

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

Ludimila Nicoli Guimarães

Resolve: Uma Ferramenta de Apoio à Resolução de Problemas em Projetos de Software em ODE

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

16 de Março de 2012

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

Ludimila Nicoli Guimarães

Resolve: Uma Ferramenta de Apoio à Resolução de Problemas em Projetos de Software em ODE

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.

Prof. Júlio Cesar Nardi, M. Sc.

Prof. Edilson Luiz Nascimento, M.Sc

Vitória, 16 de Março de 2012

A meus pais, José e Maria da Glória.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, que nunca mediram esforços para que eu pudesse fazer uma faculdade. Aos meus irmãos: Luciano, Bethânia e Rodrigo que sempre me apoiaram. Aos meus pequenos sobrinhos: Hugo, Henrique e Otávio que até mesmo antes de nascer já me traziam alegrias.

A todos os colegas do Nemo: Cadu, Julião, Viola, Vitor, Vinícius, Rafael, Alexandre, Bruno e em especial a Ana Flávia, que desde que conheci tem sido uma irmã para mim.

A minha orientadora Monalessa, que não só me orientou, mas incentivou e deu apoio pra que eu pudesse realizar este trabalho.

A Deus, que é o Senhor e dono de tudo.

RESUMO

Quando organizações se deparam com problemas, poucas sabem o que deve ser feito para resolvê-los. Isso acontece por falta de um processo definido ou de uma ferramenta que possa auxiliar essas organizações. Em um mundo cada vez mais competitivo, gastar maior parte do tempo de um projeto resolvendo problemas é totalmente inviável e contribui para o comprometimento da qualidade do produto.

O uso de uma ferramenta de apoio ao processo de resolução de problemas pode ajudar as organizações a registrarem seus problemas, causas e soluções, alimentando uma base histórica que constitui conhecimento relevante aos membros dos projetos. Diante desse contexto, este trabalho propõe um processo para a resolução de problemas e uma ferramenta de apoio, desenvolvida no contexto do ambiente ODE (*Ontology-based software Development Environment*).

Palavras-chave: Resolução de Problemas, Problemas, Causas, Soluções e Desenvolvimento Web.

SUMÁRIO

Capítulo 1 - Introdução	8
1.1 Introdução	8
1.2 Objetivos	9
1.3 Histórico de Desenvolvimento do Trabalho.....	10
1.4 Organização do Texto	11
Capítulo 2 - Resolução de Problemas	12
2.1 Introdução.....	12
2.2 Resolução de Problemas	12
2.3 Abordagens para Resolução de Problemas	14
2.4 O Ambiente ODE	16
Capítulo 3 - Especificação de Requisitos e Análise da Ferramenta Resolve	18
3.1 Introdução.....	18
3.2 Processo de Resolução de Problemas	18
3.3 Resolve: Propósito do Sistema e Descrição do Minimundo	21
3.4 Subsistemas da Ferramenta Resolve	23
3.5 Casos de Uso da Ferramenta <i>Resolve</i>	24
3.6 Diagramas de Classes da Ferramenta <i>Resolve</i>	27
Capítulo 4 - Projeto e Implementação da Ferramenta Resolve.....	32
4.1 Introdução.....	32
4.2 Arquitetura de Software.....	32
4.3.1 Subsistema resolucaoProblema	34
4.3.1.1 Camada Lógica do Negócio.....	35
4.3.1.1.1 Componente Domínio do Problema	35
4.3.1.1.2 – Componente de Gerência de Tarefas	37
4.3.1.2– Camada de Interface com o Usuário.....	37
4.3.1.2 – Camada de Gerência de Dados	38
Capítulo 5 - Considerações Finais	47
5.1 Conclusões.....	47
5.2 Trabalhos Futuros	48
Referências Bibliográficas	50

Capítulo 1

Introdução

1.1 Introdução

Em um ambiente cada vez mais competitivo, a qualidade dos produtos e serviços oferecidos pelas organizações de TI (Tecnologia de Informação) é fator determinante de sucesso. A complexidade dos produtos de software atuais, principalmente devido à crescente e incessante evolução da tecnologia, pode fazer com que o número e a complexidade dos problemas encontrados também cresçam.

De acordo com uma pesquisa realizada pela consultoria norte americana PM Solutions (PM Solutions, 2011), mais de um terço dos projetos de TI apresentam algum tipo de problema, o que representa desperdício de recurso financeiro, tempo e perda de credibilidade dos profissionais da área de TI. Foram realizadas 163 entrevistas com diversas empresas e calculou-se que 37% dos projetos de TI enfrentam algum tipo de problema e 12% deles falham, causando um prejuízo de aproximadamente US\$ 74 milhões ao ano.

Mas afinal, o que vem a ser um problema? Um problema é a diferença entre a situação atual e a situação ideal ou a situação atual e o objetivo (HOSOTANI, 1992 *apud* PARIS 2003). Ou, segundo LORIGGIO (1996) problema é uma situação inadequada, causada por desvio em relação aos objetivos organizacionais.

Resolver problemas adequadamente favorece a criação de produtos com maior qualidade, menor desperdício e, conseqüentemente, mais competitivos diante de um mercado cada vez mais exigente. Porém, o ideal não é somente resolver problemas encontrados, mas adquirir uma maturidade para que problemas frequentes sejam resolvidos de forma mais fácil do que problemas vistos pela primeira vez.

Embora a resolução de problemas seja algo tão importante no contexto dos projetos, poucas empresas possuem uma técnica e, muito menos, uma ferramenta que possa auxiliar esse processo. Na maioria das vezes, os gerentes partem para a chamada resolução antes mesmo de refletir sobre a natureza real de seu problema. Essa é uma situação comum na área de gerência de projetos de software, onde a resolução de problemas muitas vezes é realizada por “tentativa e erro” (THOMAZ, 2005 *apud* GONÇALVES 2008).

Buscando produzir melhores produtos e aumentar sua competitividade, muitas organizações de software têm se interessado por normas e modelos que apoiam a melhoria de processos, tais como: MR MPS.BR (SOFTTEX, 2011), CMMI (*Capability Maturity Model*

Integration) (SEI, 2010), ISO/IEC 12207 e PMBOK (PMI, 2008). Todos esses modelos incluem resolução de problemas em seus padrões, seja explícita ou implicitamente.

Nesse contexto, que envolve a necessidade de lidar com problemas ao longo de projetos e de reutilizar soluções já adotadas de forma bem sucedida, este trabalho propõe a ferramenta *Resolve*, para fornecer apoio computacional para organizações de software conduzirem o processo de resolução de problemas em seus projetos. Além de apoiar o processo de resolução de problemas, o apoio computacional visa também auxiliar no armazenamento do conhecimento sobre eles e suas soluções, para que, caso voltem a acontecer, situações anteriores similares possam servir de ajuda.

A ferramenta encontra-se no contexto do Ambiente ODE (FALBO *et al.*, 2005), um ambiente de desenvolvimento de software centrado em processo e baseado em ontologias desenvolvido no contexto do Projeto ODE, o qual é uma das iniciativas do Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO), do Departamento de Informática (DI) da Universidade Federal do Espírito Santo (UFES).

O ambiente foi originalmente desenvolvido para plataforma *desktop* e vem sendo migrado para plataforma *web*, que é chamado de ODE-Web. Até este trabalho, ODE não possuía ferramenta de apoio à resolução de problemas. Sendo assim, decidiu-se pelo desenvolvimento da ferramenta no contexto de ODE, mas que também possa ser utilizada independente dele.

1.2 Objetivos

O objetivo geral desse trabalho é definir um processo de resolução de problemas e implementar uma ferramenta de apoio para ele no ambiente ODE-Web.

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

- i) Definir um processo para resolução de problemas, ou seja, um passo a passo de como se deve fazer para tratar um problema no contexto de projetos de software;
- ii) Identificar e documentar os requisitos da ferramenta de apoio ao processo;
- iii) Realizar a modelagem comportamental e estrutural da ferramenta e documentar na Especificação de Requisitos da ferramenta;
- iv) Definir a arquitetura e detalhá-la em um Documento de Projeto;
- v) Implementar a versão inicial da ferramenta.

1.3 Histórico de Desenvolvimento do Trabalho

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

- a) *Aquisição de conhecimento sobre o tema:* O trabalho teve início com uma revisão bibliográfica sobre alguns assuntos, tais como: análise de problemas e tomadas de decisão, modelos para estruturar e avaliar alternativas de decisão em TI e processos e modelos para resolução de problemas.
- b) *Elaboração do Processo de Resolução de Problemas:* Nesta etapa foi definido um processo para resolução de problemas.
- c) *Elaboração da Documentação da Ferramenta:* Nessa etapa foi elaborada a documentação da ferramenta. Foi criado o documento de requisitos, contendo o propósito do sistema, uma descrição do minimundo, requisitos funcionais, não funcionais e regras de negócio. A seguir, foi feita a especificação de requisitos, contendo a identificação de subsistemas, os modelos detalhados de casos de uso, modelo estrutural e dinâmico e um glossário do projeto. Por último, o documento de projeto, que leva em consideração os aspectos tecnológicos e contém a base para a implementação do sistema, foi elaborado.
- d) *Estudo do ambiente ODE:* Em paralelo à elaboração da documentação da ferramenta, teve início a fase de ambientação ao Ambiente ODE. Foram estudados aspectos da literatura relacionados a ferramentas CASE e ambientes de desenvolvimento de software, bem como sobre o ambiente ODE. Nesta etapa também foram realizados treinamentos e estudos sobre as tecnologias envolvidas no projeto, entre elas: linguagem Java, desenvolvimento Web, arquitetura em camadas e infraestrutura do ODE.
- e) *Implementação e Testes da Ferramenta:* Tendo em mãos a documentação necessária deu-se início a implementação do sistema, a qual foi acompanhada por testes para verificar erros de implementação e se a ferramenta desenvolvida atendia os requisitos definidos.
- f) *Organização da Monografia:* Finalmente, a escrita da monografia foi conduzida, paralelamente ao término da implementação da ferramenta.

1.4 Organização do Texto

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

Capítulo 2 – Resolução de Problemas: apresenta uma fundamentação teórica sobre resolução de problemas: o que é, como funciona, quais as principais abordagens encontradas na literatura e como alguns modelos de melhoria de processo tratam a resolução de problemas.

Capítulo 3 – Especificação de Requisitos e Análise do Sistema: apresenta o processo de resolução de problemas proposto, a descrição do minimundo contemplado pela ferramenta, seus casos de uso e modelos de classe.

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

Capítulo 5 – Considerações Finais: Apresenta a 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 desse.

Capítulo 2

Resolução de Problemas

2.1 Introdução

Neste capítulo serão apresentados os principais conceitos relacionados aos temas envolvidos neste trabalho e, para isso, ele está assim organizado: a Seção 2.2 apresenta uma visão geral sobre a resolução de problemas, a Seção 2.3 apresenta algumas abordagens propostas para a resolução de problemas e a Seção 2.4 apresenta o ambiente ODE.

2.2 Resolução de Problemas

De acordo com uma pesquisa realizada por Macoratti (MACORATTI, 2010), problemas ocorridos em projetos de TI levam a um estouro de, em média, mais de 189% do orçamento e mais de 222% do cronograma previstos para os projetos. Ainda de acordo com a pesquisa, os principais fatores que contribuem para os problemas são: falta de informações dos usuários, requisitos/especificações incompletos ou ausência de gerência de mudanças e especificações.

A não aderência ao cronograma e orçamento dos projetos ocorre, basicamente, devido a problemas encontrados e não resolvidos ou resolvidos de forma ineficaz. Embora possa parecer trivial, identificar adequadamente um problema não é uma tarefa fácil, pois pode envolver ideias de diferentes pessoas, cenário comum nas empresas de software. MORITA *et al.* (2003) afirmam que, normalmente, a identificação ou a definição do problema não é imediata, pois o número de variáveis e de alternativas envolvidas nesse contexto, usualmente, é grande.

