

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Vinícius Soares Fonseca

**Uma Abordagem Baseada em Ontologias
para Integração Semântica de Ferramentas
de Apoio à Medição de Software**

VITÓRIA
2015

Vinícius Soares Fonseca

Uma Abordagem Baseada em Ontologias para Integração Semântica de Ferramentas de Apoio à Medição de Software

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Orientadora: Monalessa Perini Barcellos

Coorientador: Ricardo de Almeida Falbo

VITÓRIA
2015

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

F676a Fonseca, Vinícius Soares, 1987-
Uma abordagem baseada em ontologias para integração semântica de
ferramentas de apoio à medição de software / Vinícius Soares Fonseca. –
2015.
139 f. : il.

Orientador: Monalessa Perini Barcellos.
Coorientador: Ricardo de Almeida Falbo.
Dissertação (Mestrado em Informática) – Universidade Federal do
Espírito Santo, Centro Tecnológico.

1. Medição de software. 2. Ontologia. 3. Integração semântica (sistemas
de computador). 3. Interoperabilidade semântica. I. Barcellos, Monalessa
Perini. II. Falbo, Ricardo de Almeida. III. Universidade Federal do Espírito
Santo. Centro Tecnológico. IV. Título

CDU: 004

Vinícius Soares Fonseca

Uma Abordagem Baseada em Ontologias para Integração Semântica de Ferramentas de Apoio à Medição de Software

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.
Universidade Federal do Espírito Santo (UFES)
Orientador

Prof. Ricardo de Almeida Falbo, D. Sc.
Universidade Federal do Espírito Santo (UFES)
Coorientador

Prof. João Paulo A. Almeida, Ph. D.
Universidade Federal do Espírito Santo (UFES)

Prof. Julio Cesar Nardi, D. Sc.
Instituto Federal do Espírito Santo (IFES)

Vitória, 17 de Dezembro de 2015

Agradeço a Deus por me abençoar durante mais essa jornada, agradeço à minha família e especialmente à minha esposa Tarcila, por estar sempre ao meu lado me incentivando a lutar pelos meus sonhos.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por estar comigo mais uma vez nesta jornada e me permitir chegar até aqui.

Agradeço aos meus pais Manoel e Selma e ao meu irmão Lucas pela convivência de anos com muito amor e por terem me dado todo apoio que se possa esperar de uma família.

Agradeço à minha esposa Tarcila por compartilhar comigo mais de 10 anos de sua vida, celebrando os momentos de alegria e enfrentando os momentos difíceis ao meu lado.

Agradeço aos professores do mestrado por contribuir para a minha aquisição de conhecimento.

Agradeço aos meus professores orientadores Dra. Monalessa Perini Barcellos e Dr. Ricardo de Almeida Falbo. O apoio de vocês foi essencial para ajudar-me a desenvolver minhas ideias e melhorar a qualidade dessa dissertação. Sem vocês este trabalho certamente não aconteceria.

Agradeço aos membros da banca de defesa do mestrado, os professores Dr. João Paulo Andrade Almeida e Dr. Julio Cesar Nardi, por cederem seu tempo participando da defesa e contribuindo para a melhoria deste trabalho.

Agradeço ao meu amigo Henrique Castro pela enorme colaboração durante a implementação da integração proposta pela abordagem defendida nessa dissertação.

Agradeço ao Paulo Sérgio e à equipe do LEDS por oferecerem um ambiente onde a abordagem pudesse ser usada e avaliada.

Agradeço aos funcionários do PPGI pelo atendimento prestado durante o mestrado.

Agradeço aos colegas do NEMO que cooperaram e me ajudaram durante a elaboração dessa dissertação.

Por fim, agradeço a todos que contribuíram de alguma forma com este trabalho.

