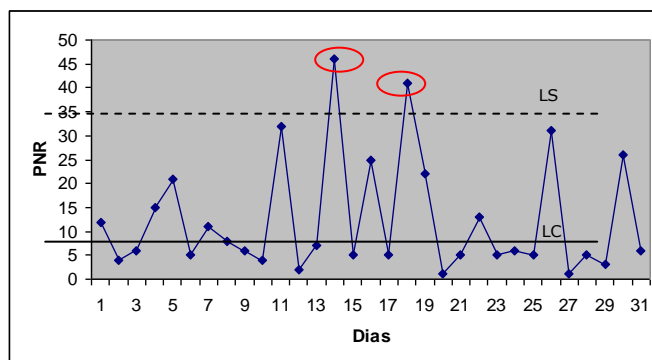


Figura 2.3 – Processo com causas especiais explícitas.

2.4 Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos

O Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos (CRMS) foi proposto em (BARCELLOS, 2009a) e reúne orientações que visam apoiar a realização do processo de medição de software. Ele baseia-se, principalmente, nos resultados de um estudo baseado em revisão sistemática da literatura que identificou fatores relacionados à medição que influenciam negativa e positivamente na realização do controle estatístico de processos (BARCELLOS, 2009a), na conceituação provida pela Ontologia de Medição de Software proposta em (BARCELLOS, 2009a), em relatos da literatura e no conhecimento obtido através de experiências práticas realizadas durante o desenvolvimento da estratégia proposta em (BARCELLOS, 2009a).

Além disso, algumas normas e padrões foram considerados na definição do CRMS: ISO/IEC 15939 (ISO/IEC, 2002), IEEE Std 1061 (IEEE, 1998), *Practical Software Measurement* (PSM) (McGARRY *et al* , 2002), aspectos relacionados à medição presentes no MR MPS (SOFTEX, 2011), CMMI (SEI, 2010) e ISO/IEC 15504 (ISO/IEC, 2003).

O CRMS é composto por vinte recomendações organizadas em cinco grupos: Preparação da Medição de Software, Alinhamento da Medição de Software aos Objetivos Organizacionais e dos Projetos, Definição de Medidas de Software, Realização de Medições de Software e Análise de Medições de Software.

Na Figura 2.4, é apresentada a visão geral do CRMS, onde são identificados os aspectos tratados pelas recomendações que compõem cada um de seus grupos. Em seguida, cada grupo de recomendações é descrito.

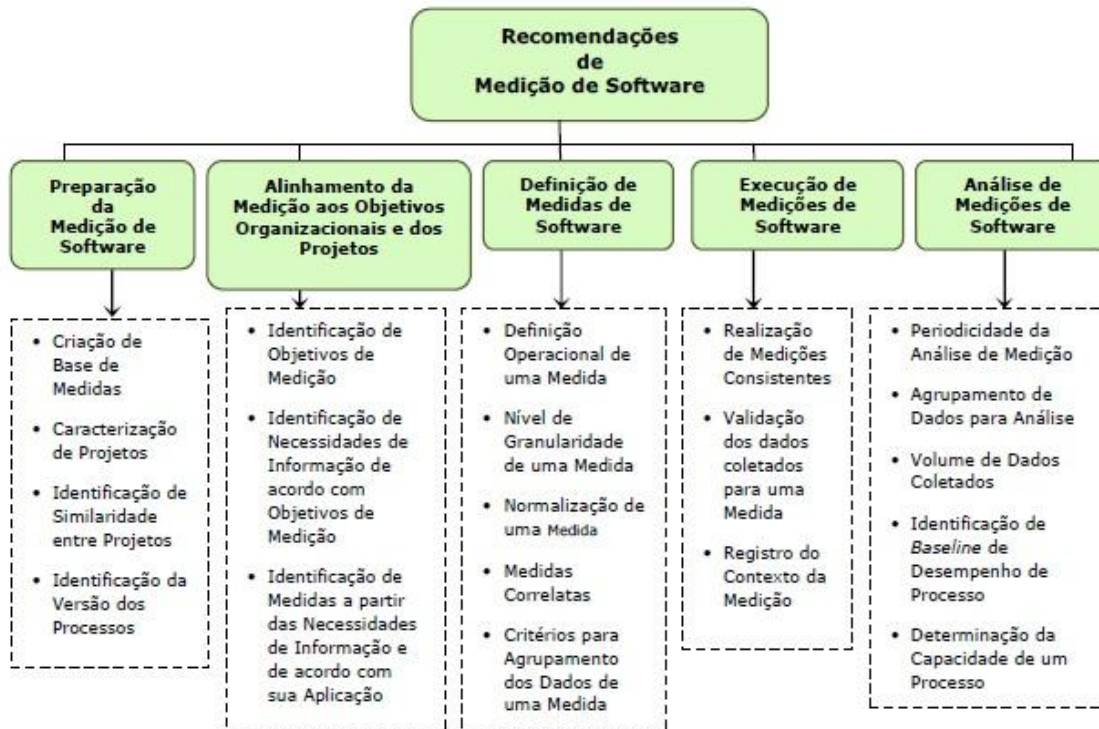


Figura 2.4 – Visão geral do Conjunto de Recomendações para Medição de Software.

- *Preparação da Medição de Software*: contém recomendações relacionadas a aspectos que devem ser considerados antes da implantação de medição em uma organização e sem os quais não é possível realizar a medição de forma adequada.
- *Alinhamento da Medição de Software aos Objetivos Organizacionais e dos Projetos* contém recomendações que visam à realização de medições alinhadas aos objetivos de negócio da organização e aos objetivos específicos dos projetos. Abordam a elaboração do Plano de Medição da organização e dos projetos considerando-se o Planejamento Estratégico da organização. Para isso, guiam a identificação de medidas úteis e alinhadas aos objetivos estabelecidos.
- *Definição de Medidas de Software*: contém recomendações para definir medidas adequadamente. Inclui o estabelecimento de definições operacionais para as medidas e trata outros aspectos relevantes, a saber: nível de granularidade, normalização, medidas correlatas e critérios para agrupamento de dados das medidas.
- *Realização de Medições de Software*: contém recomendações para a execução de medições de software, que consiste da coleta e armazenamento de dados para as medidas.

- *Análise de Medições de Software*: contém recomendações para que a análise dos dados coletados para as medidas forneça as informações necessárias identificadas no Plano de Medição, apoiando, assim, a tomada de decisões e a identificação de ações corretivas e de melhoria.

Em cada grupo de recomendações, além das recomendações propriamente ditas, o CRMS também inclui uma pequena fundamentação teórica referente a cada aspecto abordado, a fim de que as informações nela contidas contribuam para um melhor entendimento das recomendações realizadas.

A seguir, como exemplo, são apresentadas as recomendações que tratam o aspecto Caracterização de Projetos:

| Grupo: Preparação da Medição de Software | | | | | | | |
|---|---|-----|--|-----|---|-----|---|
| Aspecto: Caracterização de Projetos | | | | | | | |
| Propósito | Orientar a caracterização dos projetos em uma organização. A caracterização deve permitir identificar os perfis de projetos que são desenvolvidos, bem como obter informações de contexto dos dados coletados para as medidas nos projetos. | | | | | | |
| Fundamentação Teórica | A identificação de critérios que caracterizem os projetos de uma organização é imprescindível para a identificação dos projetos similares e uso dos dados coletados para as medidas de maneira correta. A caracterização dos projetos deve incluir os critérios relevantes para o registro e posterior identificação dos perfis de projetos. Ela é considerada satisfatória quando os subconjuntos formados pelos projetos que possuem o mesmo perfil, ou seja, cujos critérios de caracterização possuem os mesmos valores, são homogêneos (KITCHENHAM <i>et al.</i> , 2007). | | | | | | |
| Recomendações | <table border="1"> <tr> <td>R1.</td> <td>Não definir uma caracterização baseada em poucos critérios ou em critérios muito amplos, que, normalmente, permitem a formação de grupos heterogêneos de projetos.</td> </tr> <tr> <td>R2.</td> <td>Incluir, no conjunto de critérios, características de todos os elementos relevantes envolvidos em um projeto, tais como: ambiente (exemplos de critérios: distribuição geográfica dos participantes do projeto e infraestrutura disponível), recursos humanos (exemplos de critérios: experiência da equipe do projeto em relação ao domínio, tecnologia e processo utilizados, tamanho da equipe do projeto), produto desenvolvido (exemplos de critérios: tipo de software e domínio do software), processo utilizado (exemplos de critérios: modelo de ciclo de vida utilizado e processo adotado), tecnologias envolvidas (exemplos de critérios: linguagem de programação e banco de dados utilizados), cliente (exemplo de critério: tipo de cliente e porte do cliente) e o próprio projeto (exemplos de critérios: tamanho do projeto e restrições do projeto).</td> </tr> <tr> <td>R3.</td> <td>Tornar a caracterização de projetos explícita na base de medidas, permitindo a identificação dos critérios definidos e do valor atribuído para cada critério em cada projeto realizado.</td> </tr> </table> | R1. | Não definir uma caracterização baseada em poucos critérios ou em critérios muito amplos, que, normalmente, permitem a formação de grupos heterogêneos de projetos. | R2. | Incluir, no conjunto de critérios, características de todos os elementos relevantes envolvidos em um projeto, tais como: ambiente (exemplos de critérios: distribuição geográfica dos participantes do projeto e infraestrutura disponível), recursos humanos (exemplos de critérios: experiência da equipe do projeto em relação ao domínio, tecnologia e processo utilizados, tamanho da equipe do projeto), produto desenvolvido (exemplos de critérios: tipo de software e domínio do software), processo utilizado (exemplos de critérios: modelo de ciclo de vida utilizado e processo adotado), tecnologias envolvidas (exemplos de critérios: linguagem de programação e banco de dados utilizados), cliente (exemplo de critério: tipo de cliente e porte do cliente) e o próprio projeto (exemplos de critérios: tamanho do projeto e restrições do projeto). | R3. | Tornar a caracterização de projetos explícita na base de medidas, permitindo a identificação dos critérios definidos e do valor atribuído para cada critério em cada projeto realizado. |
| R1. | Não definir uma caracterização baseada em poucos critérios ou em critérios muito amplos, que, normalmente, permitem a formação de grupos heterogêneos de projetos. | | | | | | |
| R2. | Incluir, no conjunto de critérios, características de todos os elementos relevantes envolvidos em um projeto, tais como: ambiente (exemplos de critérios: distribuição geográfica dos participantes do projeto e infraestrutura disponível), recursos humanos (exemplos de critérios: experiência da equipe do projeto em relação ao domínio, tecnologia e processo utilizados, tamanho da equipe do projeto), produto desenvolvido (exemplos de critérios: tipo de software e domínio do software), processo utilizado (exemplos de critérios: modelo de ciclo de vida utilizado e processo adotado), tecnologias envolvidas (exemplos de critérios: linguagem de programação e banco de dados utilizados), cliente (exemplo de critério: tipo de cliente e porte do cliente) e o próprio projeto (exemplos de critérios: tamanho do projeto e restrições do projeto). | | | | | | |
| R3. | Tornar a caracterização de projetos explícita na base de medidas, permitindo a identificação dos critérios definidos e do valor atribuído para cada critério em cada projeto realizado. | | | | | | |

O conjunto completo de recomendações encontra-se registrado em (BARCELLOS,2009a).

2.5 Considerações Finais

Tem sido crescente o número de organizações que buscam melhorar os seus processos de software, para que sejam capazes de atender as demandas de alta qualidade do mercado. Nesse contexto, é fundamental que haja um processo de medição bem definido, que seja capaz de apoiar os objetivos organizacionais.

Na alta maturidade nos processos, a medição deve ser feita associada ao controle estatístico de processos, buscando obter processos estáveis e capazes, para que seja possível prever e controlar o comportamento dos processos.

Buscando apoiar a realização da medição visando ao controle estatístico de processos, Barcellos (2009a) propôs um Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos (CRMS).

Ao propor o conjunto de recomendações, Barcellos (2009a) registrou que não se pretendia que o conjunto de recomendações proposto fosse completo, contendo todas as possíveis recomendações para realização de medição adequada ao controle estatístico de processos. O conjunto de recomendações foi dito ser um conjunto inicial de recomendações, que foi avaliado por especialistas, e cuja utilização por parte de organizações propiciaria sua evolução.

Uma vez que o CRMS não possuía apoio computacional, a utilização das recomendações pelas organizações tem sido feita informalmente, através da leitura do conteúdo disponível em (BARCELLOS, 2009a). Até o momento, não foi obtido um *feedback* detalhado da utilização das recomendações pelas organizações. Um *feedback* dessa natureza permitiria avaliar a efetividade das recomendações na prática, levando à sua melhoria e, possivelmente, à identificação de novas recomendações.

Buscando atender essa necessidade, a ferramenta *RecMed* foi desenvolvida e é apresentada no próximo capítulo.

Capítulo 3

Especificação de Requisitos e Análise da Ferramenta

RecMed

3.1 Introdução

A Engenharia de Requisitos é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados e gerenciados ao longo de todo o ciclo de vida do software (AURUM e WOHLIN, 2005).

Este capítulo aborda os resultados da Engenharia de Requisitos da ferramenta *RecMed*. Na seção 3.2, são apresentados o objetivo da ferramenta e a descrição do minimundo; na seção 3.3, são apresentados os subsistemas de *RecMed*; na seção 3.4, são apresentados diagramas de casos de uso e na seção 3.5, são apresentados os diagramas de classes.