Diversos são os problemas que ocorrem no contexto de projetos. A Figura 2.1 apresenta os resultados de um estudo realizado pelos *chapters* brasileiros do PMI (Project Management Institute) em 2008, que visou à identificação dos principais problemas encontrados nas organizações.

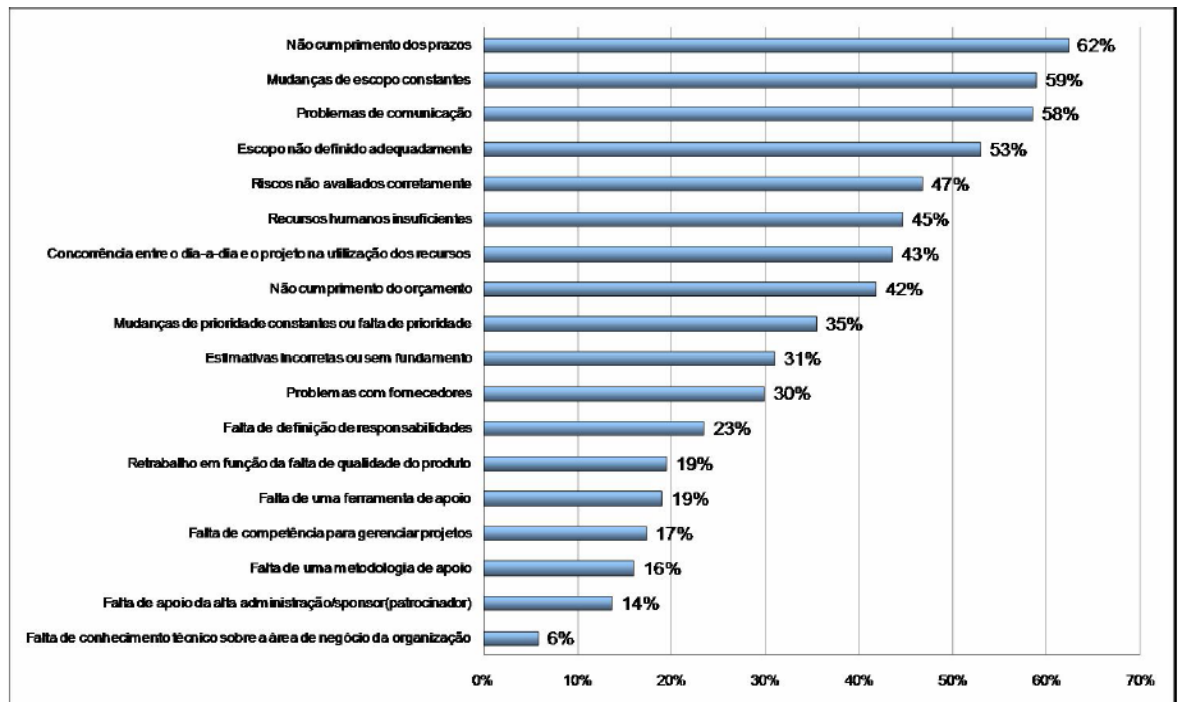


Figura 2.1 - Principais problemas encontrados nas organizações.

Fonte: PROJECT MANAGEMENT INSTITUTE – Chapters Brasileiros (2008).

E, o que vem a ser um problema? Segundo (LORIGGIO, 2002, p.35) um problema pode ser visto como:

- a) Desejar algo e não saber como obter;
- b) Necessidade não satisfeita;
- c) Estado de incerteza ou caos;
- d) Comportamento ou situação indesejável presente ou futura;
- e) Resultado que necessite de correção;
- f) Diferença entre condições atuais e condições desejadas;
- g) Obstáculo ao alcance de metas;
- h) Ameaça à sobrevivência.

Resumidamente, problema pode ser, então, definido como a diferença entre o resultado alcançado e o que se esperava alcançar (GONÇALVES, 2008).

Muitas vezes, para solucionar um problema, são cometidos muitos erros, como atacar o efeito sem descobrir a causa, encontrar soluções antes de conhecer o problema ou, simplesmente apostar na experiência. Porém, como afirma GONÇALVES (2008) a solução de problemas utilizando-se unicamente a experiência pode ser inviável. Problemas mais complexos exigem uma análise aprofundada através do uso de metodologias e técnicas para que se obtenha um bom resultado. Muitas vezes, o responsável pela tomada de decisão

acaba incorporando seus valores intrínsecos, utilizando, de forma muitas vezes inconsciente, todos os recursos pessoais para a busca da solução. Essa utilização se dará, provavelmente, de forma vaga e imprecisa, muitas vezes, excluindo variáveis relevantes para a decisão.

Assim, é comum as organizações se depararem com algumas armadilhas durante a análise e resolução das causas dos problemas, tais como (GRIMALD *et al.*, 1994 *apud* GONÇALVES, 2008):

- *Não consultar dados históricos*: não levar em consideração problemas similares já ocorridos no passado, que são fonte de informação.
- *Concluir por intuição*: ir direito à solução do problema sem analisar os ângulos da questão ou explorar outras alternativas.
- *Decidir pelo mais fácil*: desprezar dados e fatos fundamentais, por pressa ou dificuldade em obtê-los.
- *Dimensionar mal o problema*: às vezes, a solução está em esfera superior de decisão, não sendo de competência do grupo.
- *Contentar-se com uma solução*: insistir na solução encontrada, desprezando objeções e dificuldades.
- *Isolar-se com o problema*: não consultar pessoas-chave, nem as que serão responsáveis pela ação e nem as que serão afetadas pela solução.
- *Desprezar os detalhes*: encontrar solução sem aprofundar a sua viabilização.
- *Utilizar técnicas inadequadas*: escolher a técnica mais fácil, porém, menos adequada para o tipo de problema.
- *Não acompanhar os resultados obtidos*: após a realização das ações, não é realizado um acompanhamento efetivo para analisar os efeitos gerados.

A utilização de metodologias e ferramentas de apoio à resolução de problemas pode ajudar os gerentes no sentido de que eles evitem essas armadilhas.

Nas últimas décadas, alguns métodos de apoio à resolução de problemas foram propostos (PINHEIRO, 2005). Na próxima seção, são apresentadas duas abordagens.

2.3 Abordagens para Resolução de Problemas

A seguir são apresentadas duas abordagens de resolução de problemas: MASP (**Metodologia para Análise e Solução de Problemas**) e Método 8D (**Método das 8 Disciplinas**), que são abordagens muito comuns para resolver problemas em organizações.

Abordagem MASP - Metodologia para Análise e Solução de Problemas

A abordagem MASP consiste em uma sequência de etapas que levam a um planejamento participativo para a melhoria da qualidade de um produto ou serviço de determinado setor em uma organização. MASP se baseia na obtenção de dados que justifiquem ou comprovem fatos previamente levantados e que comprovadamente causem problemas e é composto por oito etapas, descritas a seguir (ANDO *et al.*, 1994 *apud* ROSSATO,1996).

- 1) *Identificação do problema*: consiste na identificação do problema a ser tratado, análise se ele é frequente ou se é a primeira vez que ocorre na organização, análise do impacto provocado, seleção das causas principais e alocação das pessoas responsáveis por tratá-lo.
- 2) *Observação*: nesta etapa as características do problema são conhecidas analisando-se dados como tempo, espaço e pessoas envolvidas no contexto da ocorrência do problema.
- 3) *Análise*: consiste na identificação das causas do problema, escolha e análise das mais prováveis.
- 4) *Plano de ação*: consiste na elaboração da estratégia que será adotada para resolver o problema e planejamento de como a estratégia será realizada (quem, quando, onde e como).
- 5) *Ação*: consiste na divulgação e execução do plano de ação.
- 6) *Verificação*: consiste em verificar se o problema foi ou não resolvido e quais foram os efeitos provocados pela ação tomada.
- 7) *Padronização*: consiste na incorporação das alterações realizadas aos padrões da organização, a fim de que ações preventivas se tornem prática comum e o problema não mais se repita.
- 8) *Conclusão*: consiste na apresentação gráfica dos resultados obtidos e no planejamento de ação para problemas remanescentes ao que foi resolvido.

Método das 8 disciplinas - 8D

Foi idealizado pela empresa Ford Motors e consiste em uma sequência de fases que deve ser seguida a partir do momento que o problema é identificado. Deve ser usado quando a causa do problema for desconhecida e a resolução do problema usando o

método 8D exige que haja formação de uma equipe. As etapas do método 8D são descritas a seguir:

- a) *Criação da Equipe*: consiste em escolher um pequeno grupo de pessoas com conhecimento sobre o processo e o produto, atribuir tempo, responsabilidades e conhecimentos técnicos necessários para a resolução do problema e implementação de ações corretivas. O grupo deverá ter um líder.
- b) *Descrição do problema*: consiste em descrever o problema, respondendo perguntas como: O quê? Onde? Quando? Quantos? Qual a importância?
- c) *Implementação e verificação*: consiste em implementar e verificar as ações intermediárias e de contenção até que o problema seja realmente resolvido.
- d) *Definição da causa raiz*: consiste em definir as causas possíveis do problema, isolar a(s) causa(s) raiz, confrontando cada causa possível com a descrição do problema e com dados históricos.
- e) *Ações corretivas*: consiste em escolher as ações corretivas permanentes e, através de testes preliminares (ditos pré-produtivos), confirmar que as ações corretivas selecionadas vão resolver o problema e não vão causar efeitos colaterais.
- f) *Implementação das ações corretivas*: consiste em definir e planejar a implementação das ações corretivas permanentes selecionadas e definir sistemas de controle para monitorar os efeitos a longo prazo.
- g) *Prevenção*: consiste em prevenir a reincidência do problema.
- h) *Congratulações*: consiste em congratular a equipe envolvida na resolução do problema.

2.4 O Ambiente ODE

O Ambiente ODE (*Ontology-based software Development Environment*) (FALBO *et al.*, 2003) é um ambiente de Desenvolvimento de Software (ADS) centrado em processo e baseado em ontologias. Esse ambiente é desenvolvido no NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias), do Departamento de Informática da UFES.

A equipe do Projeto ODE é composta por professores, alunos de doutorado, mestrado e graduação da UFES. Recentemente, professores do IFES (Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo), que foram membros do Projeto ODE quando alunos da UFES, retornaram à equipe com alguns de seus alunos de graduação.

Algumas características de ODE que merecem destaque são: a uniformidade de conceitos provida pelas ontologias, que facilita a integração, deixa o ambiente mais

homogêneo e torna mais efetiva a comunicação entre pessoas e entre ferramentas; a forte base em conhecimento, que permite que o ambiente ofereça um apoio especializado ao usuário na realização de suas tarefas e possibilita que as informações geradas mantenham-se interligadas e consistentes ao longo de todo o processo; e o foco em ferramentas gerenciais, uma vez que a gerência é uma área de grande importância e ainda carente em termos de ferramentas.

O ODE constitui um conjunto de ferramentas baseadas em áreas da engenharia de software como, apoio à definição de processos (BERTOLLO, SEGRINI *et al.*, 2006), de acompanhamento de projetos (MORO *et al.*, 2005), de gerência de requisitos (MARTINS *et al.*, 2006), de gerência de riscos (FALBO *et al.*, 2004) e de gerência de conhecimento (NATALI *et al.*, 2003).

ODE foi originalmente desenvolvido em plataforma *desktop*, porém com o crescimento do ambiente, as mudanças tecnológicas e a necessidades de organizações clientes, em 2009 teve início a migração do ODE para a plataforma *web*.