RESUMO

Medição de software é considerada uma prática fundamental para a melhoria de processos e a gerência de projetos, uma vez que fornece dados para apoiar a tomada de decisão nos níveis organizacional e de projetos. Para que a medição de software seja eficaz, ela deve ser orientada aos objetivos da organização ou dos projetos, caso contrário, podem ser definidas medidas e coletados dados não alinhados às reais necessidades da organização. Devido à natureza das atividades relacionadas à medição de software, o uso de ferramentas computacionais é essencial. No entanto, considerando que a medição de software pode ser realizada no contexto de vários processos de software e que esses processos geralmente possuem diferentes ferramentas de apoio, é comum a utilização de várias ferramentas para que seja possível obter dados relacionados aos diferentes processos. Comumente, as ferramentas são desenvolvidas em diferentes momentos, por equipes diferentes e sem que haja preocupação com integração. Como resultado, as organizações têm que lidar com problemas de integração para permitir a devida comunicação entre as ferramentas e apoio ao processo de medição de software. Um fator chave para a integração de aplicações é possibilitar que elas compartilhem um entendimento comum do significado dos termos e serviços trocados. Em outras palavras, é importante que haja integração não só no nível sintático das aplicações, mas também no nível semântico. Entre os instrumentos utilizados para atribuição de semântica, as ontologias são reconhecidas como um importante meio para apoiar a integração semântica de aplicações. Elas podem ser usadas como uma interlíngua a fim de mapear os conceitos e serviços usados pelas diferentes aplicações. Considerando esse cenário, este trabalho explora a utilização de ontologias de domínio e de tarefa de medição de software na integração de ferramentas para apoiar o processo de medição de software. Para isso, é proposta uma especialização de OBA-SI (*Ontology-Based Approach for Semantic Integration*) (CALHAU, 2011) para auxiliar na integração de ferramentas de apoio à medição de software utilizando-se uma abordagem baseada em objetivos para o levantamento de requisitos para a integração. Para avaliar a abordagem proposta, ela foi utilizada para integrar ferramentas visando apoiar o processo de medição de software em uma organização de desenvolvimento de software real.

Palavras-chave: Medição de Software, Ontologia, Integração, Interoperabilidade, Semântica.

ABSTRACT

Software measurement is considered a fundamental practice to support process improvement and project management, since it provides useful data for decision making at organizational and project levels. In order to be effective, software measurement must be guided by organizational or project goals, otherwise measures can be defined and data can be collected without alignment with the actual organizational needs. Due to the nature of software measurement activities, the use of computational supporting tools is essential. However, considering that software measurement can be performed in the context of various software processes and these processes generally have different supporting tools, it is common to use several tools to allow collecting data regarding the processes. Tools are usually developed at different time, by different teams and without concern for integration. As a result, organizations have to deal with integration issues to enable the communication between tools and properly support the measurement process. A key factor for integration is the sharing of a common understanding between the tools regarding the meaning of the exchanged terms and services. In other words, it is important to deal with integration not only in the syntactic level, but also in the semantic level. Among the instruments used to address semantics, ontologies have been acknowledged as an important means to address tool semantic integration. They can be used as an interlingua to map concepts and services used by the different applications. Considering this scenario, this work explores the use of measurement domain and task ontologies for tool integration aiming to support the software measurement process. For that, OBA-SI (*Ontology-Based Approach for Semantic Integration*) (CALHAU, 2011) is specialized to help integration of software measurement supporting tools by using a goal-based approach for requirements elicitation. To evaluate the proposal, the approach was applied to integrate tools aiming to support the measurement process in an actual software development organization.

Keywords: Software Measurement, Ontology, Integration, Interoperability, Semantics.

SUMÁRIO

Capítulo 1 Introdução	12
1.1 Contexto.....	12
1.2 Motivação.....	13
1.3 Objetivos da Pesquisa.....	15
1.4 Método de Pesquisa.....	15
1.5 Organização da Dissertação.....	16
Capítulo 2 Medição de Software e Integração	17
2.1. Medição de Software	17
2.2. Ontologias de Medição de Software	20
2.2.1. Ontologia de Referência para Medição de Software (ORMS).....	21
2.2.1.1. <i>Ontologia de Processo de Software e Ontologia de Organização de Software</i>	23
2.2.1.2. <i>Subontologias de Entidades Mensuráveis, Medida e Definição Operacional de Medida</i>	25
2.2.1.3. <i>Subontologia Planejamento de Medição</i>	26
2.2.1.4. <i>Subontologia Medição</i>	27
2.2.1.5. <i>Subontologia Análise de Medição</i>	28
2.2.2. Ontologia de Tarefa de Medição de Software (OTMS)	28
2.2.2.1. <i>Modelo Comportamental Geral</i>	29
2.2.2.2. <i>Modelo Comportamental de Planejar Medição</i>	30
2.2.2.3. <i>Modelo Comportamental de Executar Medições</i>	32
2.2.2.4. <i>Modelo Comportamental de Analisar Medições</i>	32
2.3. Integração e Interoperabilidade.....	33
2.4. Ontology-Based Approach for Semantic Integration (OBA-SI).....	36
2.5. Considerações Finais do Capítulo	38
Capítulo 3 Investigação sobre Integração de Ferramentas para Apoiar Medição de Software	39
3.1 Visão Geral do Estudo.....	39
3.2 Revisão Sistemática da Literatura.....	41
3.2.1. Protocolo de Pesquisa	41
3.2.2. Síntese dos Dados.....	43
3.2.3. Discussões.....	55
3.3 Considerações Finais do Capítulo	58
Capítulo 4 Uma Abordagem Baseada em Ontologias para Integração de Ferramentas de Apoio à Medição de Software	60