3.2 *RecMed*: Propósito do Sistema e Descrição do Minimundo

A ferramenta *RecMed* tem como propósito permitir o registro e disponibilização de um conjunto de recomendações que auxiliem as organizações a realizarem medição de software adequada ao controle estatístico de processos e permitir que sejam registradas avaliações dos resultados da utilização dessas recomendações, bem como sugestões de melhoria. Além disso, a ferramenta deve permitir que essas sugestões de melhoria sejam analisadas por um especialista em medição, contribuindo, assim, para a evolução das recomendações.

O minimundo considerado por *RecMed* é descrito a seguir.

Em (BARCELLOS, 2009a), foi definido um conjunto de recomendações para realização de medição de software adequada para o controle estatístico de processos. Esse conjunto de recomendações é o conjunto inicial de recomendações que serão armazenadas na ferramenta *RecMed*. A estrutura do conjunto de recomendações proposto em (BARCELLOS, 2009a) foi utilizada como base para definir a ferramenta *RecMed*.

Recomendações de medição tratam aspectos os quais são organizados em grupos. Inicialmente, há vinte aspectos e cinco grupos predefinidos. Os grupos são: Preparação da Medição de Software, Alinhamento da Medição de Software aos Objetivos Organizacionais e dos Projetos, Definição de

Medidas de Software, Realização de Medições de Software e Análise de Medições de Software. Caracterização de Projetos é um exemplo de aspecto presente no grupo Preparação da Medição de Software.

Cada recomendação trata um determinado aspecto. Para cada aspecto, é definido o propósito das recomendações associadas a ele, além de uma fundamentação teórica e as recomendações propriamente ditas. Recomendações podem incluir exemplos e podem estar relacionadas a atividades do processo de medição.

Pessoas podem utilizar as recomendações presentes no conjunto de recomendações para medição. Para isso, elas devem se registrar, informando seu nome, sua função (por exemplo: gerente de projetos, gerente de qualidade etc.), e-mail, nome da(s) organização(ões) em que trabalham e tipo(s) da(s) organização(ões) (exemplos: empresa de desenvolvimento de software, laboratório de pesquisa de universidade etc.).

Ao utilizar as recomendações, os usuários das recomendações podem registrar informações sobre os resultados obtidos com a sua utilização. O registro dos resultados de utilização das recomendações inclui uma avaliação das recomendações. Podem ser registrados resultados de utilização de várias recomendações (desde que associadas a um mesmo aspecto) ou para uma recomendação específica. Assim, as recomendações podem ser avaliadas em conjunto (por aspecto) ou isoladamente.

Ao avaliar uma recomendação, o usuário deve indicar se, em sua opinião, ela é ótima, boa, regular, ruim ou péssima. Deve, ainda, descrever o contexto em que ela foi utilizada (número de projetos nos quais a recomendação foi utilizada e algumas características desses projetos como tamanho e experiência da equipe) e qual foi o resultado obtido, indicando se foi satisfatório, parcialmente satisfatório ou insatisfatório.

Usuários também podem registrar sugestões de melhoria para as recomendações. Para isso, deve ser indicada a razão da alteração sugerida, que pode ou não ser baseada em um registro de resultados de uso. Ao registrar uma sugestão de melhoria, o usuário indica o nível de criticidade da sugestão segundo seu ponto de vista, podendo ser alto, médio ou baixo.

Usuários não podem alterar recomendações nem criar novas. As alterações nas recomendações, criação de novas e disponibilização das recomendações para uso são responsabilidades de um especialista em medição. Ao criar uma nova recomendação, um especialista pode criar novos aspectos a serem tratados, bem como novos grupos.

O histórico das alterações em recomendações existentes, bem como quais sugestões foram consideradas para realizar as alterações, deve ser mantido.

Quando são realizadas sugestões de melhoria em recomendações, o especialista em medição é comunicado.

As sugestões de melhoria possuem um nível de prioridade (alto, médio ou baixo), que é definido com base nos registros de uso das recomendações.

Uma sugestão de melhoria pode estar *em aberto* (quando é registrada e ainda não foi analisada pelo especialista), *recusada* (quando foi analisada pelo especialista e ele a considerou não pertinente), *aceita* (quando foi analisada pelo especialista e ele a considerou adequada, mas ainda não realizou a alteração) ou *efetivada* (quando a alteração foi realizada).

Com base na descrição do minimundo, foram identificados os requisitos funcionais, não funcionais e regras de negócio de *RecMed*. Tabelas contendo essas informações encontram-se no Documento de Requisitos de *RecMed*.

3.3 Subsistemas da Ferramenta *RecMed*

A Figura 3.1 mostra os subsistemas identificados no contexto do presente projeto, os quais são descritos a seguir.

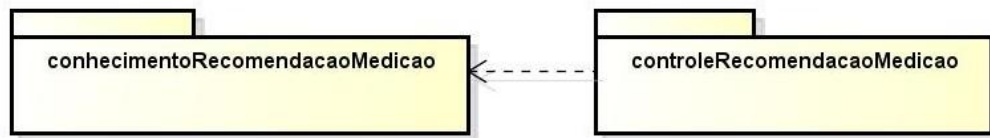


Figura 3.1 - Diagrama de Pacotes e os Subsistemas.

- **conhecimentoRecomendacaoMedicao:** Contém as classes necessárias para o armazenamento das recomendações para medição de software adequada ao controle estatístico de processos.
- **controleRecomendacaoMedicao:** Contém as classes relacionadas com o uso e controle

das recomendações, abrangendo o registro de informações sobre o seu uso, assim como as suas avaliações, sugestões de melhoria e controle de alterações.

3.4 Casos de Uso da Ferramenta *RecMed*

O modelo de casos de uso visa capturar e descrever as funcionalidades que um sistema deve prover para os atores que interagem com o mesmo. Os atores identificados no contexto deste projeto são:

- **Usuário de Recomendação:** Pessoa que utiliza as recomendações, registra informações sobre a utilização, avalia recomendações e sugere melhorias.
- **Especialista em Medição:** Pessoa responsável pelo registro, atualização, avaliação e disponibilização das recomendações.

A seguir são apresentados os casos de uso da ferramenta *RecMed*, organizados por subsistema. Para cada caso de uso são apresentadas breves descrições. A descrição completa dos casos de uso encontra-se no Documento de Especificação de Requisitos de *RecMed*.

3.4.1 Subsistema *conhecimentoRecomendacaoMedicao*

A Figura 3.2 apresenta o diagrama de casos de uso do subsistema *conhecimentoRecomendacaoMedicao*.

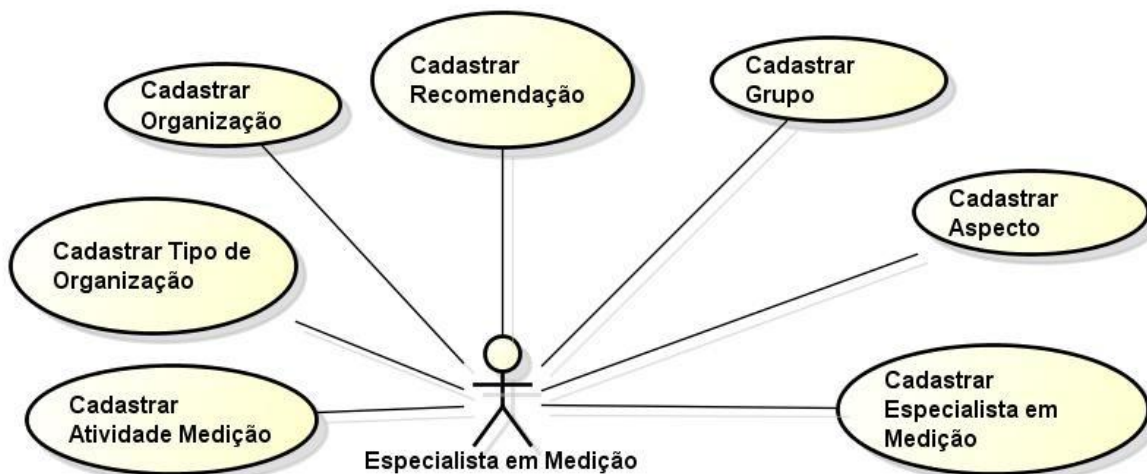


Figura 3.2 – Diagrama de Casos de Uso do Subsistema *conhecimentoRecomendacaoMedicao*.

Todos os casos de uso do subsistema *conhecimentoRecomendacaoMedicao* são cadastrais e envolvem inclusão, exclusão, consulta e alteração. Os cadastros de grupo, aspecto e recomendação são necessários para que as recomendações de medição sejam cadastradas. O cadastro de atividade de medição permite que sejam registradas as atividades do processo de medição. Quando o especialista em medição cadastra recomendações, ele indica em quais das atividades do processo de medição elas serão úteis. Há, ainda, os cadastros de especialista em medição, organização e tipo de organização. O cadastro de organização é a única funcionalidade do subsistema *conhecimentoRecomendacaoMedicao* que pode ser realizada tanto pelo especialista em medição quanto pelo usuário de recomendação, uma vez que, quando o usuário realiza seu cadastro, ele precisa informar a organização em que atua. Caso a organização não esteja registrada na ferramenta, deve ser possível a ele fazer o registro.

3.4.2 Subsistema *controleRecomendacaoMedicao*

A Figura 3.3 apresenta o diagrama de casos de uso do subsistema *controleRecomendacaoMedicao*.

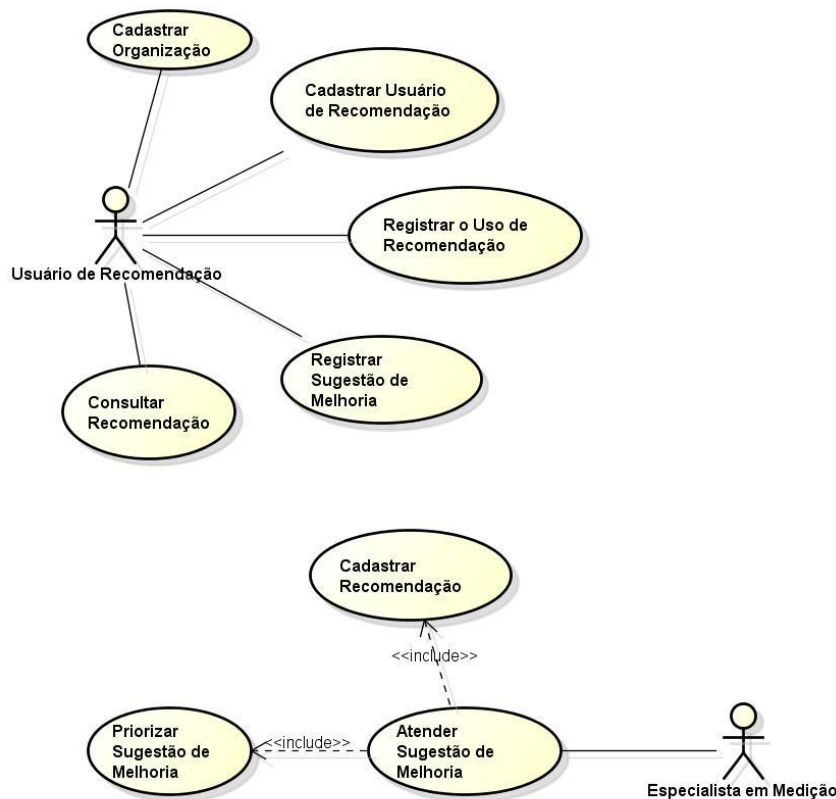


Figura 3.3 – Diagrama de Casos de Uso do Subsistema *controleRecomendacaoMedicao*.

O usuário de recomendação pode realizar o seu cadastro. Ao se cadastrar, é necessário informar a organização em que atua. Caso a organização não esteja registrada na ferramenta, ele pode cadastrá-la. Além disso, ele pode consultar uma recomendação e, após utilizá-la, ele pode registrar os resultados de uso e fazer sugestões de melhoria nas recomendações. O registro de uso da recomendação pode ser de apenas uma recomendação ou de recomendações relacionadas a um aspecto. O registro de uma sugestão de melhoria pode ser baseado em um registro de uso ou não.