A migração vem sendo realizada gradativamente. Algumas ferramentas consideradas centrais, como a de apoio à definição de processos e a de apoio à caracterização foram selecionadas para serem as primeiras a serem migradas. Além das ferramentas que serão migradas, novas ferramentas estão sendo desenvolvidas para o ODE, dentre elas, a ferramenta *Resolve*, proposta neste trabalho.

Cabe ressaltar que uma grande diferença da versão de ODE para *web* em relação à versão para *desktop* é que na versão atual (Web) as ferramentas podem ser utilizadas tanto integradas ao ODE quanto isoladas.

Capítulo 3

Especificação de Requisitos e Análise da Ferramenta

Resolve

3.1 Introdução

Sistemas de software são reconhecidamente importantes ativos estratégicos para diversas organizações. Uma vez que tais sistemas, em especial os sistemas de informação, têm um papel vital no apoio aos processos de negócio das organizações, é fundamental que os sistemas funcionem de acordo com os requisitos estabelecidos. Nesse contexto, uma importante tarefa no desenvolvimento de software é a identificação e o entendimento dos requisitos dos negócios que os sistemas vão apoiar (AURUM; WOHLIN, 2005 *apud* FALBO 2011).

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; WOHLIN, 2005 *apud*, FALBO 2011).

Este capítulo aborda os resultados da Engenharia de Requisitos da ferramenta *Resolve* e apresenta o processo que a ferramenta apoia. Na seção 3.2 é apresentado o processo de resolução de problemas definido neste trabalho; na seção 3.3 é apresentado o objetivo da ferramenta e a descrição do minimundo; na seção 3.4 são apresentados os subsistemas de *Resolve*; na seção 3.5 são apresentados diagramas de casos de uso e na seção 3.6 são apresentados os modelos de classes.

3.2 Processo de Resolução de Problemas

O processo de resolução de problemas proposto é composto por oito atividades, como mostra a Figura 3.1. As atividades são descritas após a figura.

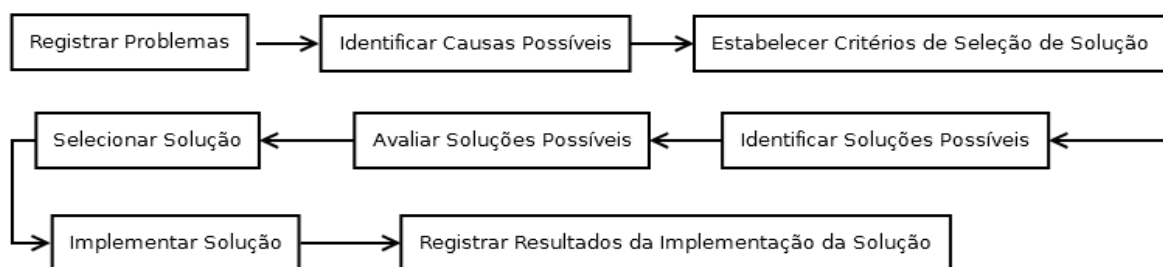


Figura 3.1 – Atividades do processo de Resolução de Problemas.

Atividade 1: Registrar Problemas

Nesta atividade, problemas identificados ao longo dos projetos são registrados. O registro de um problema deve incluir informações para caracterizá-lo como, por exemplo, o projeto ou a fase em que foi detectado e sua categoria. Exemplos de categorias de problemas podem ser: problemas tecnológicos, problemas nos requisitos e problemas de prazo. Caso seja registrado mais de um problema, eles devem ser priorizados, para que os problemas considerados mais críticos sejam resolvidos primeiro.

Pré-atividade: Não há.

Entradas: Problemas identificados nos projetos.

Saídas: Problemas registrados e priorizados.

Atividade 2: Identificar Causas do Problema

Nesta atividade, as possíveis causas de um problema registrado são identificadas. Algumas formas de identificar causas de problemas são: buscar situações anteriores similares e o que as causou ou realizar *brainstorming*. As possíveis causas do problema devem ser registradas. Isso pode ser feito textualmente, criando-se uma lista de causas possíveis para o referido problema, ou pode-se utilizar um diagrama de peixe, também chamado de diagrama de *Ishikawa* ou diagrama causa e efeito. Através desse diagrama, é possível representar a relação entre o “efeito” (o problema) e todas as possibilidades de “causa” que podem contribuir para esse efeito. Após serem registradas as possíveis causas, deve-se determinar qual (ou quais) serão consideradas para solucionar o problema.

Pré-atividades: Registrar Problemas.

Entradas: Problemas registrados.

Saídas: Possíveis causas do problema registradas e causas que serão consideradas para solucionar o problema identificadas.

Atividade 3: Estabelecer Critérios de Seleção de Solução

Nesta atividade, os critérios que devem ser levados em consideração para selecionar uma solução para o problema são estabelecidos. Exemplos de critérios são: custo da solução, tempo para implementação da solução e grau de riscos associados à implementação da solução.

Pré-atividades: Identificar Causas do Problema.

Entradas: Informações úteis para estabelecimento dos critérios (restrições do projeto, objetivos do projeto, objetivos da organização etc)

Saídas: Critérios de seleção de solução definidos.

Atividade 4: Identificar Soluções Possíveis

Nesta atividade as possíveis soluções para o problema são identificadas. As possíveis soluções podem ser identificadas analisando-se situações similares anteriores, realizando-se um *brainstorming* ou utilizando-se outras técnicas específicas.

Pré-atividades: Estabelecer Critérios de Seleção de Solução

Entradas: Problema registrado e causas que serão consideradas para solucionar o problema.

Saídas: Soluções possíveis identificadas.

Atividade 5: Avaliar as Soluções Possíveis

Nesta atividade, as soluções identificadas são avaliadas considerando os critérios estabelecidos.

Pré-atividades: Identificar Soluções Possíveis.

Entradas: Soluções possíveis e critérios de seleção de solução.

Saídas: Resultados de avaliação das soluções possíveis.

Atividade 6: Selecionar Solução

Nesta atividade, é feita a escolha da melhor solução, considerando os resultados das avaliações das soluções possíveis, realizadas na atividade anterior. Vale destacar que é possível que mais de uma solução seja selecionada.

Pré-atividades: Avaliar as Soluções Possíveis

Entradas: Resultados de avaliação das soluções possíveis.

Saídas: Solução selecionada.

Atividade 7: Implementar Solução

Nesta atividade, a solução selecionada é implementada.

Pré-atividades: Selecionar Solução.

Entradas: Solução selecionada.

Saídas: Solução implementada.

Atividade 8: Registrar Resultados da Implementação da Solução

Nesta atividade, os resultados da implementação da solução são registrados. Deve-se registrar se a solução, de fato, resolveu o problema (ou parte dele) e quais foram seus

resultados em relação a aspectos considerados relevantes como, por exemplo, tempo, custos e qualidade.

Pré-atividades: Implementar Solução.

Entradas: Dados da implementação da solução.

Saídas: Resultados da implementação da solução registrados.

3.3 Resolve: Propósito do Sistema e Descrição do Minimundo

A ferramenta *Resolve* tem como propósito auxiliar no registro de problemas identificados durante o curso de projetos de software, na identificação de suas causas, na seleção das soluções mais adequadas e no registro dos resultados obtidos com a implementação das soluções selecionadas.

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

Durante o desenvolvimento de projetos de software, problemas podem ser encontrados. Esses problemas devem ser solucionados e os resultados obtidos com as ações tomadas devem ser armazenados para contribuir na solução de problemas futuros.

O primeiro passo da resolução de problemas é a identificação destes. Uma vez identificado um problema, o mesmo deve ser registrado. O registro de um problema inclui seu nome, descrição e categoria. Exemplos de categorias podem ser: problemas tecnológicos, problemas nos requisitos e problemas de prazo. O registro da ocorrência de um problema em um projeto específico deve incluir informações para caracterizar essa ocorrência como: características do projeto em que o problema ocorreu, momento do projeto em que o problema foi detectado e a entidade em que o problema foi detectado (por exemplo, em que processo ou em que artefato). Além disso, também deve ser registrada uma breve descrição da ocorrência do problema, incluindo informações relevantes que podem facilitar seu entendimento. Ocorrências de problemas são registradas no estado '*aguardando análise*'.

Uma ocorrência de problema pode ser selecionada para ser resolvida tão logo seja registrada. No entanto, é possível que várias ocorrências de problemas sejam registradas. Nesse caso, é necessário priorizar as ocorrências de problemas registradas, para que aquelas consideradas mais críticas sejam resolvidas primeiro. A priorização pode se basear na avaliação de alguns aspectos, tais como os danos/prejuízos causados pelo problema até o momento e o impacto sobre o projeto, caso o problema não seja resolvido de imediato. Os resultados da avaliação das ocorrências de problemas segundo aspectos como os citados,

permite que as ocorrências mais críticas sejam identificadas e fiquem no topo da lista de ocorrências de problemas a serem resolvidas.

Uma ocorrência de problema selecionada para ser resolvida, passa para o estado ‘*em resolução*’, e suas causas devem ser identificadas. A identificação de causas pode ser feita por meio de uma consulta a ocorrências anteriores do problema, que pode fornecer as causas do problema em cada ocorrência. A identificação de causas também pode ser feita por meio de *brainstorming*, entrevistas, investigação de documentos e análise de dados históricos do projeto. O conhecimento tácito de gerentes de projetos e outros membros da equipe que tenham passado por situação similar em projetos anteriores também é útil. As possíveis causas do problema devem, então, ser registradas. O registro pode ser feito textualmente, criando-se uma lista de causas possíveis para o referido problema, ou pode-se utilizar um diagrama espinha de peixe, também chamado de diagrama de *Ishikawa* ou diagrama causa e efeito. Através desse diagrama é possível representar a relação entre o “efeito” (o problema) e todas as possibilidades de “causa” que podem contribuir para esse efeito. Após serem registradas as possíveis causas do problema, deve-se determinar qual (ou quais) serão tratadas para solucionar a ocorrência do problema

Para determinar a solução a ser adotada para resolver a ocorrência do problema, é necessário que sejam estabelecidos critérios que devem ser levados em consideração para selecionar a solução mais adequada. Exemplos de critérios são: custo da solução, tempo para implementação da solução e grau de riscos associados à implementação da solução.

Assim como na identificação das possíveis causas, as possíveis soluções podem ser identificadas com base em dados de ocorrências anteriores do problema, em *brainstormings* e na experiência das pessoas envolvidas no projeto. As soluções identificadas devem ser avaliadas considerando os critérios de seleção estabelecidos. Como resultado, tem-se a seleção de uma solução (ou de um conjunto de soluções) que deve ser implementada a fim de resolver a ocorrência do problema.

Após a implementação da solução, os resultados obtidos devem ser registrados para auxiliar na solução de novas ocorrências do problema. Para isso, deve-se registrar se a solução, de fato, resolveu o problema (ou parte dele) e quais foram seus resultados em relação a aspectos considerados relevantes como, por exemplo, tempo, custos e qualidade. Quando a solução adotada foi efetiva, a ocorrência do problema passa para o estado ‘*resolvida*’. Caso apenas parte da ocorrência do problema tenha sido solucionada, a ocorrência do problema passa para o estado ‘*parcialmente resolvida*’ e caso a solução não tenha sido efetiva, a ocorrência recebe o estado ‘*não resolvida*’. Ocorrências de problemas

parcialmente resolvidas ou *não resolvidas* podem ser resubmetidas ao processo de resolução de problemas, a fim de que novas soluções possam ser implementadas. Caso isso ocorra, é necessário identificar as ocorrências de problemas resubmetidas.