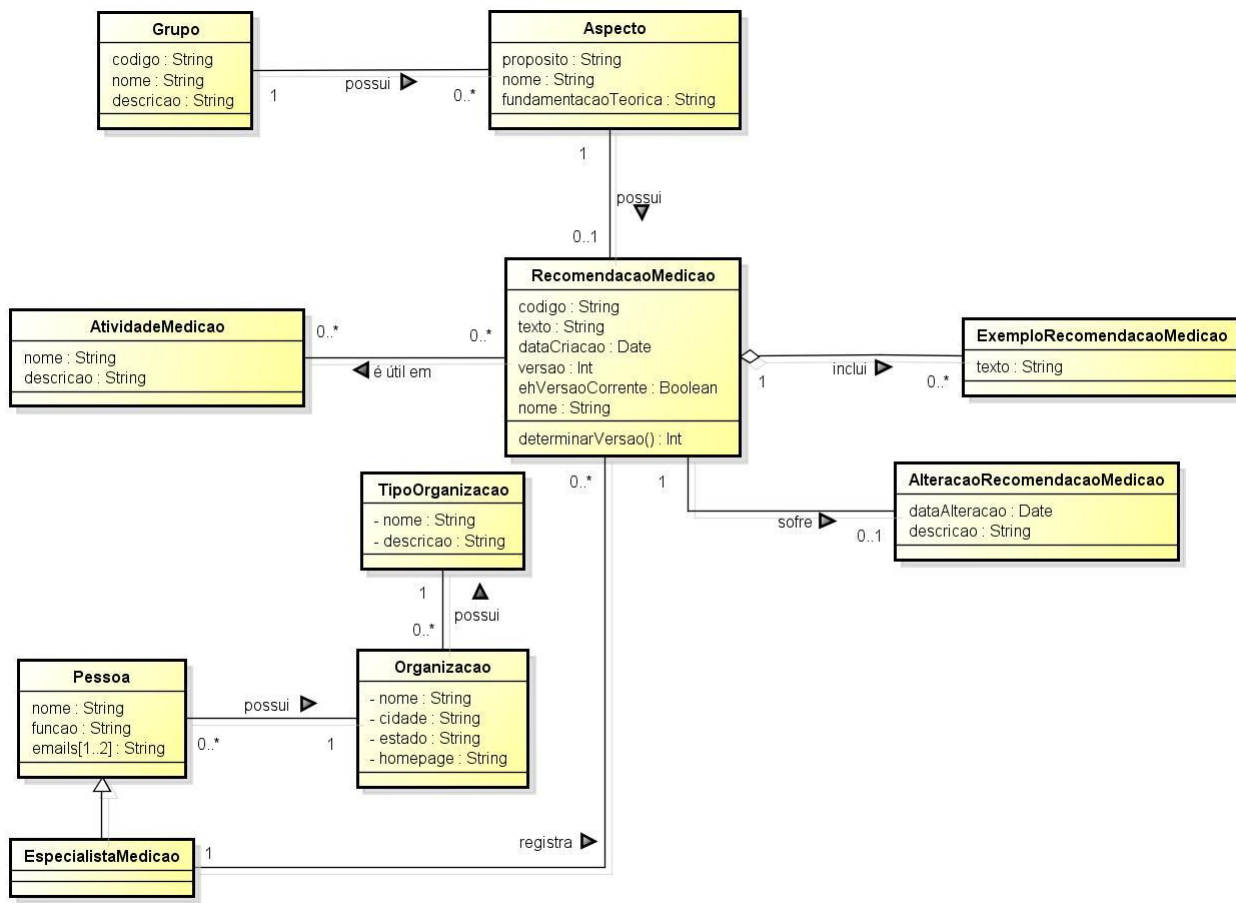
O especialista em medição, por sua vez, pode atender sugestões de melhoria realizadas. Para isso, ele avalia uma sugestão de melhoria realizada e, caso seja aprovada, ele a efetiva, alterando recomendações de medição existentes ou criando novas. Caso haja várias sugestões de melhoria para serem analisadas, o especialista em medição pode solicitar que sejam priorizadas de acordo com seu nível de criticidade.

3.5 Diagramas de Classes

O modelo conceitual estrutural visa capturar e descrever as informações (classes, associações e atributos) que o sistema deve representar para prover as funcionalidades descritas na seção anterior. A seguir, são apresentados os diagramas de classes e uma breve descrição para cada um dos subsistemas identificados no contexto deste projeto. Outras informações, tais como restrições de integridade, diagrama de estados e glossário do projeto, que não são apresentadas neste capítulo, podem ser encontradas no Documento de Especificação de Requisitos de *RecMed*.

3.5.1 Subsistema *conhecimentoRecomendacaoMedicao*

A Figura 3.4 apresenta o diagrama de classes do subsistema *conhecimentoRecomendacaoMedicao*.



powered by astah

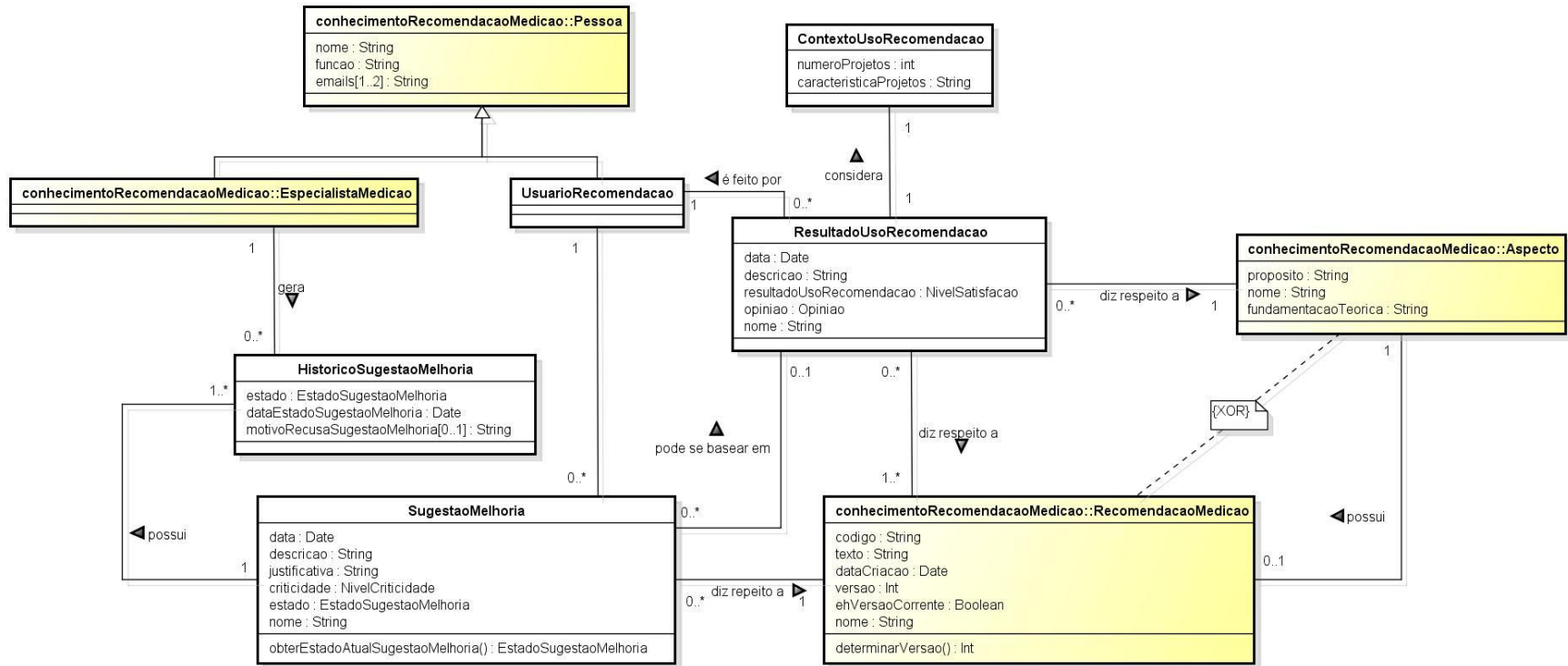
Figura 3.4 – Diagrama de Classes do Subsistema *conhecimentoRecomendacaoMedicao*.

Recomendações de medição (*RecomendacaoMedicao*) tratam aspectos (*Aspecto*), os quais são organizados em grupos (*Grupo*). De um grupo, deseja-se saber o nome, código e descrição. Um grupo possui nenhum (um grupo é existencialmente independente de aspectos) ou mais aspectos. De um aspecto, deseja-se saber o grupo ao qual pertence, nome, propósito e fundamentação teórica. Um aspecto possui nenhuma (um aspecto é existencialmente independente de recomendações) ou mais recomendações de medição. Uma recomendação de medição é útil em nenhuma ou várias atividades de medição (*AtividadeMedicao*). De uma recomendação de medição, deseja-se saber o aspecto ao qual ela pertence, seu código, nome, texto, data de criação, exemplos (se houver), a versão (recomendações podem sofrer alterações, gerando novas versões) e se é a versão corrente. O especialista em medição (*EspecialistaMedicao*) é a pessoa (*Pessoa*) responsável por registrar uma recomendação de medição. De um especialista em medição, deseja-se saber o nome, a organização, a função e o email. De uma organização (*Organizacao*), deseja-se saber o nome, a cidade, o estado, a

homepage e o tipo dessa organização (*TipoOrganizacao*).

3.5.2 Subsistema *controleRecomendacaoMedicao*

A Figura 3.5 apresenta o diagrama de classes do subsistema *controleRecomendacaoMedicao*. As classes em amarelo são as classes do subsistema *conhecimentoRecomendacaoMedicao*.



powered by astah

Figura 3.5 – Diagrama de Classes do Subsystema *controleRecomendacaoMedicao*.

Usuários de recomendação fazem uso de recomendações de medição e podem registrar os resultados desse uso. De um usuário de recomendação (UsuarioRecomendacao), deseja-se saber o nome, a função e o email. O usuário registra o uso de uma recomendação (ResultadoUsoRecomendacao) informando a data, o nome, a descrição, o resultado de uso (satisfatório, parcialmente satisfatório ou insatisfatório), a opinião sobre a recomendação (ótima, boa, regular, ruim ou péssima) e o contexto em que a recomendação foi utilizada (ContextoUsoRecomendacao). O resultado de uso pode ser referente a apenas uma recomendação ou a todas as recomendações de um aspecto. Usuários também registram sugestões de melhoria (SugestaoMelhoria), que podem ser baseadas em um resultado de uso de uma recomendação ou não. De uma sugestão de melhoria, deseja-se saber a data, o nome, a descrição, a justificativa para a sugestão, a criticidade (alta, média ou baixa) e o estado da sugestão de melhoria (em aberto, recusada, aceita ou efetivada). Um especialista em medição pode avaliar uma sugestão de melhoria, e pode aceitar ou recusar essa sugestão. Caso seja aceita, a sugestão é efetivada e uma nova versão da recomendação é gerada. Caso seja recusada, uma justificativa deve ser guardada. O histórico das mudanças de estados das sugestões de melhoria deve ser armazenado (HistoricoSugestaoMelhoria), informando-se a data da mudança de estado, o estado e o motivo da recusa, caso o estado seja recusada.

Capítulo 4

Projeto e Implementação da Ferramenta *RecMed*

4.1 Introdução

Uma vez identificados, documentados e modelados os requisitos, ocorre a fase de projeto, onde o produto de software é modelado de forma que os aspectos tecnológicos que serão utilizados para a implementação do sistema sejam levados em conta. Esses aspectos envolvem: linguagem de programação e *frameworks* utilizados, características de interface com o usuário, arquitetura de software e de hardware e forma de persistência de dados. Por fim, o produto de software é construído e testado.

Neste capítulo, são apresentados os principais aspectos do Projeto de Sistema da ferramenta *RecMed*. Algumas de suas telas também são apresentadas. Na seção 4.2, a arquitetura de software de *RecMed* é descrita. Na seção 4.3, os componentes da arquitetura são detalhados e na seção 4.4, são apresentadas algumas telas da ferramenta. Vale ressaltar, que na seção 4.3, serão detalhados os componentes de apenas um subsistema da ferramenta. O projeto completo encontra-se no Documento de Projeto de Sistema de *RecMed*, que também contém informações sobre as tecnologias utilizadas e sobre as táticas de projeto consideradas para atender os requisitos não funcionais identificados no Documento de Requisitos de *RecMed*.

4.2 Arquitetura de Software

A arquitetura de software da ferramenta *RecMed* baseia-se na combinação de camadas e partições. Inicialmente, para cada subsistema identificado na fase de análise foi definida uma partição, como mostra a Figura 4.1.

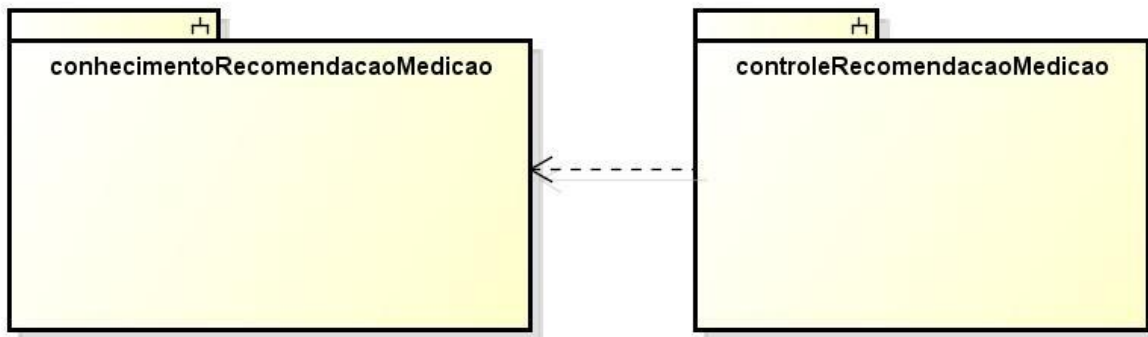


Figura 4.1 – Arquitetura de Software Inicial.

Cada uma dessas partições, por sua vez, está organizada em três camadas, a saber: camadas de Interface com o Usuário (ciu), que trata de aspectos relacionados às interfaces gráficas com os usuários; Lógica de Negócio (cln), onde é implementada a lógica de negócio; e Gerência de Dados (cgd), responsável pela persistência de objetos. A camada de Lógica de Negócio, por sua vez, é subdividida em dois componentes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt).

Além das partições definidas a partir dos subsistemas identificados na fase de análise, foram reutilizados os componentes *_infraestruturaBase* e *_infraestruturaCRUD* presentes na arquitetura do ambiente ODE-Web e próprios para reuso. Também foi reutilizado o pacote *_controleUsuario*, que faz parte do núcleo de ODE-Web e contém funcionalidades referentes a controle de acesso.

A Figura 4.2 mostra o projeto da arquitetura de software da ferramenta *RecMed*.

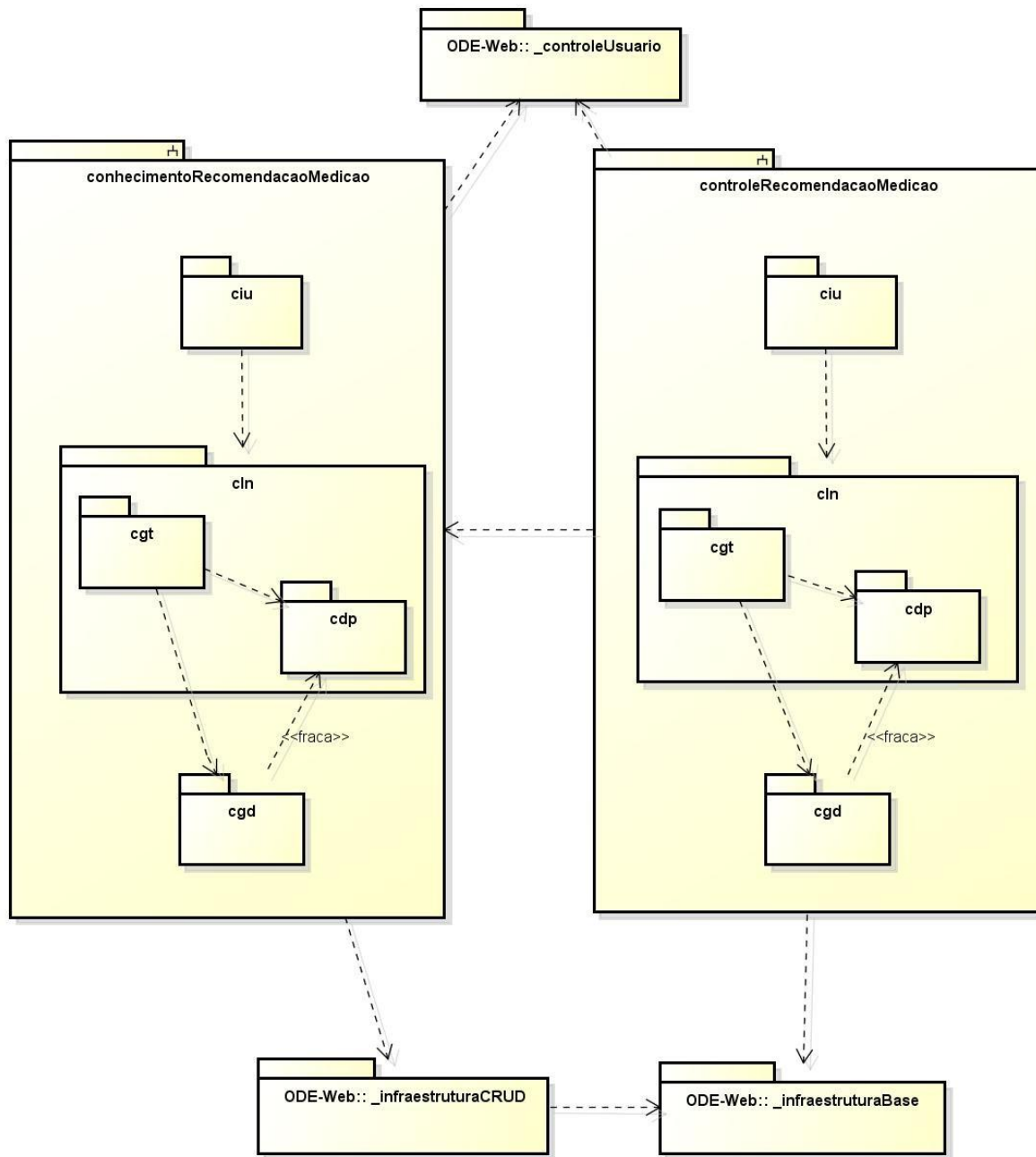


Figura 4.2 – Arquitetura de Software Completa.

4.3 Detalhamento dos Componentes da Arquitetura

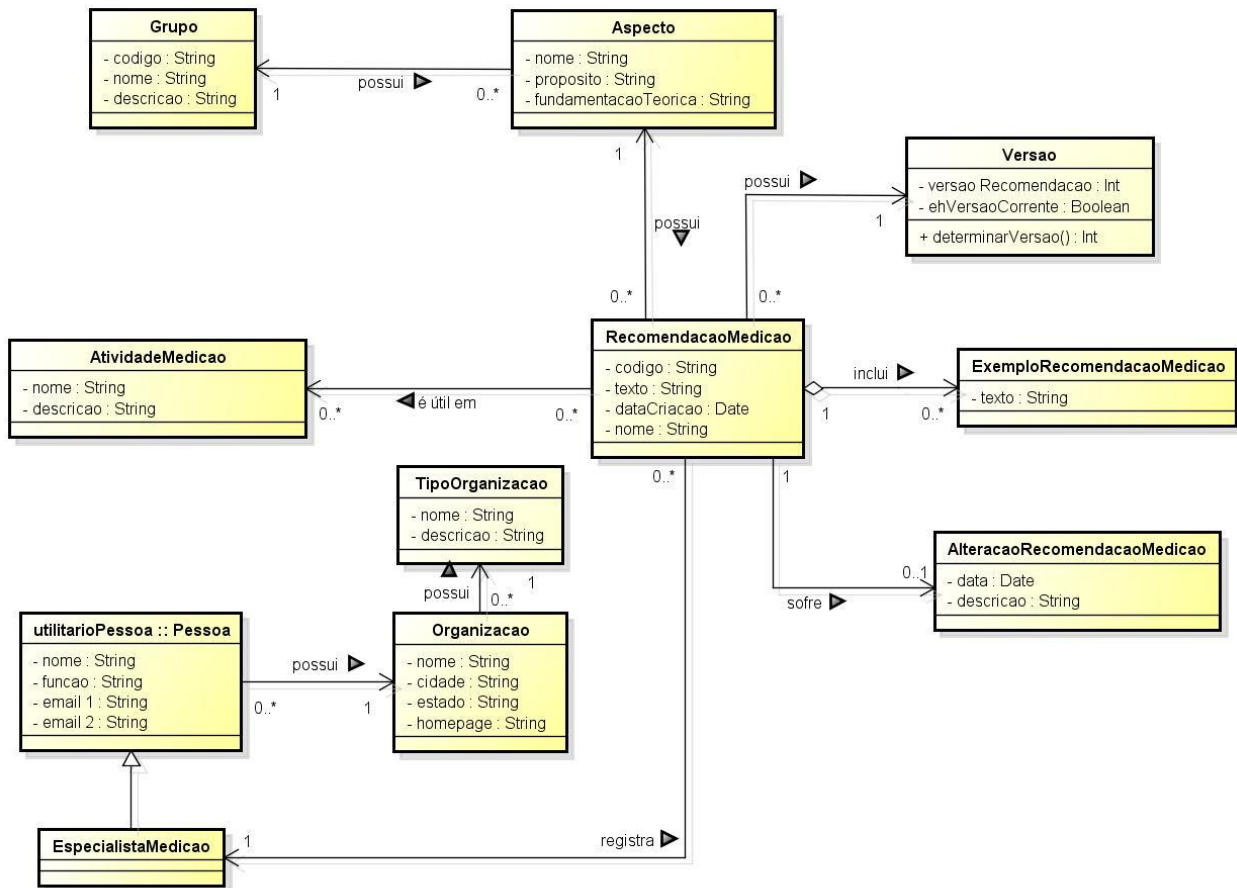
A seguir, é apresentado o projeto detalhado dos componentes da arquitetura dos subsistemas de *RecMed*. Como dito na introdução deste capítulo, apenas um subsistema terá todos os componentes descritos no texto desta monografia. Assim, nesta seção são descritos os componentes da arquitetura do subsistema *controleRecomendacaoMedicao*. Para o subsistema *conhecimentoRecomendacaoMedicao*, é apresentado o modelo de classes de seu componente de domínio do problema

4.3.1 Subsistema *conhecimentoRecomendacaoMedicao*

Conforme discutido anteriormente, o subsistema *conhecimentoRecomendacaoMedicao* está organizado em três camadas: Camada de Lógica de Negócio, Camada de Interface com o Usuário e Camada de Gerência de Dados.

Para organizar a camada de lógica de negócio foi escolhido o padrão Camada de Serviço. Sendo assim, essa camada é dividida em dois componentes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt), como apresentado anteriormente na Figura 4.2. Esse padrão utiliza um componente para tratar a lógica de aplicação (o cgt), o qual recebe as requisições da interface, e um componente para tratar os conceitos do domínio do problema, advindos do modelo conceitual estrutural elaborado na fase de análise (o cdp).

A Figura 4.3 apresenta o diagrama de classes do CDP do subsistema *conhecimentoRecomendacaoMedicao*. Nele, as navegabilidades entre as classes foram identificadas, bem como a visibilidade dos atributos. Na classe Pessoa, o atributo e-mails, que podia receber dois valores, foi transformado em dois atributos: email1 e email2, sendo um deles obrigatório (email1) e o outro não (email2).



powered by astah

Figura 4.3 – Diagrama de Classes do CDP do Subsystema *conhecimentoRecomendacaoMedicao*.

4.3.2 Subsystema *controleRecomendacaoMedicao*

Assim como o subsystema *conhecimentoRecomendacaoMedicao*, o subsystema *controleRecomendacaoMedicao* está organizado em três camadas: Camada de Lógica de Negócio, Camada de Interface com o Usuário e Camada de Gerência de Dados. O detalhamento dos componentes de cada camada é apresentado a seguir.

4.3.2.1 Camada Lógica de Negócio

4.3.2.1.1 Componente de Domínio do Problema (CDP)

A Figura 4.4 apresenta o diagrama de classes do CDP do subsystema *controleRecomendacaoMedicao*. As navegabilidades entre as classes foram identificadas, os tipos

específicos de domínio foram representados como classes e a visibilidade dos atributos foi explicitada.

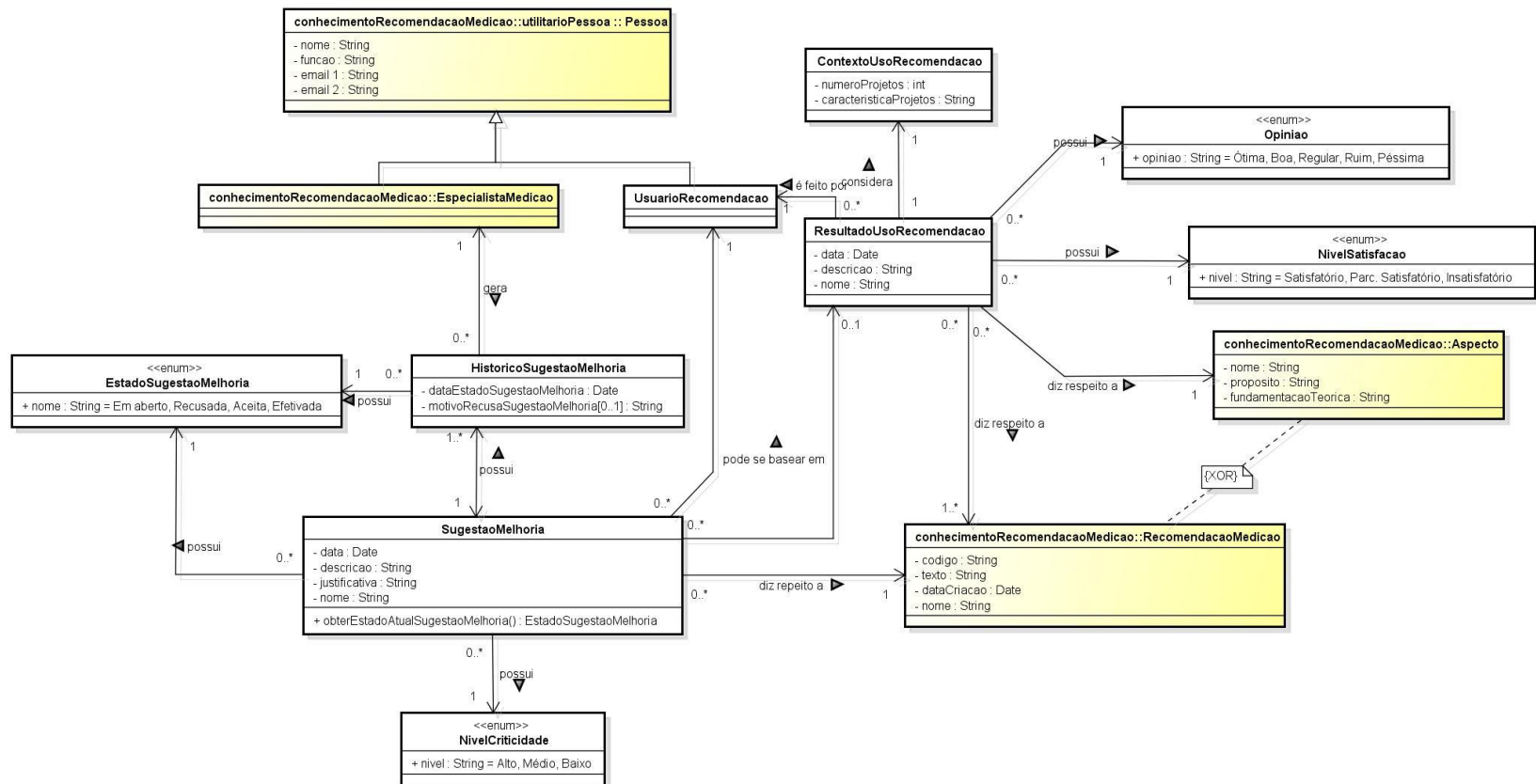


Figura 4.4 – Diagrama de Classes do CDP do Subsistema *controleRecomendacaoMedicao*.