Com base na descrição do minimundo foram identificados os requisitos funcionais, não funcionais e regras de negócio de *Resolve*. Tabelas contendo essas informações encontram-se no Documento de Requisitos de *Resolve*, disponível no site do Projeto ODE (<https://sites.google.com/site/projetoode>).

3.4 Subsistemas da Ferramenta Resolve

A Figura 3.2 mostra os subsistemas identificados no contexto do presente projeto, os quais são descritos em seguida.

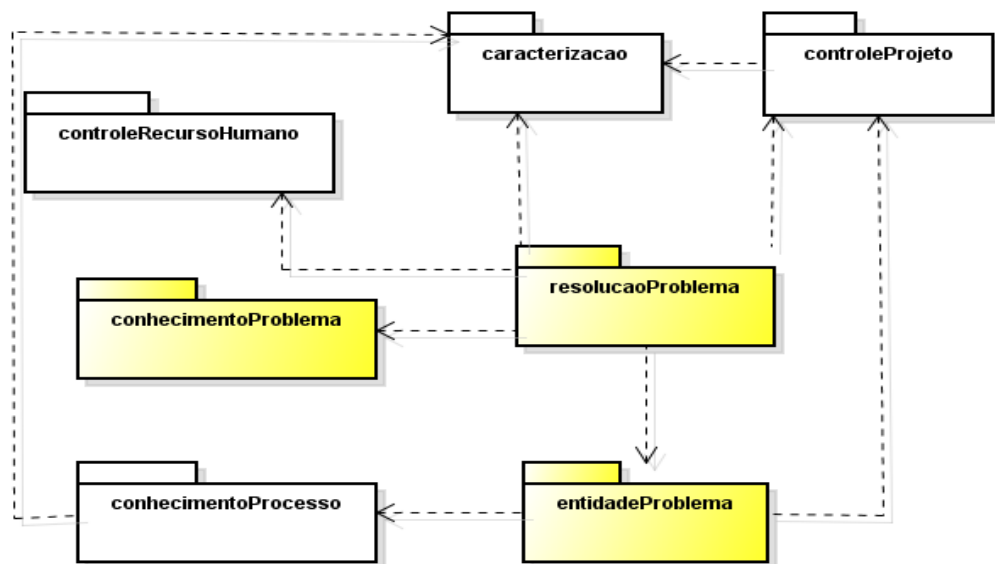


Figura 3.2 – Diagrama de Pacotes e os Subsistemas Identificados.

Os pacotes na cor branca representam pacotes oriundos de ODE e que são utilizados pela ferramenta *Resolve*. São eles:

- *conhecimentoProcesso*: contém as classes que representam conhecimento acerca de processos de software.
- *controleRecursoHumano*: contém as classes e funcionalidades responsáveis pelo registro e controle dos recursos humanos de uma organização.
- *caracterizacao*: envolve a infraestrutura genérica para caracterização de itens de software e uso dessa caracterização para recuperar itens similares.

- *controleProjeto*: contém as classes e funcionalidades necessárias ao registro de projetos.

Os pacotes da cor amarela foram desenvolvidos durante este trabalho exclusivamente para a ferramenta *Resolve*. São eles:

- *conhecimentoProblema*: contém as classes que representam conhecimento acerca de resolução de problemas.
- *resoluçãoProblema*: contém as classes e funcionalidades necessárias para a resolução de problemas em projetos, incluindo seu registro, análise de causas, seleção de solução e registro dos resultados de implementação de solução.
- *entidadeProblema*: envolve as entidades onde problemas podem ser encontrados. No contexto deste projeto, essas entidades estão limitadas a artefatos e projetos.

3.5 Casos de Uso da Ferramenta *Resolve*

Em um levantamento de requisitos, é interessante capturar uma porção discreta da funcionalidade e representá-la pela descrição e diagramação de casos de uso. Nesse contexto, os atores são aqueles que interagem com o sistema e executam as funcionalidades descritas pelo diagrama.

Em *Resolve* três são os atores que interagem com a ferramenta: O **membro do projeto** é um membro da organização que faz parte de equipes de projetos e pode registrar a ocorrência de problemas nos projetos dos quais participa. O **gerente de projeto** é um membro da organização responsável pela gerência de projetos, realiza a análise dos problemas que são registrados e conduzir sua resolução. O **engenheiro de software** é responsável pelo registro do conhecimento necessário para a resolução de problemas, o que inclui o registro de problemas, causas e soluções em geral.

A seguir serão apresentados os casos de uso da ferramenta *Resolve*, 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 *Resolve*, disponível site do Projeto ODE (<https://sites.google.com/site/projetoode>).

3.5.1 Subsistema *conhecimentoProblema*

Neste subsistema estão as funcionalidades de cadastro da ferramenta que, no contexto de ODE, representam o conhecimento da organização acerca de resolução de problemas. Os casos de uso deste subsistema são apresentados na Figura 3.3.

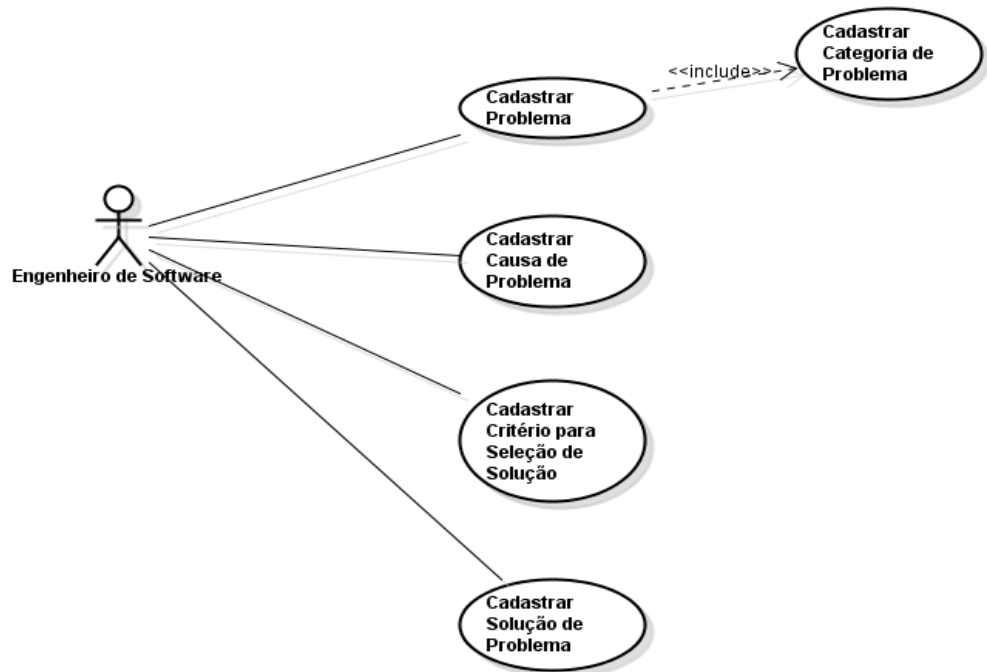


Figura 3.3 – Diagrama de Casos de Uso do Subsistema conhecimentoProblema.

Todos os casos de uso do subsistema conhecimentoProblema são cadastrais e envolvem inclusão, exclusão, consulta e alteração. O cadastro de categoria (*Cadastrar Categoria de Problema*) é necessário para que seja cadastrado um problema (*Cadastrar Problema*). Ao cadastrar uma causa (*Cadastrar Causa de Problema*) é necessário indicar os problemas que podem ser provocados pela causa. Ao cadastrar uma solução (*Cadastrar Solução de Problema*) é necessário indicar as causas que podem ser tratadas pela solução. Há ainda o cadastro de critério para seleção de solução (*Cadastrar Critério para Seleção de Solução*). Todos os cadastros devem ser feitos pelo engenheiro de software.

3.5.2 Subsistema *entidadeProblema*

A Figura 3.4 apresenta o diagrama de casos de uso do subsistema *entidadeProblema*.



Figura 3.4 – Diagrama de Casos de Uso do Subsistema entidadeProblema.

O caso de uso (*Registrar Entidade Problema*) permite ao engenheiro de software criar, consultar, editar e excluir entidades onde possam ocorrer problemas.

3.5.3 Subsistema *resolucaoProblema*

Neste subsistema encontra-se todo o processo de resolução de problemas propriamente dito. Nele estão as funcionalidades que permitem registrar uma ocorrência de problema, resolver a ocorrência de um problema e registrar os resultados da implementação da solução adotada.

A Figura 3.5 apresenta o diagrama de casos de uso do subsistema resoluçãoProblema.

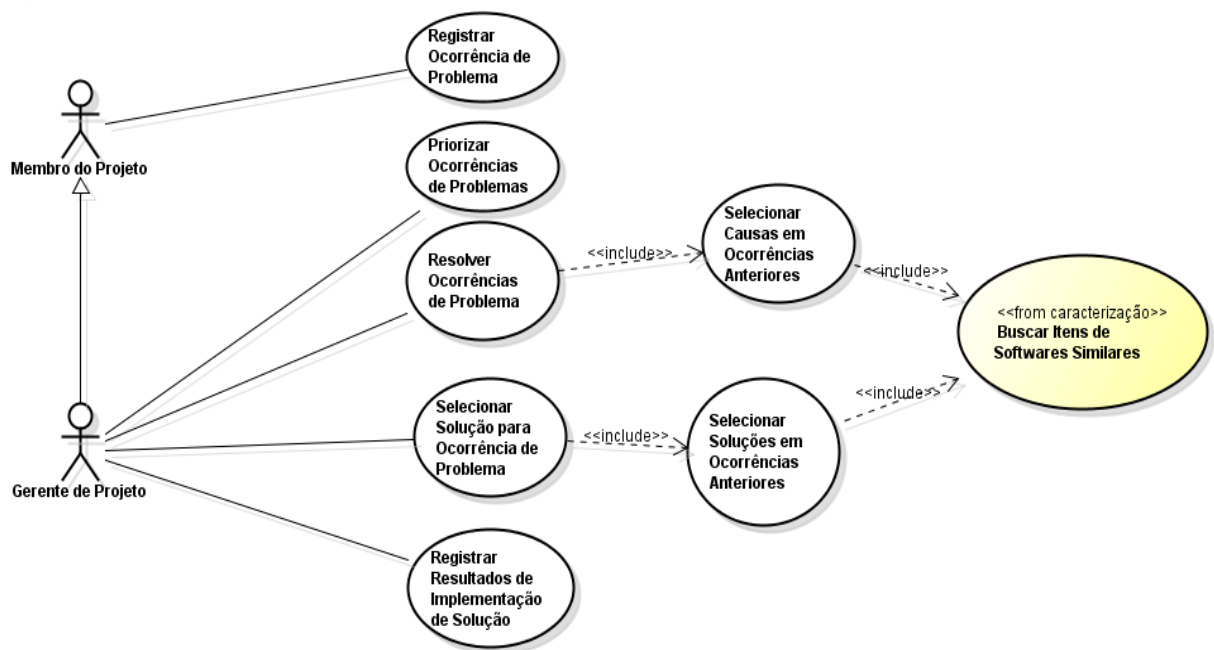


Figura 3.5 – Diagrama de Casos de Uso do Subsistema resolucaoProblema

A única funcionalidade que o membro do projeto pode realizar é registrar uma ocorrência de problema (*Registrar Ocorrência de Problema*). Ele seleciona, dentre os problemas cadastrados, quais são pertinentes para serem resolvidos. Se mais de uma ocorrência de problema for registrada, é possível que o gerente de projeto priorize quais serão solucionadas primeiro (*Priorizar Ocorrências de Problemas*).