4.3.2.1.2 Componente de Gerência de Tarefas (CGT)

No projeto do CGT, optou-se por mapear as classes de aplicação exatamente como o caso de uso propunha. A Tabela 4.1 sumariza as relações existentes entre as classes do CGT e os casos de uso por elas tratados.

Tabela 4.1 – Classes do CGT e Casos de Uso.

| Classe | Casos de Uso |
|---------------------------------------|-----------------------------------|
| AplCadastrarUsuarioRecomendacao | Cadastrar Usuário de Recomendação |
| AplRegistrarResultadoUsoRecomendacao | Registrar o Uso de Recomendação |
| AplRegistrarSugestaoMelhoria | Registrar Sugestão de Melhoria |
| AplCadastrarRecomendacaoMedicao | Consultar Recomendação |
| AplRegistrarSugestaoMelhoria | Atender Sugestão de Melhoria |
| AplRegistrarHistoricoSugestaoMelhoria | Atender Sugestão de Melhoria |
| AplRegistrarSugestaoMelhoria | Priorizar Sugestão de Melhoria |
| AplCadastrarRecomendacaoMedicao | Cadastrar Recomendação |
| AplCadastrarOrganizacao | Cadastrar Organização |

Uma vez que o projeto do CGT está fortemente relacionado ao projeto da Interface com o Usuário, um único diagrama foi elaborado, o qual é mostrado na Figura 4.5.

4.3.2.2 Camada de Interface com o Usuário.

A Figura 4.5 apresenta o diagrama de classes (parcial) do CIU do subsistema *controleRecomendacaoMedicao*. O diagrama é referente ao registro de uma sugestão de melhoria.

Uma vez que as funcionalidades dos casos de uso Registrar Sugestão de Melhoria com base em um Registro de Uso e Registrar Sugestão de Melhoria sem se basear em um Registro de Uso são praticamente as mesmas, foi criada uma classe controladora geral para o registro de uma sugestão de melhoria, a partir dela, os controladores específicos foram especializados.

O controlador `CtrlRegistroSugestaoMelhoria` relaciona-se às classes de aplicação `AplRegistrarSugestaoMelhoria` e `AplCadastrarGrupo`. A relação com a classe `AplCadastrarGrupo` se faz necessária para que sejam apresentadas as instâncias dessa classe na listagem para que a partir dela, o usuário possa selecionar uma recomendação de medição ou um registro de uso para registrar uma sugestão de melhoria.

4.3.2.3 Camada de Gerência de Dados

A persistência dos objetos deste sistema é realizada em um banco de dados relacional, utilizando a infraestrutura de persistência desenvolvida no contexto do Projeto ODE. Essa infraestrutura utiliza JPA com o *framework* de persistência Hibernate e adota o Padrão DAO. A Figura 4.6 apresenta as principais classes do utilitário de Persistência.

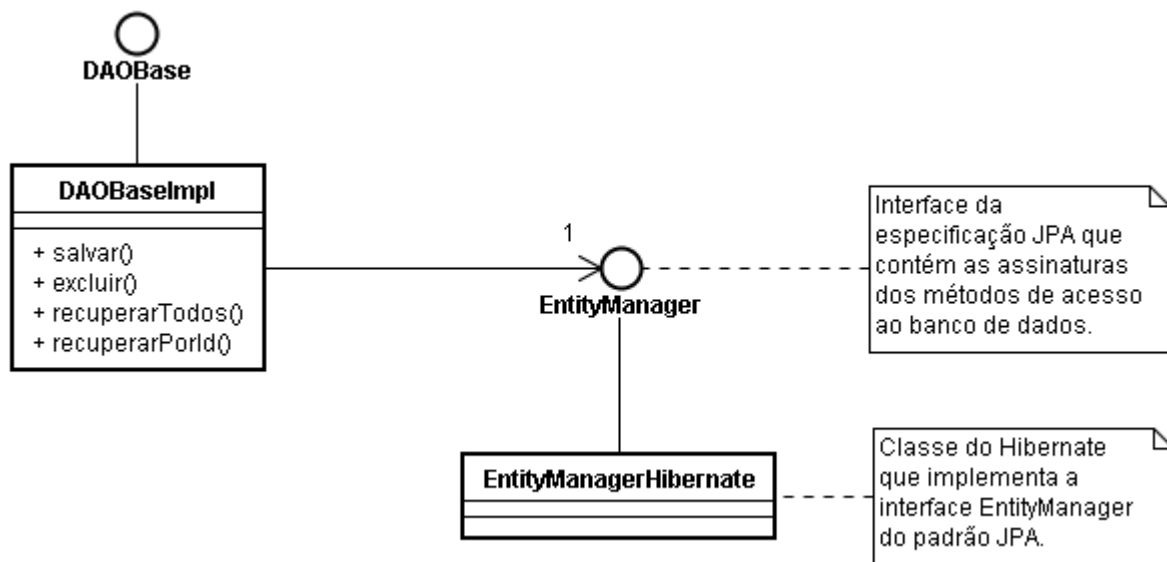


Figura 4.6 – Infraestrutura de Persistência do Projeto ODE.

Para cada classe de domínio a ser persistida, devem ser criadas uma classe (*)DAOImpl e uma interface (*)DAO correspondente. A primeira deve herdar de DAOBaseImpl, uma classe genérica

que possui as funcionalidades básicas de acesso ao mecanismo de persistência, e deve implementar a interface DAO associada. Já a interface DAO da classe a ser persistida deve herdar da interface genérica DAOBase. Seguindo essa abordagem, cada classe a ser persistida tem uma correspondente classe de persistência, responsável pela interação com o banco de dados relacional, e implementa uma interface correspondente, como mostra a Figura 4.7. Na figura, apenas uma classe é apresentada, uma vez que as demais classes do subsistema são modeladas da mesma forma.

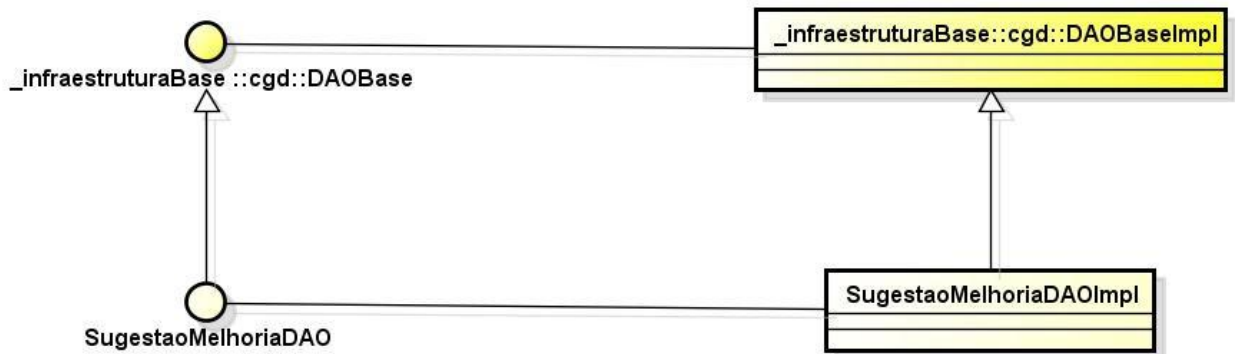


Figura 4.7 – CGD do Subsistema *controleRecomendacaoMedicao*, referente ao registro de uma Sugestão de Melhoria.

4.4 A Ferramenta *RecMed*

Nesta seção, são apresentadas algumas telas da ferramenta *RecMed*. De forma geral, *RecMed* é composta por um conjunto de funcionalidades cadastrais, que permitem o armazenamento de dados iniciais para a utilização da ferramenta, como por exemplo, cadastro das recomendações de medição, e por um conjunto de funcionalidades que apoiam a utilização e evolução das recomendações.

A parte cadastral da ferramenta está presente no subsistema *conhecimentoRecomendacaoMedicao*. Ela permite o armazenamento de dados como, por exemplo, os dados pessoais dos usuários, das organizações, dos grupos, dos aspectos tratados pelas recomendações e das recomendações de medição propriamente ditas. Vale à pena ressaltar que todo o conjunto de recomendações, com seus respectivos aspectos e grupos, definido em (BARCELLOS, 2009), é previamente armazenado na ferramenta. Por se tratarem de cadastros simples, essa parte não será detalhada nesta seção.

A parte da utilização e evolução das recomendações de medição, que está presente no subsistema *controleRecomendacaoMedicao*, permite ao usuário consultar, registrar resultados de uso (por

recomendações e por aspecto) e sugerir melhorias (baseadas ou não em um registro de resultado de uso), que serão avaliadas pelo especialista de medição. É essa parte da ferramenta que merece um maior destaque, e será apresentada a seguir.

Com o intuito de mostrar o funcionamento da ferramenta, são apresentadas as suas principais telas. Como dito nos parágrafos anteriores, as telas de cadastro foram omitidas e além disso, apenas o fluxo normal das funcionalidades (casos de uso) é apresentado. Fluxos variantes e de exceção foram omitidos, pois resultariam apenas em pequenas variações das telas apresentadas.

Na Figura 4.8 é apresentada a tela principal da ferramenta RecMed.



Figura 4.8 – Tela principal da ferramenta RecMed.

Considerando-se que as recomendações de medição estão previamente cadastradas na ferramenta, o usuário pode consultá-las. A consulta às recomendações é realizada em uma estrutura de árvore, onde o usuário seleciona o grupo, o aspecto e a recomendação que deseja consultar, como mostra a Figura 4.9.

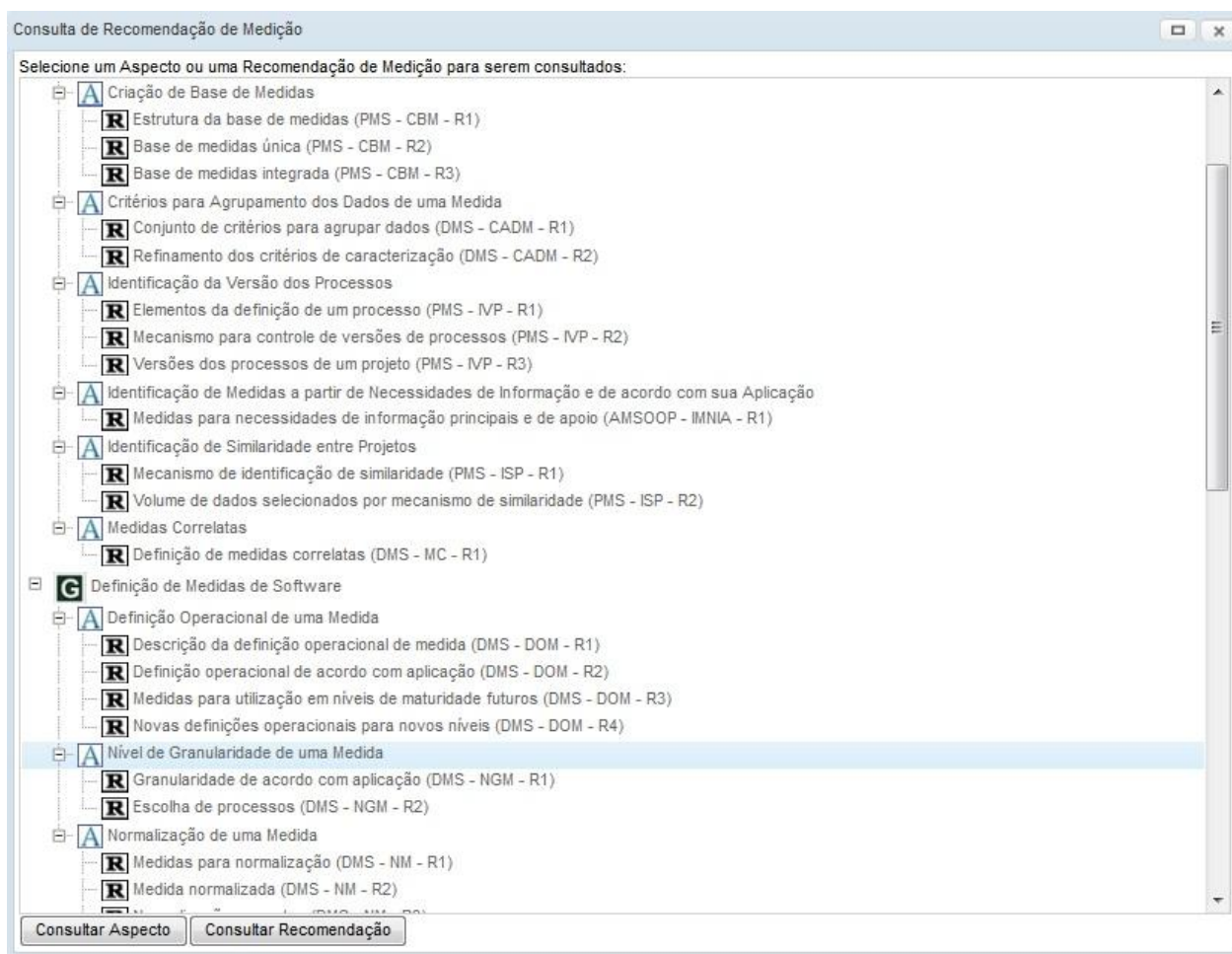


Figura 4.9 – Árvore de seleção para Consulta de Recomendação de Medição.