Para resolver uma ocorrência de problema (*Resolver Ocorrência de Problema*), o gerente seleciona dentre as possíveis causas de um problema, quais são pertinentes para a ocorrência do problema sendo analisada. Além disso, é possível selecionar uma ou mais soluções para a ocorrência de problema. O gerente de projeto também pode selecionar causas (*Selecionar Causas em Ocorrências Anteriores*) e soluções (*Selecionar Soluções em Ocorrências Anteriores*) a partir de ocorrências anteriores do problema. Além disso, o gerente de projeto tem a opção de selecionar uma ou mais soluções para a ocorrência de problema de acordo com critérios (*Selecionar Solução para Ocorrência de Problema*).

Após a implantação da solução, o gerente de projetos registra resultados de implementação da solução (*Registrar Resultado de Implementação de Solução*) que foi escolhida, informando se a solução resolveu (total ou parcialmente) ou não a ocorrência do problema.

3.6 Diagramas de Classes da Ferramenta *Resolve*

O diagrama de classes visa capturar e descrever as informações que o sistema deve representar para prover as funcionalidades descritas nos casos de uso. A seguir, serão apresentados os diagramas de classes de cada subsistema. Outras informações, tais como restrições de integridade, glossário do projeto e modelos conceituais comportamentais encontram-se no Documento de Especificação de Requisitos na página do projeto ODE <https://sites.google.com/site/projetoode>.

3.6.1 Subsistema *conhecimentoProblema*

A Figura 3.6 apresenta o diagrama de classes do subsistema *conhecimentoProblema*. Nenhuma classe de outro pacote é utilizado neste diagrama.

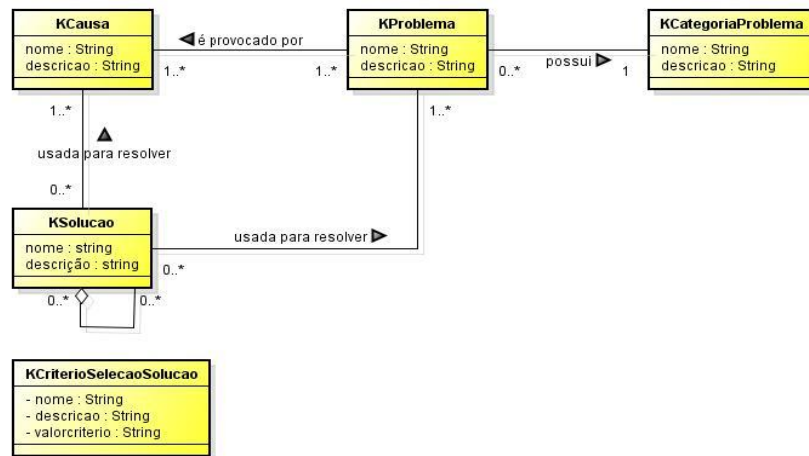


Figura 3.6– Diagrama de Classes do Subsistema *entidadeProblema*.

Problemas (KProblema) possuem uma categoria de problema (KCategoriaProblema) e são provocados por uma ou mais causas (KCausa). Uma solução (KSolucao) pode ser formada por várias soluções e é utilizada para resolver um ou mais problemas. De problema, causa, solução e categoria de problema deseja-se saber nome e descrição, apenas. Critérios de seleção solução (KCritérioSelecaoSolucao) são critérios que podem ser utilizados para selecionar as soluções mais apropriadas em um determinado contexto. De um critério seleção de solução deseja-se saber nome, descrição e o valor de critério de seleção de solução.

Subsistema *entidadeProblema*

A Figura 3.7 apresenta o diagrama de classes do subsistema *entidadeProblema*

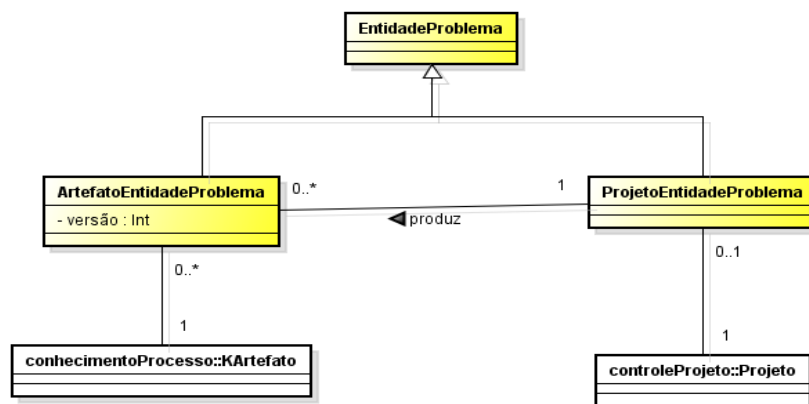


Figura 3.7– Diagrama de Classes do Subsistema *entidadeProblema*.

EntidadeProblema: classe que especializa as entidades nas quais problemas podem ocorrer. No contexto deste trabalho, uma EntidadeProblema pode ser um projeto (ProjetoEntidadeProblema) ou um artefato (ArtefatoEntidadeProblema).

ProjetoEntidadeProblema: classe que armazena os projetos registrados em ODE como entidade problema.

ArtefatoEntidadeProblema: classe que armazena os artefatos registrados em ODE como entidade problema

Subsistema *resolucaoProblema*

A Figura 3.8 apresenta o diagrama de classes do subsistema *resolucaoProblema*.

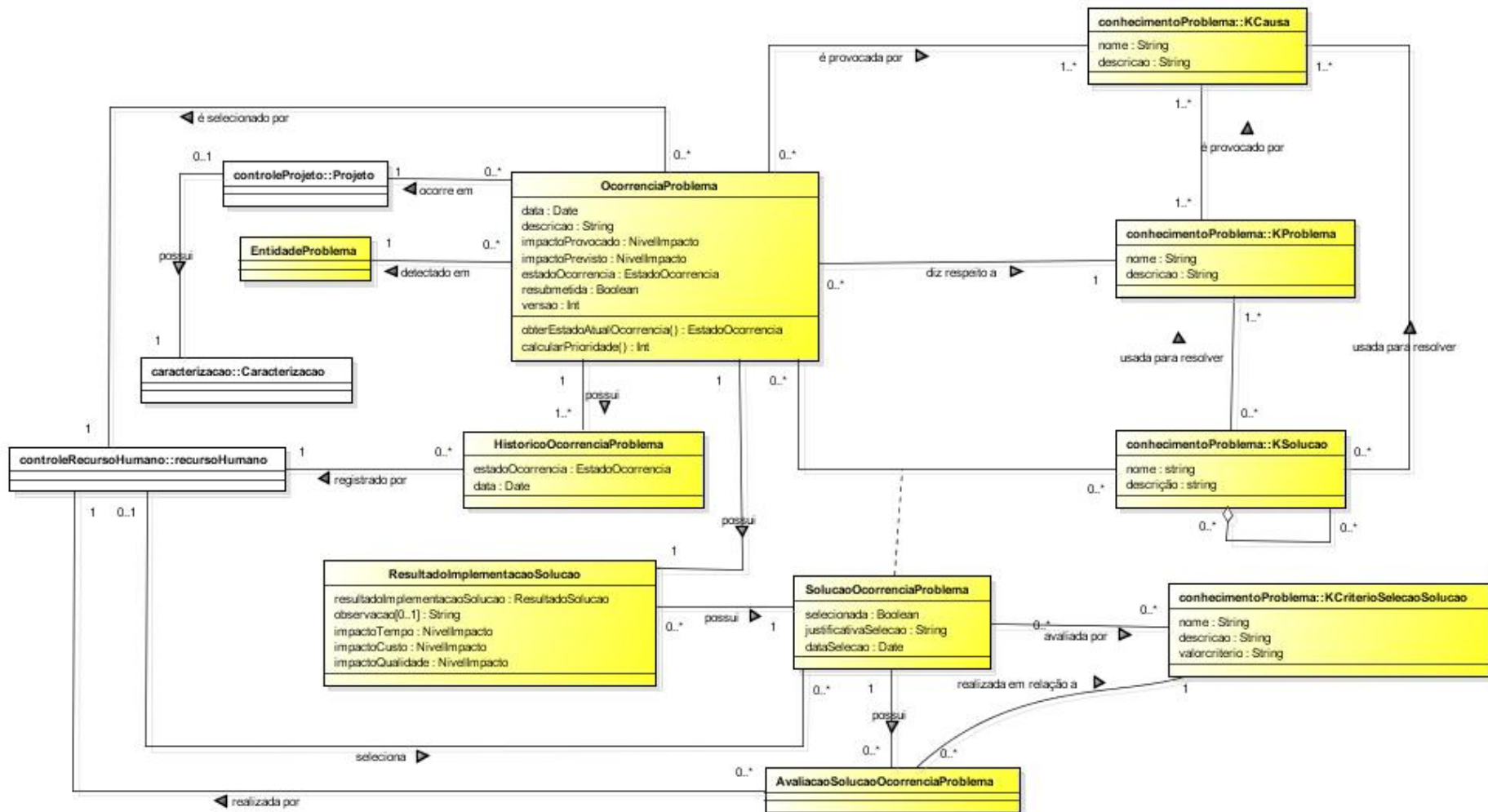


Figura 3.8 – Diagrama de Classes do Subsistema *resolucaoProblema*.

- OcorrenciaProblema: classe que armazena as ocorrências de problemas em projetos.
- HistoricoOcorrenciaProblema: classe que armazena o histórico das alterações nos estados da ocorrência de problema.
- SolucaoOcorrenciaProblema: classe que armazena as soluções propostas para resolver uma ocorrência de problema.
- AvaliacaoSolucaoOcorrenciaProblema: classe que armazena a avaliação realizada para as soluções propostas para a ocorrência de um problema, a fim de selecionar quais serão implementadas.
- ResultadoImplementacaoSolucao: classe que armazena o resultado obtido com a implementação de uma solução.

Capítulo 4

Projeto e Implementação da Ferramenta *Resolve*

4.1 Introdução

Uma vez identificados e modelados os requisitos, a tecnologia deve ser incorporada à solução, o que ocorre na fase de projeto. Sendo assim, é necessário conhecer a tecnologia disponível e os ambientes de hardware e software onde o sistema será desenvolvido e implantado. Durante o projeto, deve-se decidir como o problema será resolvido, começando em um alto nível de abstração, próximo da análise, e progredindo sucessivamente para níveis mais detalhados até se chegar a um nível de abstração próximo da implementação (FALBO, 2011). Uma vez definido o projeto, o sistema pode ser implementado e testado.

Neste capítulo são apresentados os principais aspectos do Projeto de Sistema da ferramenta *Resolve*. Algumas de suas telas também são apresentadas. Na seção 4.2 a arquitetura de software de *Resolve* 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 todos os componentes de apenas um subsistema da ferramenta. O detalhamento de projeto dos demais subsistemas encontra-se no Documento de Projeto de Sistema de *Resolve*, disponível site do Projeto ODE (<https://sites.google.com/site/projetoode>). No Documento de Projeto de Sistema de *Resolve*, além do detalhamento dos componentes da arquitetura de todos os subsistemas, também há informações sobre as tecnologias utilizadas e sobre as táticas de projeto realizadas para atender os requisitos não funcionais identificados no Documento de Requisitos de *Resolve* (também disponível no site do Projeto ODE).

4.2 Arquitetura de Software

A arquitetura de software da ferramenta *Resolve* baseia-se na combinação de camadas e partições. Inicialmente, para cada subsistema identificado na fase de análise foi definida uma partição. Cada uma dessas partições, por sua vez, está organizada em três camadas, a saber: camadas de Interface com o Usuário (ciu), que contém 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 ODE e que são próprios para o reuso.