Clicando nos botões *Consultar Recomendação* ou *Consultar Aspecto*, o usuário pode consultar, respectivamente, uma única recomendação ou um aspecto. A diferença entre as consultas é que a consulta por recomendação apresenta os dados da recomendação selecionada, enquanto que a consulta por aspecto apresenta as informações sobre o aspecto selecionado e lista as recomendações a ele associadas, podendo estas, então, serem consultadas.

Ao consultar um aspecto são apresentados seus dados: o propósito, a fundamentação teórica e as recomendações relacionadas a esse aspecto, como mostra a Figura 4.10.

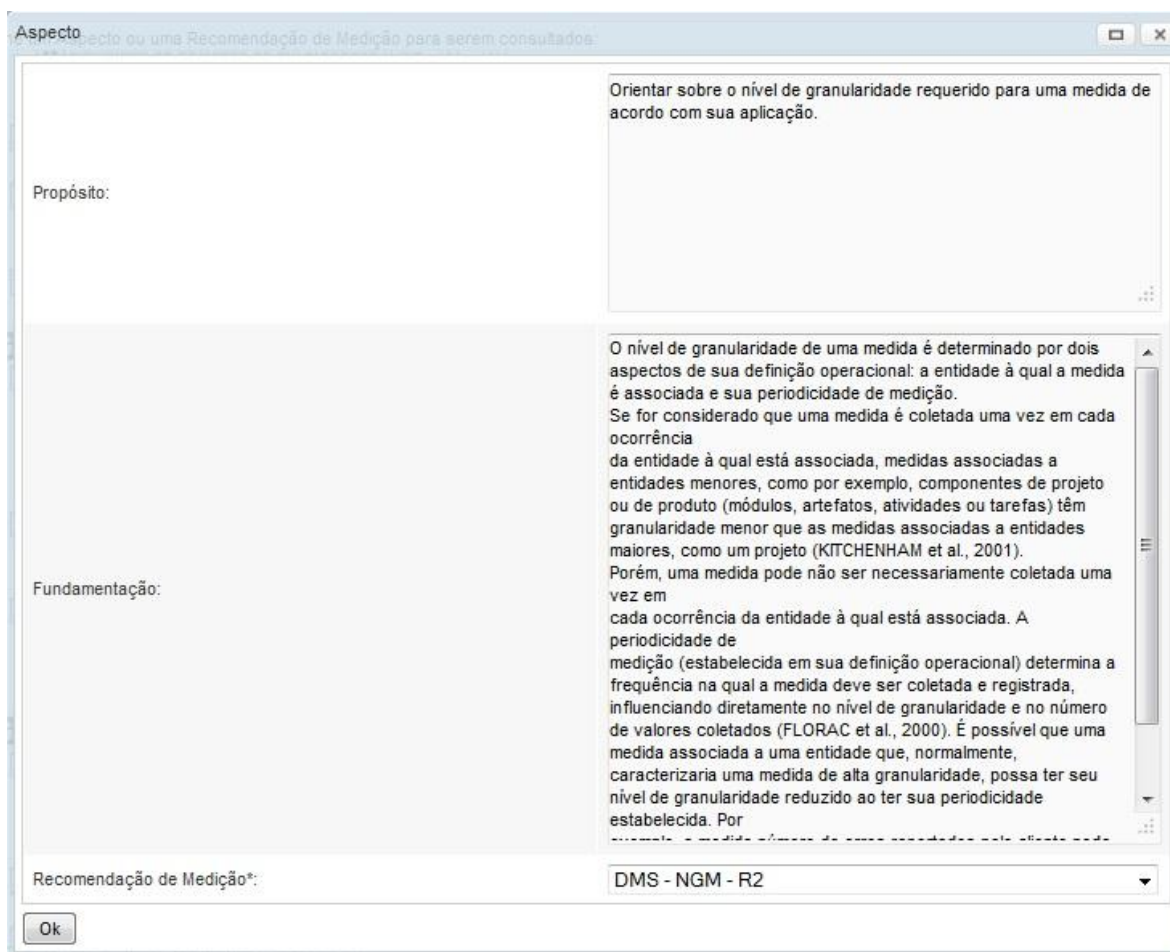


Figura 4.10 – Consulta de Aspecto.

O usuário pode, então, consultar as recomendações. Ao selecionar uma recomendação de medição, são apresentados: o especialista em medição que cadastrou a recomendação, a data de criação, a versão da recomendação, se é a versão corrente, o aspecto ao qual a recomendação pertence, a atividade de medição em que a recomendação pode ser útil, o nome, o código, o texto da recomendação propriamente dita e exemplos, se existirem (Figura 4.11).

Recomendação de Medição

| | |
|--------------------------|---|
| Especialista em Medição: | Ana Flávia Campos Leão |
| Data de Criação: | 06/03/2012 |
| Versão: | 1 |
| Versão Corrente: | Não |
| Aspecto: | Nível de Granularidade de uma Medida |
| Atividade de Medição: | Coleta de Medidas |
| Nome: | Escolha de processos |
| Código: | DMS - NGM - R2 |
| Texto: | Selecionar para o controle estatístico processos/subprocessos que sejam executados várias vezes ao longo dos projetos, para que a coleta das medidas a eles associadas seja realizada um número maior de vezes em cada projeto. |
| Exemplo: | |

Ok

Figura 4.11 – Consulta de Recomendação de Medição.

A consulta de recomendações permite que os usuários acessem o conhecimento por elas provido e o utilize em situações reais em projetos. Uma vez que tenha utilizado uma recomendação, o usuário pode fazer um registro do resultado de uso dessa recomendação. Esse registro pode ser feito para uma recomendação específica ou para um aspecto.

Para registrar os resultados de uso de recomendações, o usuário deve selecionar um grupo na estrutura de árvore, um aspecto (caso queira registrar o resultado de uso por aspecto) ou uma recomendação de medição (caso queira registrar o resultado de uso por recomendação) (Figura 4.12).

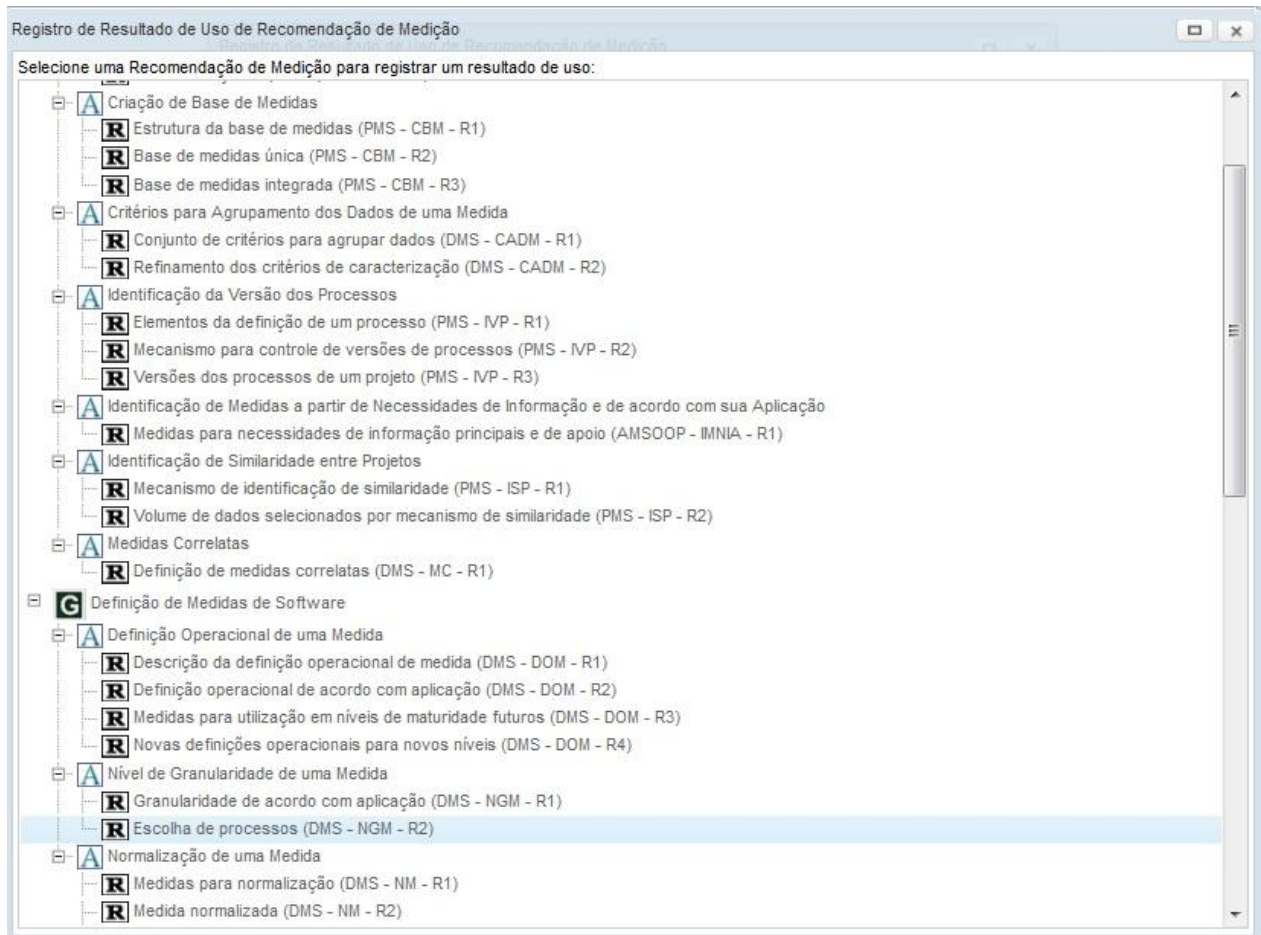


Figura 4.12 – Árvore de seleção para Registro de Resultado de Uso de Recomendação de Medição.

Para o registro de uso, o usuário deve informar a data, um nome para identificação do registro de uso, sua descrição, o contexto em que a recomendação foi utilizada (número de projetos e característica dos projetos), a sua opinião sobre a utilização da recomendação de medição (ou aspecto) (ótima, boa, regular, ruim e péssima), o nível de satisfação com os resultados da sua utilização (satisfatório, parcialmente satisfatório, insatisfatório) e o usuário que está fazendo o registro de uso (Figura 4.13).

Registro de Resultado de Uso de Recomendação de Medição

Data: 11/03/2012

Nome: Uso da recomendação "Escolha de Processos".

Descrição: A Recomendação de Medição foi útil para a escolha dos processos utilizados. O uso da recomendação levou à seleção de processos que se repetiam ao longo do desenvolvimento, o que permitiu a coleta de dados com frequência adequada para analisar o comportamento dos processos usando o controle estatístico. Como os projetos nos quais a recomendação foi utilizada eram grandes, se tivessem sido

Número de Projetos: 3

Características dos Projetos: Projetos Grandes e Complexos.

Opinião*: Boa

Nível de Satisfação*: Satisfatório

Usuário*: Taís Padrão

Salvar Cancelar

Figura 4.13 – Registro de Resultado de Uso de Recomendação de Medição.

O usuário pode, também, sugerir melhorias para uma recomendação de medição. A sugestão de melhoria pode ser baseada em um registro de resultado de uso ou não. Para sugerir uma melhoria, o usuário deve selecionar uma recomendação de medição (caso a sugestão não seja baseada em um registro de uso) ou um registro de uso (caso a sugestão seja baseada em um registro de uso (Figura 4.14).