Para organizar a camada de interface com o usuário, foi adotado o padrão Modelo-Visão-Controlador (MVC). O padrão Modelo-Visão-Controlador (MVC) considera três papéis relacionados à interação humano-computador. O Modelo refere-se aos objetos que representam alguma informação sobre o negócio e corresponde, de fato, a objetos da camada de Lógica de Negócio. A Visão refere-se à entrada e à exibição de informações na IU. Qualquer requisição é tratada pelo terceiro papel: o controlador. Este pega a entrada do usuário, envia uma requisição para a camada de lógica de negócio, recebe sua resposta e solicita que a visão se atualize conforme apropriado (FALBO, 2011).

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. Para cada classe de domínio a ser persistida, devem ser criadas uma classe `DAOBaseImpl*` e uma interface `DAOBase*` 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, .

A Figura 4.1 mostra o projeto da arquitetura de software da ferramenta *Resolve*, e nas próximas seções serão apresentados fragmentos dos diagramas de projeto.

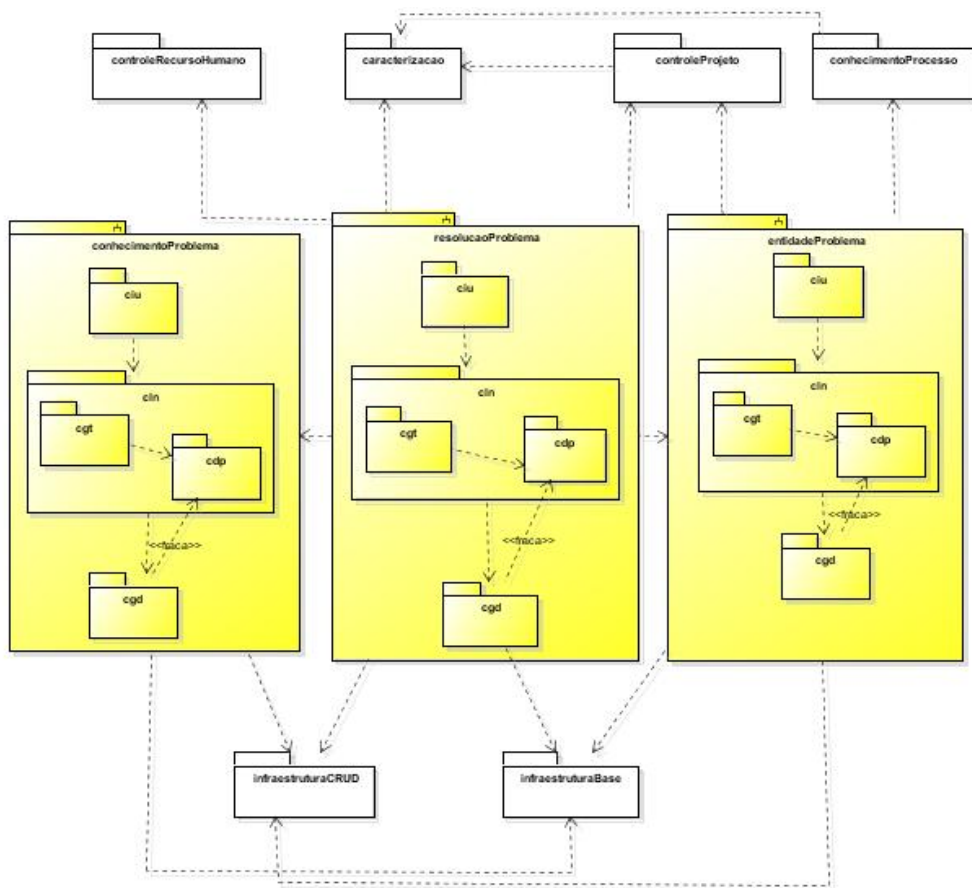


Figura 4.1 – Arquitetura de Software Completa

A seguir é apresentado o projeto detalhado dos componentes da arquitetura dos subsistemas de Resolve. Como dito na introdução deste capítulo, apenas um subsistema tem todos os componentes descritos no texto desta monografia. Assim, na próxima seção são descritos os componentes da arquitetura do *subsistemaResolucaoProblema*. Para os demais subsistemas, são apresentados os modelos de classes de seus componentes de domínio do problema.

4.3.1 Subsistema resolucaoProblema

4.3.1.1 Camada Lógica do Negócio

Para organizar a camada de lógica de negócio deste subsistema e dos demais subsistemas de Resolva, 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). 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 seguir, o projeto desses dois componentes é apresentado.

4.3.1.1.1 Componente Domínio do Problema

A Figura 4.2 apresenta o diagrama de classes do cdp do subsistema resolucaoProblema. Pode-se observar que os tipos específicos de domínio identificados na etapa de análise passaram a ser representados como classes no modelo de classes de projeto, as classes associativas tornaram-se classes regulares e as navegabilidades entre as classes foram identificadas.

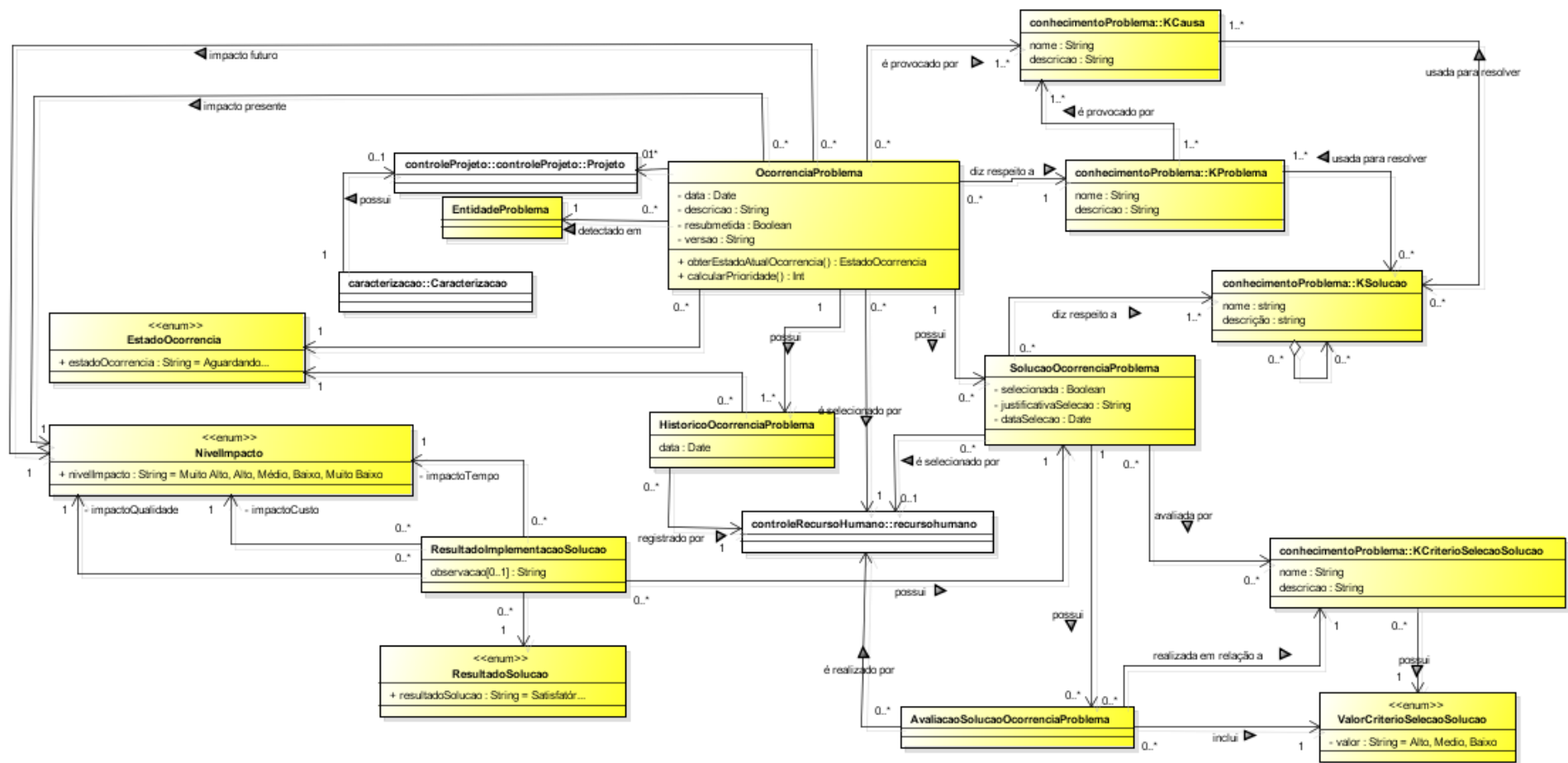


Figura 4.2 - Diagrama de Classes do cdp do subsistema *resolucaoProblema*

4.3.1.1. 2 – Componente de Gerência de Tarefas (CGT)

No projeto do CGT, optou-se por estabelecer uma classe de aplicação para cada caso de uso . 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
AplRegistrarOcorrenciaProblema	Registrar Ocorrência de Problema
AplPriorizarOcorrenciaProblema	Priorizar Ocorrências de Problemas
AplResolverOcorrenciaProblema	Resolver Ocorrências de Problema
AplSelecionarSolucao	Selecionar Solução para Ocorrência de Problema
AplRegistrarResultadoImplementacao	Registrar Resultados de Implementação ds Solução

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 4.3.

4.3.1.2– Camada de Interface com o Usuário

A Figura 4.3 apresenta o diagrama de classes (parcial) do CIU do subsistema resolucaoProblema, referente ao registro do resultado da implementação da solução. A Figura 4.4 apresenta o diagrama de classes (parcial) do CIU do subsistema resolucaoProblema, referente ao caso de uso Registrar Ocorrência de Problema.

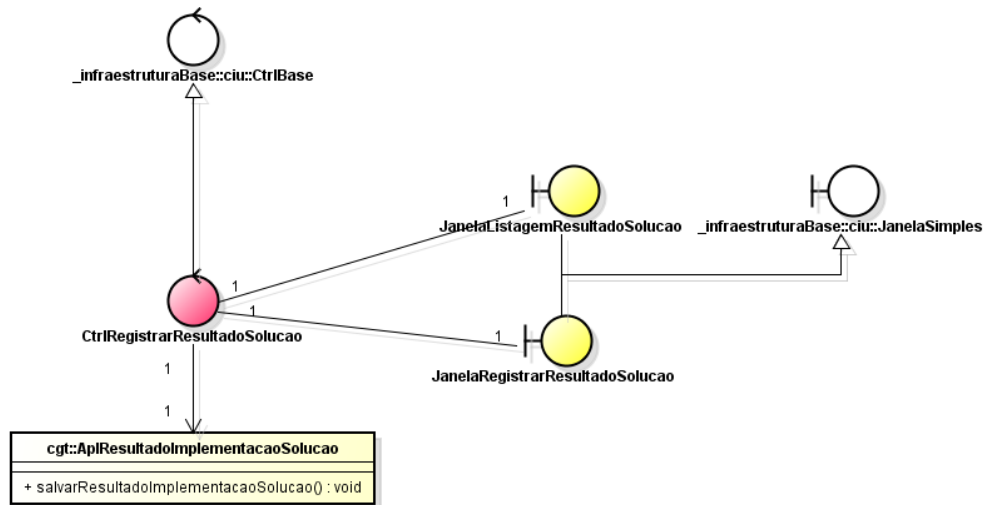


Figura 4.3- Diagrama de Classes (parcial) do CIU do Subsistema resolucaoProblema, referente ao caso de uso Registrar Resultado Implementação Solução.