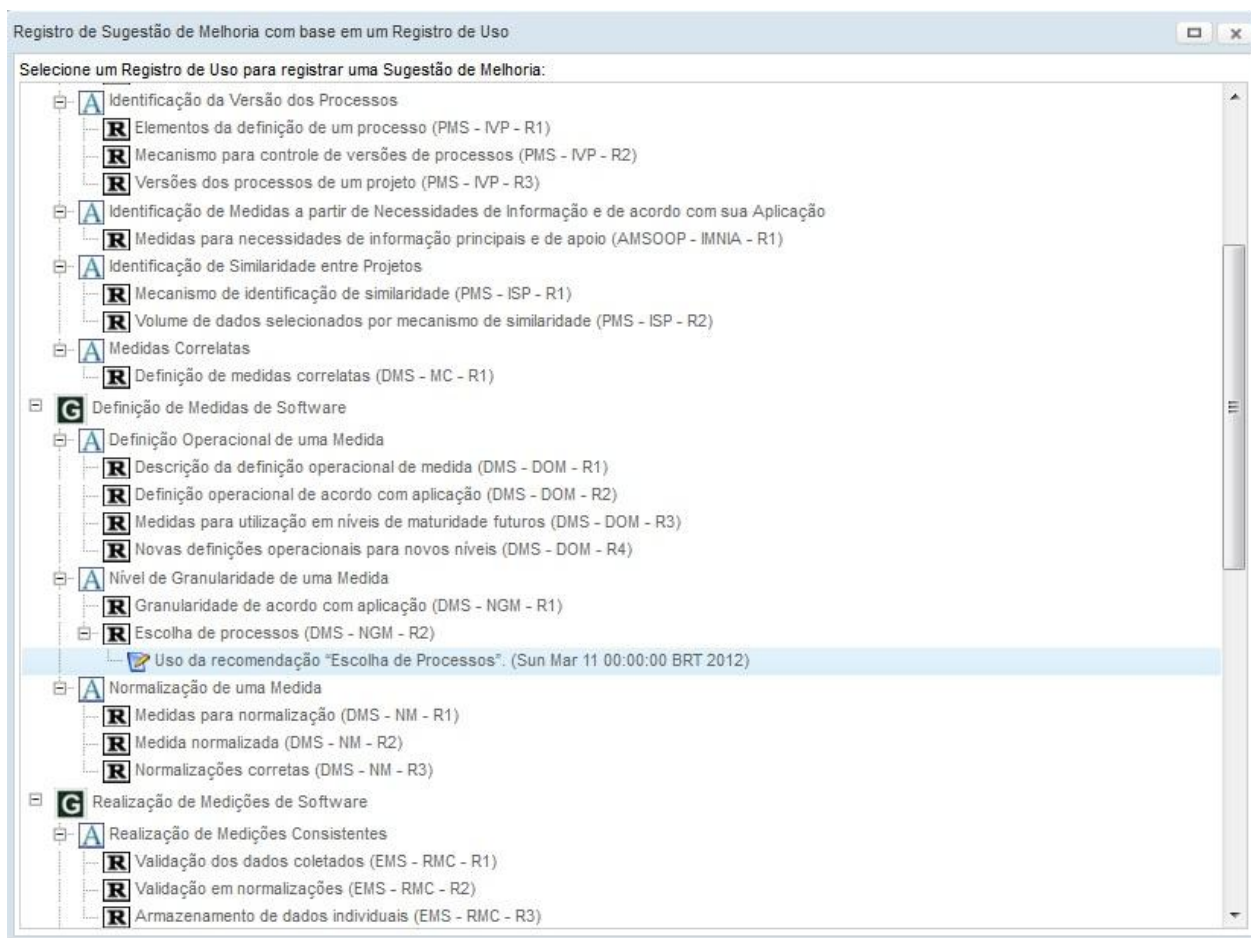


Figura 4.14 – Árvore de seleção para Registro de Sugestão de Melhoria com base em um Registro de Resultado de Uso.

Para registrar a sugestão de melhoria, o usuário deve informar a data, um nome para identificar a sugestão de melhoria, sua descrição, uma justificativa para a sugestão e o nível de criticidade (alto, médio, baixo) que ele considera que tem a sugestão. A sugestão de melhoria é cadastrada com o *status Em Aberto* (Figura 4.15).

The image shows a software window titled "Registro de Sugestão de Melhoria". It contains the following fields and values:

- Data:** 11/03/2012
- Nome:** Fornecimento de exemplos de processos.
- Descrição:** Dar exemplos de processos que poderiam ser selecionados.
- Justificativa:** Apesar da recomendação de medição ter sido muito útil, o fornecimento de exemplos facilitaria a escolha dos processos.
- Nível de Criticidade*:** Médio
- Estado da Sugestão de Melhoria*:** Em aberto

At the bottom of the window are two buttons: "Salvar" and "Cancelar".

Figura 4.15 – Registro de Sugestão de Melhoria com base em um Registro de Resultado de Uso.

As sugestões de melhoria registradas na ferramenta são analisadas por um especialista, para aceitá-las ou não. Para analisar as sugestões de melhoria cadastradas, o especialista acessa a tela apresentada na Figura 4.16.

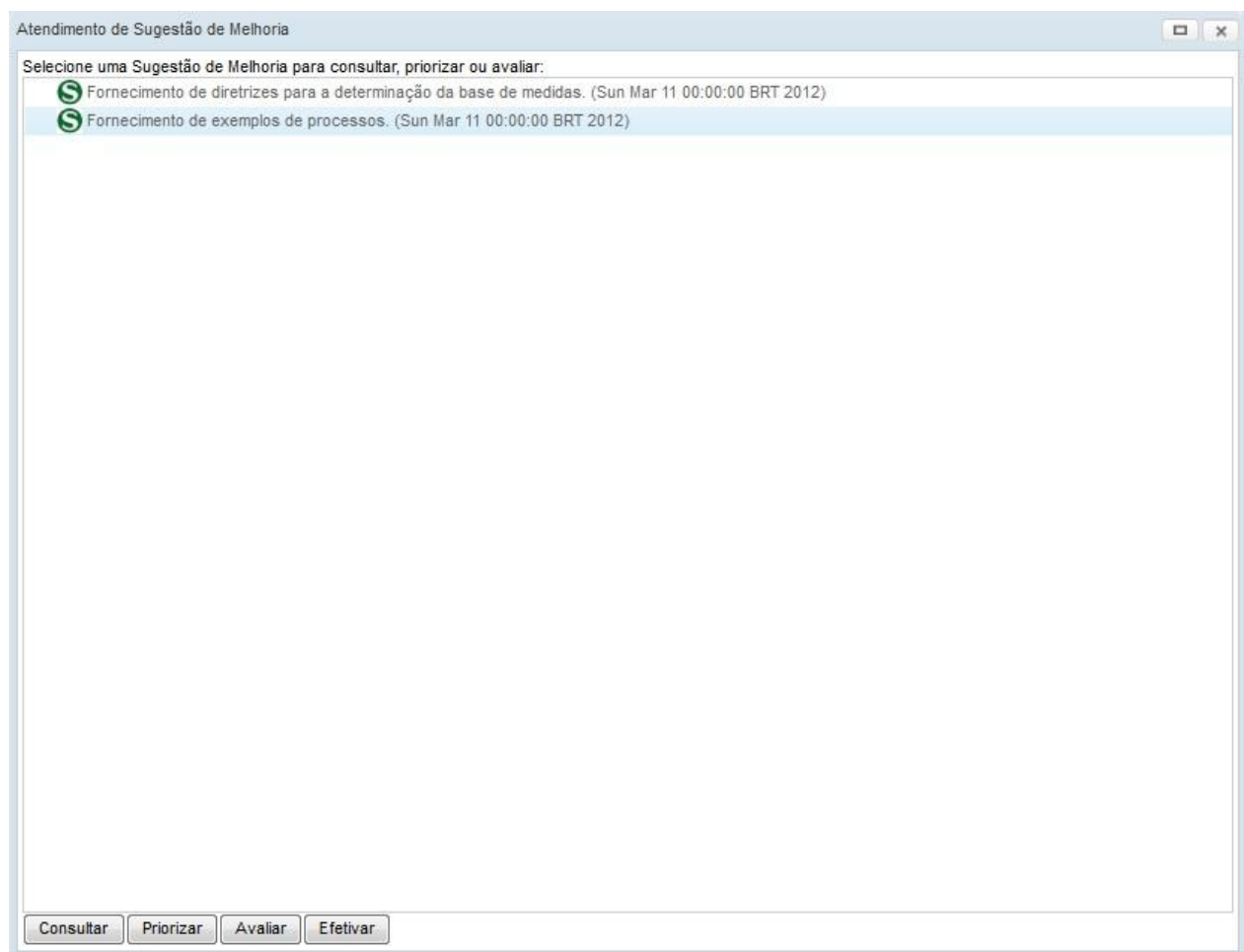


Figura 4.16– Atendimento da Sugestão de Melhoria.

Na tela da Figura 4.16, o especialista pode consultar uma sugestão (botão *Consultar*), pode solicitar que elas sejam exibidas em ordem de prioridade (botão *Priorizar*) (inicialmente as sugestões são exibidas em ordem de data de registro), pode avaliar uma sugestão de melhoria (botão *Avaliar*), indicando se ela foi *Aceita* ou *Recusada*, e pode efetivar sugestões que tenham sido aceitas (botão *Efetivar*).

A avaliação de uma sugestão de melhoria é registrada na tela mostrada na Figura 4.17. Os dados apresentados na tela dizem respeito à avaliação da sugestão “Fornecimento de exemplos de processos”. Quando o especialista recusa uma sugestão de melhoria, ele deve registrar o motivo.

Avaliação de Sugestão de Melhoria

Avaliação da Sugestão de Melhoria*: Aceita

Data:

Motivo:

Especialista em Medição*: Ana Flávia Campos Leão

Salvar Cancelar

Figura 4.17 – Avaliação da Sugestão de Melhoria.

Para efetivar as sugestões de melhoria, o especialista em medição deve selecionar a recomendação de medição relacionada à sugestão de melhoria *Aceita* e registrar uma nova versão para ela, conforme mostra a Figura 4.18

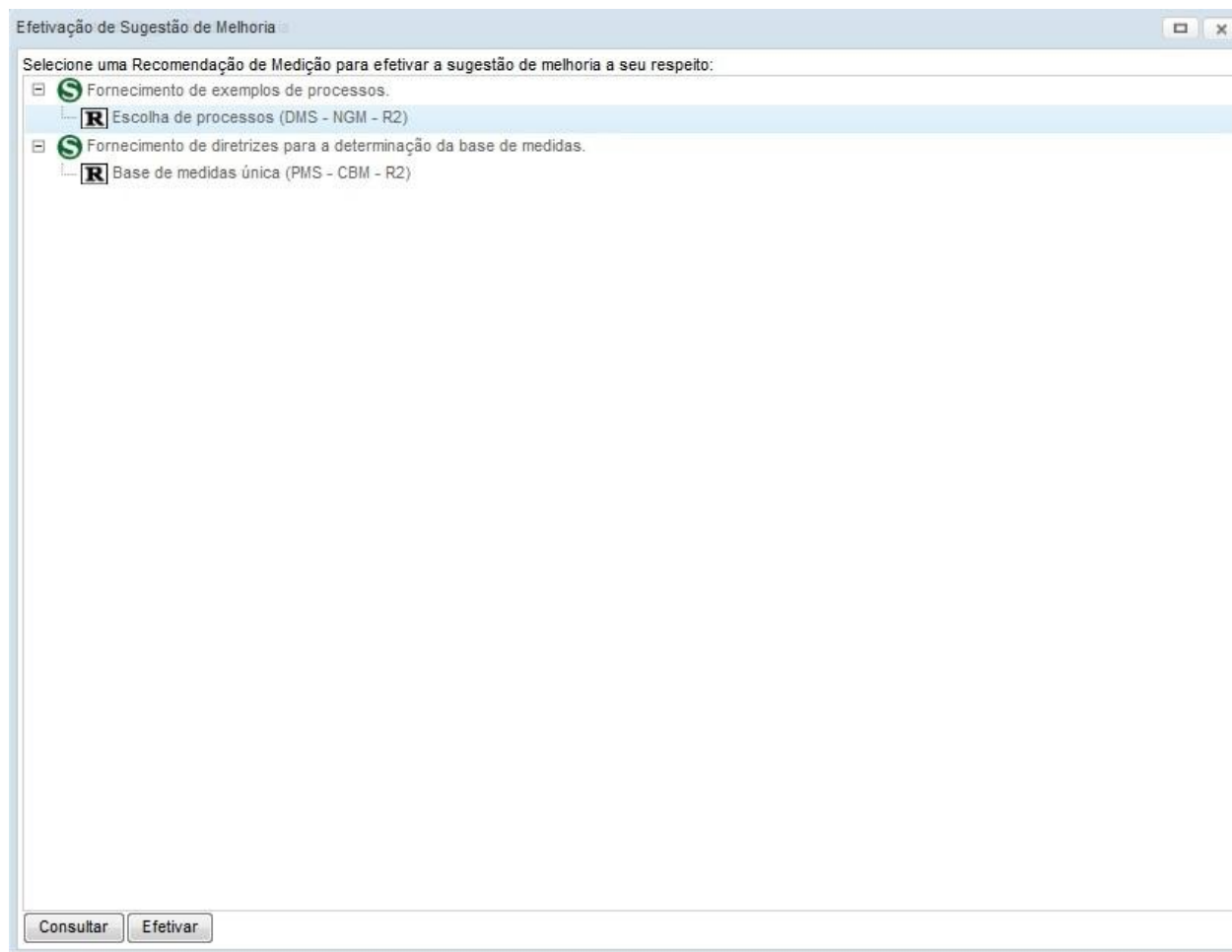


Figura 4.18 – Efetivação da Sugestão de Melhoria.

Quando o especialista registra uma nova versão para uma recomendação, contendo as alterações resultantes da sugestão de melhoria realizada, a versão anterior da recomendação tem seu atributo *Versão Corrente*, alterado para *Não*. A nova versão criada passará a ser a versão corrente da recomendação. Para registrar uma nova versão para uma recomendação, o especialista de medição deve informar a data e uma descrição das alterações feitas. Após salvar a nova recomendação de medição, a sugestão de melhoria que originou as alterações realizadas tem seu *status* alterado para *Efetivada*.