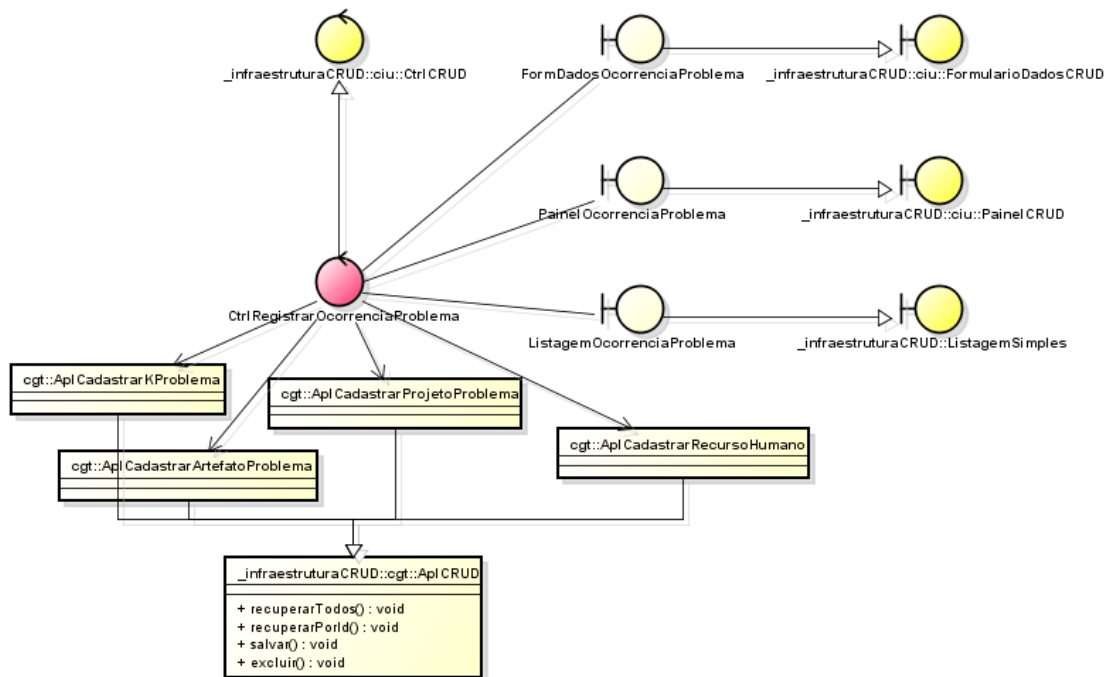


Figura 4.4 - Diagrama de Classes (parcial) do CIU do Subsistema resolucaoProblema, referente ao caso de uso Registrar Ocorrência de Problema.

4.3.1.2 – Camada de Gerência de Dados

A Figura 4.5 mostra um fragmento do modelo da camada de Gerência de Dados do subsistema *resolucaoProblema* referente ao registro do resultado da implementação da solução e a Figura 4.6 referente ao registro de ocorrência de problema. Na figura, apenas algumas classes são apresentadas, uma vez que as demais classes do subsistema são modeladas da mesma forma.

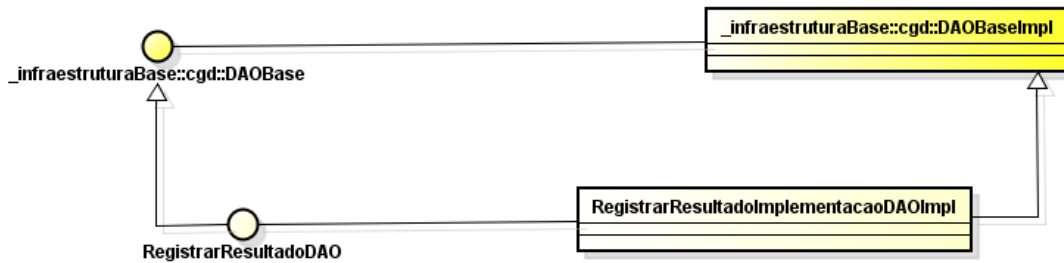


Figura 4.5- CGD do Subsistema ResolucaoProblema, referente ao caso de uso Registrar Resultado de Implementação

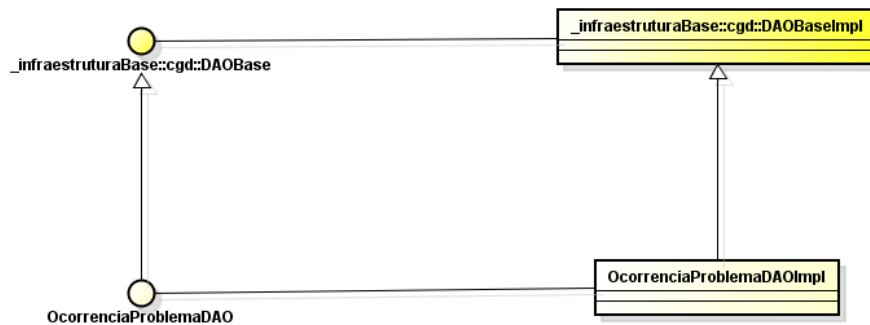


Figura 4.6- CGD do Subsistema ResolucaoProblema, referente ao caso de uso Registrar Ocorrência de Problema

4.4 A Ferramenta Resolve

Nesta seção são apresentadas algumas telas da ferramenta *Resolve*. De forma geral, *Resolve* é composta por um conjunto de funcionalidades cadastrais, que permitem o armazenamento de dados iniciais para a utilização da ferramenta, como, por exemplo, cadastro de problemas, causas e soluções, e por um conjunto de funcionalidades que apoiam o processo de resolução de problemas.

A parte cadastral da ferramenta está presente no subsistema *conhecimentoProblema*. Ela permite o armazenamento de dados tais como categorias de problemas, problemas e suas possíveis causas e soluções. Por se tratarem de cadastros simples, essa parte de *Resolve* não será detalhada nesta seção.

Com o intuito de mostrar o funcionamento da ferramenta, serão apresentadas a 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 4.7 é apresentada a tela principal do ambiente ODE, de onde a ferramenta *Resolve* é acessada.

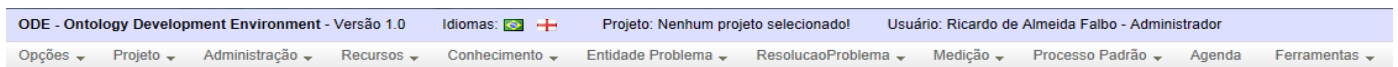
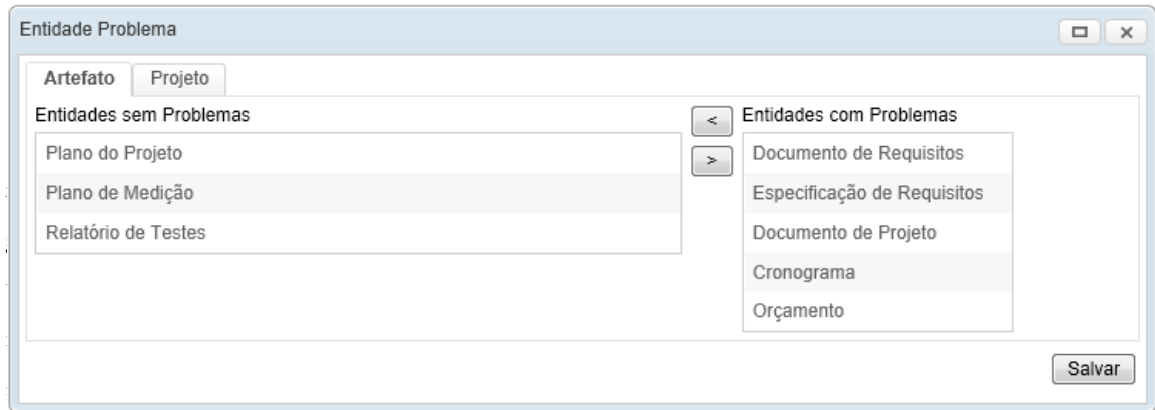


Figura 4.7– Tela principal de ODE, para acesso à *Resolve*.

No subsistema *EntidadeProblema* é realizado o registro de projetos e artefatos para os quais poderão ser informados problemas. Esse subsistema se fez necessário para ser possível utilizar a ferramenta *Resolve* independente de ODE. Quando *Resolve* é utilizada integrada ao ODE, projetos e artefatos são cadastrados em ferramentas específicas. Uma vez que essas ferramentas não estão disponíveis no uso de *Resolve* desacoplada de ODE, foi necessário prover uma funcionalidade para que o usuário possa registrar projetos e artefatos, que, na primeira versão de *Resolve*, são as entidades para as quais podem ser registrados problemas. O cadastro de projetos é um cadastro básico e, por isso, não é mostrado aqui.

Em ODE, tipos de artefatos (ex.: Especificação de Requisitos, Documento de Projeto, Plano do Projeto etc.) são registrados no cadastro de artefatos presente no pacote *conhecimentoProcesso*. Em *Resolve*, o engenheiro de software, que é quem faz os cadastros iniciais necessários à execução do processo de resolução de problemas, seleciona dentre os tipos de artefatos registrados, quais serão considerados pelo processo de resolução de problemas. Isso é feito utilizando-se a tela apresentada na Figura 4.8.



4.8 – Tela para registro de artefatos que serão considerados no processo de resolução de problemas.

O processo de resolução de problemas propriamente dito está presente no subsistema *ResolucaoProblema*, no qual o usuário pode registrar ocorrências de problemas, selecionar soluções para resolver as ocorrências de problemas e registrar o resultado da implementação das soluções.

Para registrar uma ocorrência de problema, o usuário utiliza a tela apresentada na Figura 4.9. O usuário deve informar um nome, o problema relacionado ao qual a ocorrência se refere, a descrição da ocorrência, a data, a entidade onde o problema ocorreu (artefato ou projeto), o responsável pelo registro da ocorrência, os impactos presente (causado até o momento) e futuro (será causado se não for resolvido). Além disso, é inserido uma breve descrição sobre a ocorrência do problema.

Cadastro de Ocorrência de Problema

Nome:	Problema em Requisitos
Data:	11/03/2012
Problema*:	Ambiguidade em Requisitos
Artefato*:	Documento de Requisitos
Versão:	1.2
Projeto*:	Resolve
Recurso Humano*:	Ricardo de Almeida Falbo
Impacto Presente*:	Medio
Impacto Futuro*:	Muito Alto
Descrição:	A descrição dos requisitos R28 e R29 não está muito clara.

Salvar Cancelar

Figura 4.9 – Tela para registro de Ocorrência de Problemas

Ocorrências de problemas podem ser selecionadas para serem resolvidas utilizando-se a tela apresentada na Figura 4.10. Para resolver uma ocorrência de problema o usuário deve selecioná-la dentre as ocorrências de problema registradas. A princípio, as ocorrências são apresentadas em ordem de data de registro. No entanto, o usuário pode solicitar que elas sejam apresentadas em ordem de prioridade, que é estabelecida com base na data de registro e nos níveis de impactos informados. Para isso, o usuário deve clicar no botão *Priorizar Ocorrência de Problemas*.

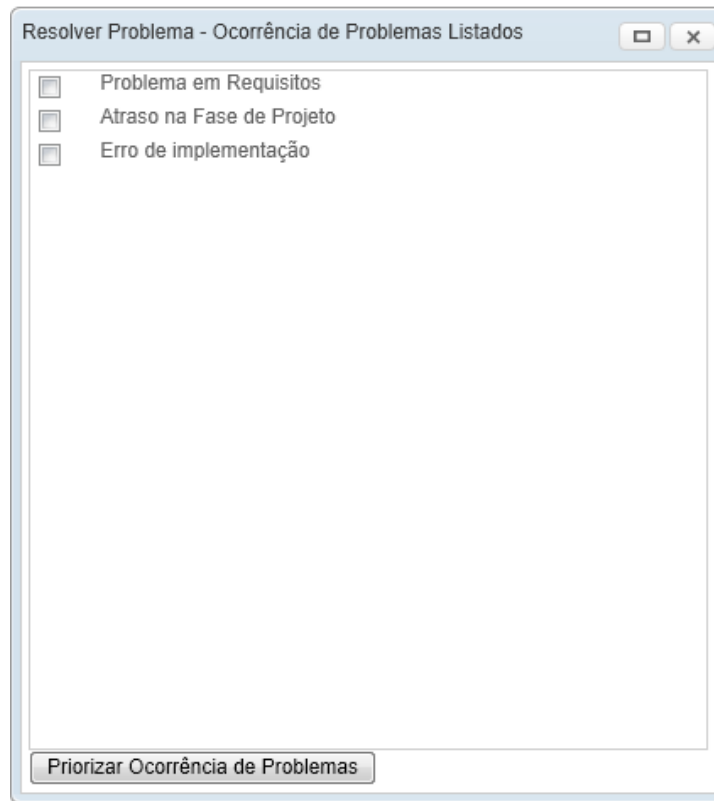


Figura 4.1-0 – Tela de seleção de ocorrência de problema para ser resolvida.

Após escolher uma ocorrência de problema, o usuário deve selecionar as causas do problema e as possíveis soluções. Isso é feito na tela da Figura 4.11. A tela apresenta as causas relacionadas ao problema escolhido no registro da ocorrência de problema. Quando o usuário escolhe uma causa (parte superior da tela), soluções relacionadas a ela são exibidas (parte inferior da tela).

Para selecionar causas e soluções, o usuário pode consultar as causas e soluções para ocorrências do problema em projetos anteriores clicando nos botões *Selecionar Causas em Ocorrências Anteriores* ou *Selecionar Soluções em Ocorrências Anteriores*. Essa opção pode ser útil quando várias causas e soluções são possíveis.

Caso existam várias soluções possíveis, o usuário pode avaliá-las utilizando alguns critérios, a fim de selecionar a mais adequada.

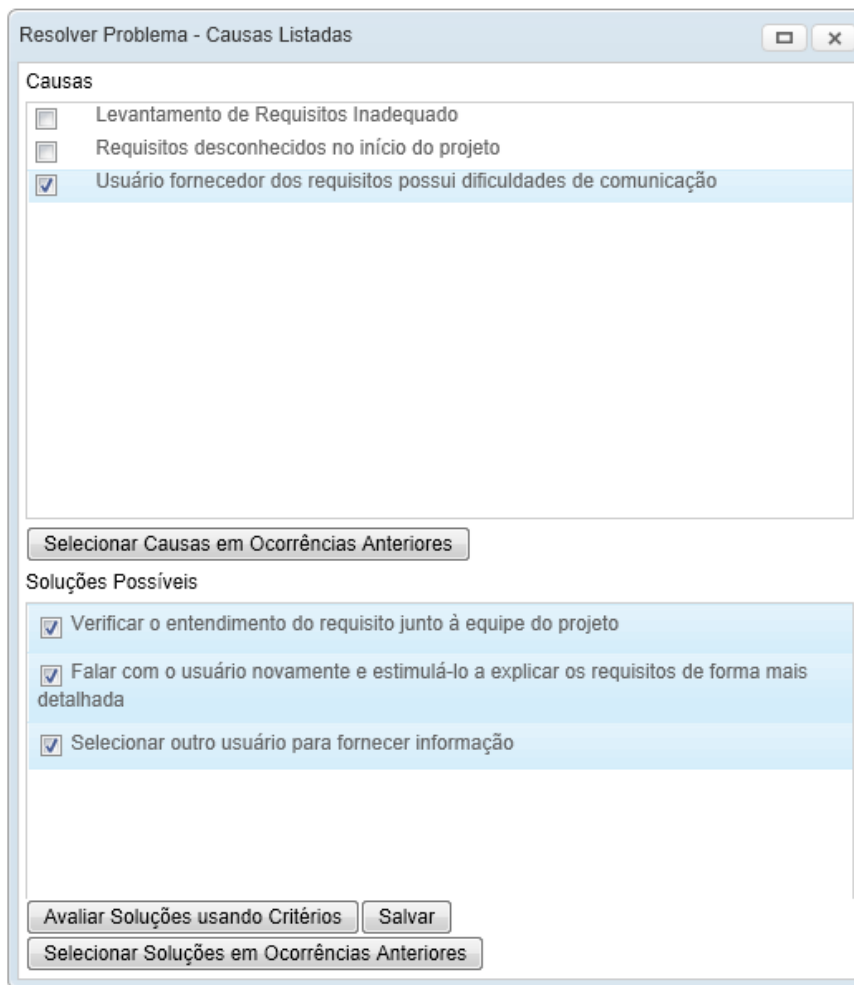


Figura 4.11 – Tela de seleção de causas e soluções.

Para avaliar uma solução utilizando critérios, o usuário utiliza a tela da Figura 4.12, na qual deve indicar o peso de cada critério de avaliação de solução. Convém dizer que os critérios são cadastrados na parte de cadastros básicos da ferramenta. Quanto maior for a importância do critério, maior deve ser o peso, que pode ter os seguintes valores: 1, 2 ou 3.

Para avaliar cada solução segundo os critérios, o usuário clica no botão *Avaliar* em frente à solução a ser avaliada.

Avaliar Soluções com Critérios	
Critérios	Peso
Custo	1
Tempo	2
Riscos	1
Soluções	Resultado da Avaliação
Verificar o entendimento do requisito junto à equipe do projeto	0 Avaliar
Falar com o usuário novamente e estimulá-lo a explicar os requisitos de forma mais detalhada	0 Avaliar
Selecionar outro usuário para fornecer informação	0 Avaliar

Figura 4.12 – Tela de avaliação de soluções(1).

Avaliar Critérios	
Critérios	
Custo	Baixo
Tempo	Baixo
Riscos	Baixo
Solução: Selecionar outro usuário para fornecer a informação	
Descrição:	<input type="text"/>
<input type="button" value="Salvar"/>	

Figura 4.13 – Tela de avaliação de soluções (2).

O usuário deve realizar a avaliação das soluções em relação aos critérios informando os valores Alto, Médio ou Baixo para cada critério e apresentando uma breve descrição sobre sua avaliação. Depois de escolher a opção Salvar, O resultado da avaliação da solução, que antes era zero na tela da Figura 4.12, é atualizado.

Uma vez selecionada a solução que a ser adotada e que esta tenha sido implementada o usuário pode registrar o resultado obtido. Para isso, ele utiliza a tela da Figura 4.14, onde devem ser informados os impactos da implementação da solução em relação ao tempo, custo e qualidade do projeto, o nível de satisfação com solução e possíveis observações.

Registro de Resultado de Solução

Impacto Tempo *:	Muito Baixo
Impacto Custo *:	Muito Alto
Impacto Qualidade *:	Baixo
Resultado da Solução *:	Satisfatorio
Observação *:	A consulta a outro usuário permitiu o esclarecimento do requisito, eliminando, assim, a ambigüidade.

Salvar Cancelar

Figura 4.14 – Tela de registro de resultado de implementação de solução.

Capítulo 5

Considerações Finais

5.1 Conclusões

A resolução de problemas é muito importante no contexto de projetos de software. A existência de uma ferramenta de auxílio ao registro dos problemas, causas e soluções, bem como dos resultados das soluções utilizadas pode auxiliar as organizações a armazenarem seu conhecimento em relação à resolução de problemas.

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>Resolve</i> . As tabelas contendo essas informações encontram-se no Documento de Requisitos de <i>Resolve</i> .
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>Resolve</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>Resolve</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 previsto.

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:

priorização das ocorrências de problemas que leva em consideração o impacto presente e futuro da ocorrência de problema e a seleção de causas e soluções em ocorrências anteriores. O último não foi feito por depender da ferramenta Caracterização que ainda não está concluída. Além disso, algumas questões de interface precisam ser melhoradas. Essas questões estão registradas em uma lista de pendências para implementação 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 desse trabalho passou-se por todas as fases do processo de desenvolvimento de software (especificação de requisitos, análise, projeto do sistema, desenvolvimento 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 da resolução de problemas, tema totalmente novo e não visto durante a graduação.

A principal dificuldade encontrada para a realização deste trabalho foi a visão de todo o ciclo de um software. Durante a graduação, não houve a implementação de um software que tenha sido devidamente documentado em suas fases de especificação e análise de requisitos e projeto de sistema. Durante este trabalho, isso ocorreu e permitiu a percepção de que todas as etapas da Engenharia de Software são importantes para a construção de uma ferramenta. Além disso, o ambiente ODE é complexo e demorou muito tempo para que pudesse ser entendido. A rotatividade dos membros do projeto faz com que a experiência adquirida por meses seja levada, enquanto os novos membros têm que aprender praticamente de forma autodidata toda a infraestrutura.

5.2 Trabalhos Futuros

No final do desenvolvimento de um software, tipicamente novas necessidades são identificadas. A manutenção e evolução de software deve 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.

Alguns trabalhos futuros tratam, na verdade, da implementação de aspectos que não foram finalizados na versão atual. Dentre eles, pode-se citar a integração com a ferramenta de caracterização de projetos de ODE, que se encontra em desenvolvimento. A integração com essa ferramenta permitiria a seleção de projetos similares para analisar causas e soluções de ocorrências de problemas.

Uma possível evolução da ferramenta, é a implementação de apoio que permita a criação de diagramas de causa e efeito para apoiar a análise de causas de problemas.

Além disso, é essencial que a ferramenta tenha melhor usabilidade. De fato, atualmente, a interface é uma das principais limitações do Ambiente ODE. Uma melhoria nesse sentido seria permitir que o usuário visualize todo o processo de resolução de problemas e identifique em que etapa ele está.

Por fim, outro trabalho futuro é a implementação da versão de *Resolve* para ser utilizada integrada ao ODE.

Referências Bibliográficas

- PARIS, W., 2003, Proposta de uma Metodologia para Identificação de Causa Raiz e Solução de Problemas Complexos em Processos Industriais: um Estudo de Caso, Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba - PR.
- PM Solutions, 2011, Strategies for Project Recovery: A PM SOLUTIONS RESEARCH REPORT, disponível em http://olhardigital.uol.com.br/negocios/digital_news/noticias/problemas_afetam_37_dos_projeto_de_ti_aponta_estudo (acessado em 27/03/2012).
- LORIGGIO, A., 1996, Um Modelo e seus Fatores Críticos de Sucesso. Caderno de Pesquisas em Administração, São Paulo Brasil, v.1, N° 2.
- LORIGGIO, A., 2002, De onde vêm os problemas. Métodos para diagnóstico eficaz. Negócio Editora, São Paulo – SP.
- GONÇALVES, F., 2008, Abordagem para Análise e Resolução de Causas de Problemas aplicando Multicritério, Dissertação de Mestrado, Universidade de Fortaleza, Fortaleza - CE.
- MORITA H., SHIMIZU T., LAURINDO F., 2003, Modelos para Estruturar e Avaliar Alternativas de Decisão em Tecnologia da Informação, Relatório Técnico, Depto. de Engenharia de Produção, Universidade de São Paulo.
- ROSSATO, I., 1996. Uma Metodologia para Análise e Solução de Problemas, Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis – SC.
- FALBO, R., 2011, Notas de Aula Engenharia de Requisitos, Universidade Federal do Espírito Santo, Departamento de Informática.
- MACORATTI, J, 2010, Os requisitos e o sucesso do seu projeto, disponível em http://www.macoratti.net/ger_proj1.htm (acessado em 27/03/2012).
- 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.
- SOFTEX, 2011, *MPS.BR: Melhoria de Processo do Software Brasileiro - Guia Geral : 2011*, Disponível em: <http://www.softex.br/mpsbr>.
- PMBOK: PMI – Project Management Institute, 2008, PMBoK – *Project Management Body of Knowledge*, 4ª Edição.