Na Figura 4.19 é apresentada a tela de registro de uma nova versão para uma recomendação de medição.

Recomendação Medição

Dados Cadastro Exemplo Nova Versão

Especialista Medicao*: Ana Flávia Campos Leão

Data de Criação: 06/03/2012

Versão: 2

Versão Corrente: Sim Não

Aspecto*: Nível de Granularidade de uma Me

Atividade Medição*: Nenhuma

Nome: Escolha de processos

Código: DMS - NGM - R2

Texto:

Selecionar para o controle estatístico processos/subprocessos que sejam executados várias vezes ao longo dos projetos, para que a coleta das medidas a eles associadas seja realizada um número maior de vezes em cada projeto.

Adicionar Exemplos: Sim Não

Salvar Cancelar

Figura 4.19 – Criação de uma Nova Versão para a Recomendação de Medição.

Capítulo 5

Considerações Finais

5.1 Conclusões

As organizações, em busca da sobrevivência em um mercado cada vez mais exigente, têm buscado uma crescente melhoria em seus processos de software. Nesse contexto, a medição de software tem papel fundamental, pois ela permite que se tenha visibilidade de alguns aspectos do produto entregue, assim como do seu processo de desenvolvimento, permitindo, assim, que se tenham resultados objetivos que possam ser usados para tomar decisões e ações corretivas apropriadas, quando necessário. Porém, para que os processos de software atinjam a alta maturidade, a simples medição não é uma prática suficiente. É necessário aliá-la ao controle estatístico de processos, buscando a obtenção de projetos estáveis e capazes.

Conforme dito anteriormente, em (BARCELLOS, 2009a) foi proposto um conjunto de recomendações para medição de software adequada ao controle estatístico de processos com o objetivo de apoiar as organizações que buscam a alta maturidade em seus processos. Porém, esse conjunto de recomendações existia apenas em um documento textual, não possuindo apoio computacional algum. Com o objetivo de prover apoio computacional a esse conjunto de recomendações, a ferramenta *RecMed* foi desenvolvida para apoiar a utilização das recomendações e obtenção de *feedback* em relação a seu uso, propiciando a realização de melhorias.

No Capítulo 1, foram identificados os objetivos estabelecidos para este trabalho. A seguir é apresentada na Tabela 5.1 a situação de cada um desses objetivos no momento de conclusão desta monografia:

Tabela 5.1 – Objetivos e sua situação na conclusão da monografia.

| Objetivo | Situação | Observação |
|--|-----------|---|
| Identificar e documentar os requisitos da ferramenta | Realizado | Foram identificados os requisitos funcionais, não funcionais e as regras de negócio de <i>RecMed</i> . As tabelas contendo essas informações encontram-se no Documento de Requisitos de <i>RecMed</i> . |

| Objetivo | Situação | Observação |
|--|---------------------------|---|
| Realizar a modelagem comportamental e estrutural da ferramenta e documentar na Especificação de Requisitos da ferramenta | Realizado | Foram elaborados diagramas de classes e de transição de estados, descritos no Capítulo 3 e detalhados no Documento de Especificação de Requisitos de <i>RecMed</i> . |
| Definir a arquitetura da ferramenta e detalhá-la em um Documento de Projeto | Realizado | A arquitetura foi definida e foram elaborados os diagramas dos componentes que a compõem, descritos no Capítulo 4 e detalhados no Documento de Projeto de <i>RecMed</i> . |
| Implementar a ferramenta | Realizado Parcialmente | A previsão é que fossem implementados todos os requisitos da ferramenta. Porém, por questão de tempo, alguns requisitos não puderam ser implementados como previstos. |

É importante ressaltar que a ferramenta possui algumas limitações. Alguns requisitos que estavam previstos para serem implementados não foram devido à falta de tempo. São eles: envio de email ao especialista de medição após uma sugestão de melhoria ter sido cadastrada e priorização das sugestões de melhoria para as recomendações, considerando o número de sugestões registradas para uma recomendação, a data de registro e a criticidade da sugestão. Além disso, algumas questões de interface precisam ser melhoradas. Essas questões estão registradas em uma lista de pendências, ainda para essa versão da ferramenta.

Pode-se dizer que a experiência adquirida com a realização desse trabalho foi muito grande. Vale à pena destacar a integração das diversas disciplinas e conceitos vistos durante toda a graduação. Para a execução deste trabalho passou-se por todas as fases do processo de desenvolvimento de software (especificação de requisitos, análise, projeto do sistema, implementação e testes). Além disso, foi muito importante a utilização de várias ferramentas e *frameworks*, até então desconhecidos para a autora deste trabalho. Não menos importante, tem-se o conhecimento obtido através da revisão de literatura acerca de medição de software e controle estatístico de processos.

Considerando as dificuldades encontradas para desenvolver este trabalho, destaca-se a necessidade de estudo do ambiente ODE, que foi utilizado como infraestrutura base para a ferramenta. Além disso, destaca-se a dificuldade de ter que estudar e aprender em um curto período de tempo algumas tecnologias e *frameworks* que foram utilizadas na construção da ferramenta. Por fim, a integração das diferentes disciplinas vistas durante o curso de Ciência da Computação foi um desafio. Na graduação, cada disciplina foi vista de forma separada, não permitindo ver de forma clara como todas as peças do desenvolvimento de um software se encaixam. Com a realização deste projeto foi possível ver e entender o funcionamento do processo de construção de um software.

5.2 Trabalhos Futuros

No final do desenvolvimento de um software, tipicamente novas necessidades são identificadas. A manutenção e a evolução de software devem ser um trabalho constante, de forma que o ciclo de vida não finalize na homologação, mas permaneça ao longo de toda a vida do software.

Sendo assim, alguns trabalhos surgirão a partir deste. Como evolução da ferramenta *RecMed*, pode-se citar a possibilidade de armazenamento de arquivos associados às recomendações de medição. Com isso, exemplos mais sofisticados, incluindo gráficos de controle (utilizados no controle estatístico de processos), poderiam ser facilmente incluídos e disponibilizados na ferramenta.

Uma outra evolução poderia ser a exportação das recomendações de medição para o Portal do Conhecimento do ODE (COELHO, 2010) e, a partir daí, uma organização específica poderia gerenciar o seu próprio conhecimento sobre medição.

Por fim, *RecMed* pode ser integrada a duas outras ferramentas:

- MeMOS (FRAUCHES, 2011), uma ferramenta de apoio à medição e monitoramento de objetivos de software. A ferramenta apoia a realização do processo de medição e as recomendações poderiam ser acessadas durante a execução das atividades.
- IABM, uma ferramenta de apoio à avaliação de bases de medida visando à utilização no controle estatístico de processos, que está sendo concluída em outro projeto de graduação. Nessa ferramenta é realizada a avaliação da adequação de itens (Plano de

Medição, estrutura da base de medidas, medidas e dados) ao controle estatístico de processos de software. Quando um item precisa de adequações, elas são identificadas e sugeridas pelo IABM. No entanto, quando um item não pode ser adequado, ele deve ser descartado. Nesse caso, as recomendações de medição poderiam ser utilizadas para fornecer o conhecimento necessário para redefinir o item (criar um novo) adequadamente.

Referências Bibliográficas

- AURUM, A., WOHLIN, C., 2005, *Engineering and Managing Software Requirements*, Springer-Verlag
- BALDASSARE, M. T., BOFFOLI, N., BRUNO, G., CAIVANO, D., 2009, "Statistically Based Process Monitoring: Lessons from the Trench", *Lecture Notes in Computer Science*, v. 5543, pp. 11-23.
- BARCELLOS, M. P., 2008, *Uma abordagem para Controle Estatístico de Processos em Organizações de Alta Maturidade*, Exame de Qualificação para o Doutorado COPPE/UFRJ, Rio de Janeiro, Brasil
- BARCELLOS, M. P., 2009a, *Estratégia para Medição de Software e Avaliação de Bases de Medidas para Controle Estatístico de Processos de Software em Organizações de Alta Maturidade*, Tese de D. Sc. PESC, COPPE/UFRJ, Rio de Janeiro, Brasil.
- BARCELLOS, M. P., 2009b, *Controle Estatístico de Processos Aplicado a Processos de Software – “Do Chão da Fábrica” para as Organizações de Software*, Engenharia de Software Magazine, Ano 1, Edição 11, pág.56-61.
- BARCELLOS, M. P., 2010, *Medição de Software: Um importante pilar da melhoria de processos de software*, Engenharia de Software Magazine, Ano 2, Edição 24.
- BASILI, V. R., ROMBACH, H. D., 1994, *Measurement*, Encyclopedia of Software Engineering, John Wiley & Sons, v. 1.

BASILI, V. R., GREEN, S., 1994, "Software Process Evolution at the SEL", *IEEE Software*, v.11, n. 4, pp. 58-56.

BASS, L., BELADY, L., BROWN, A., FREEMAN, P., ISENSEE, S., KAZMAN, R., KRASNER, H., MUSA, J., PFLEEGER, S., VREDENBURG, K., WASSERMAN, T., 1999, *Constructing Superior Software*, Software Quality Institute Series, Macmillan Technical Publishing.

BENNEYAN, J. C., LLOYD, R., PSELK, P. E., 2003, "Statistical Process Control as a Tool for Research and Healthcare Improvement", *British Medical Journal - Quality and Safety Health Care*, v. 12, pp. 458-464.

CHRISSIS, M. B., KONRAD, M., SHRUM, S., 2006, *CMMI (Second Edition): Guidelines for Process Integration and Product Improvement*, Addison-Wesley.

COELHO, Alexandre Guilherme Nicco. *Uma Infraestrutura de Gerência de Conhecimento em Organizações de Software Aplicada à Gestão de Riscos*. Dissertação de Mestrado (Mestrado em Informática), Universidade Federal do Espírito Santo, Vitória - Brasil, Agosto 2010

IEEE, 1998, Std 1061 – *IEEE Standard for a Software Quality Metrics Methodology*.

ISO/IEC, 2001, ISO/IEC 9126 -1 - *Software Engineering - Product Quality - Part 1: Quality Model*, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

ISO/IEC, 2002, ISO/IEC 15939 – *2002 (E) Software Engineering – Software Measurement Process*, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

ISO/IEC, 2003, ISO/IEC 15504-2 - *Information Technology – Software Process Assessment*, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

ISO/IEC, 2007, ISO/IEC 15939 (E) *Software Engineering - Software Measurement Process*, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

ISO/IEC, 2008, ISO/IEC 12207:2008 - *Systems and Software Engineering - Software Life Cycle Process* International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

FENTON, N. E., NEIL, M., 1999, "*Software Metrics: Success, Failures and New Directions*", *Journal of Systems and Software*, v. 47, pp. 149-157.

FLORAC, W. A., CARLETON, A. D., 1999, *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, Addison Wesley.

KITCHENHAM, B., HUGHES, R. T., LINKMAN, S. G., 2001, "*Modeling Software Measurement Data*", *IEEE Transactions on Software Engineering*, v. 27, n. 9, pp. 788-804.

KITCHENHAM, B., JEFFERY, D. R., CONNAUGHTON, C., 2007, "*Misleading Metrics and Unsound Analyses*", *IEEE Software*, v. 24, n. 2, pp. 73 - 78.

LANTZY, M. A., 1992, "Application of Statistical Process Control to the Software Process", In: *Proceedings of the 9th Washington Ada Symposium on Empowering Software Users and Developers*, ACM Press, pp. 113-123.

McGARRY, J., CARD, D., JONES, C., LAYMAN, B., CLARK, E., DEAN, J., HALL, F., 2002, *Practical Software Measurement: Objective Information for Decision Makers*, Addison Wesley, Boston, USA.

NIESSINK, F., VLIET, H., 2001, "Measurement Program Success Factors Revisited", *Information and Software Technology*, v. 43, n. 10, pp. 617-628.

SARGUT, K. U., DEMIRORS, O., 2006, "Utilization of Statistical Process Control (SPC) in Emergent Software Organizations: Pitfalls and Suggestions", *Software Quality Journal*, v. 14, n. 5, pp. 135-157.

SEI – Software Engineering Institute, *CMMI for Development version v. 1.3*, Camagie Mellon University, USA, 2010.

SHEWART, W. A., 1980, *The Economic Control of Quality of Manufactured Product*, D. Van Nostrand Company, New York, 1931, reimpresso por ASQC Quality Press, Milwaukee, Wisconsin.

SOFTEX, 2011, *MPS.BR: Melhoria de Processo do Software Brasileiro – Guia Geral: 2009*, Disponível em: <http://www.softex.br/mpsbr>.

WANG, Q., LI, M., 2005, "Measuring and Improving Software Process in China", In: *Proceedings of International Symposium on Empirical Software Engineering - ISESE 2005*, Hoosa Head, Australia, pp. 183-192.

WHEELER, D. J., CHAMBERS, D. S., 1992, *Understanding Statistical Process Control*, 2nd ed., Knoxville - SPC Press.

WHEELER, D. J., PFADT, A., 1995, "Using Statistical Process Control to Make Data-Based Clinical Decisions", *Journal of Applied Behaviors Analysis*, v. 3, pp. 349-370.

WHEELER, D. J., POLING, R. S., 1998, *Building Continual Improvement: A Guide for Business*, SPC Press.