4.1 Visão Geral da Abordagem Proposta.....	60
4.2 Processo de OBA-MSI.....	61
4.2.1. Levantar Requisitos da Integração	62
4.2.1.1. <i>Identificar e Alinhar o Processo de Medição</i>	63
4.2.1.2. <i>Identificar Objetivos, Necessidades de Informação e Medidas relevantes para a Organização</i>	67
4.2.1.3. <i>Identificar Ferramentas Fornecedoras de Dados e Serviços para a Medição de Software</i>	68
4.2.1.4. <i>Registrar Cenário de Integração</i>	71
4.2.2 Analisar Integração	71
4.2.2.1. <i>Integração nas Camadas de Serviço e Processo</i>	73
4.3 Considerações Finais do Capítulo	75
Capítulo 5 Aplicação da Abordagem Proposta	77
5.1 Integração de Ferramentas para Apoiar Medição de Software no LEDS.....	77
5.1.1 Definição do Processo de Medição alinhado à Ontologia de Tarefa de Medição de Software (OTMS).....	78
5.1.2 Identificação de Objetivos, Necessidades de Informação e Medidas Relevantes para a Organização.....	84
5.1.3 Identificação das Ferramentas Fornecedoras de Dados/Serviços para a Medição de Software.....	86
5.1.4 Registro do Cenário de Integração.....	89
5.1.5 Análise da Integração	89
5.1.5.1 <i>Selecionar Ontologia para a Integração</i>	89
5.1.5.2 <i>Modelos Conceituais das Ferramentas</i>	92
5.1.5.3 <i>Mapeamentos Verticais</i>	95
5.1.5.4 <i>Modelo Estrutural de Integração</i>	99
5.1.5.5 <i>Mapeamentos Horizontais</i>	101
5.1.6 Projeto, Implementação e Implantação da Integração.....	104
5.2 Avaliação da Integração de Ferramentas pelos Usuários do LEDS	110
5.2.1 Planejamento do Estudo.....	110
5.2.2 Resultados	113
5.2.2.1 <i>Percepção acerca do Processo de Medição</i>	113
5.2.2.2 <i>Adequação e Utilidade das Funcionalidades</i>	114
5.2.2.3 <i>Benefícios Obtidos</i>	115
5.2.2.4 <i>Dificuldades de Uso e Sugestões de Melhoria</i>	116

5.2.3 Entrevista com o Coordenador do LEDS	116
5.2.4 Discussões.....	117
5.2.5 Ameaças à Validade	118
5.3 Considerações Finais do Capítulo	119
Capítulo 6 Conclusão	121
6.1 Considerações Finais.....	121
6.2 Contribuições.....	125
6.3 Trabalhos Futuros.....	126
Referências Bibliográficas.....	128
Apêndice A Formulários Utilizados no Estudo Experimental.....	134
A1. Termo de Consentimento	134
A2. Perfil do Usuário.....	135
A3. Questionário de Avaliação.....	136

Capítulo 1

Introdução

Este capítulo apresenta o contexto, motivação e objetivos deste trabalho bem como a metodologia de pesquisa seguida e a organização do texto desta dissertação.

1.1 Contexto

Medição de software é um processo aplicado pelas organizações em vários contextos. Por exemplo, na gerência de projetos, é usada para desenvolver planos realísticos, monitorar o progresso dos projetos, identificar problemas e justificar decisões (MCGARRY *et al.*, 2002). Em iniciativas de melhoria de processos, medição apoia a análise do comportamento dos processos, permitindo a identificação de oportunidades de melhoria e predizendo se os processos serão capazes de alcançar os objetivos estabelecidos (FLORAC; CARLETON; BARNARD, 2000).

Fenton e Pfleger (1997) afirmam que medir produtos, processos e projetos de software é crucial para as organizações de software, pois as medidas quantificam as propriedades dessas entidades e permitem que se obtenha informações relevantes sobre o trabalho feito e o trabalho a ser feito. No contexto dos projetos de software, desenvolvedores podem usar a medição para verificar a consistência e completeza dos requisitos, a qualidade do projeto do sistema, o tamanho do código fonte, os defeitos, a cobertura dos testes, entre outros. Gerentes de projetos, por sua vez, podem aplicar a medição para prever quando um projeto será concluído e se o orçamento será suficiente. Clientes também podem se beneficiar de informações fornecidas pela medição. Por exemplo, medidas podem ser usadas para mostrar se o produto final está em conformidade com os padrões estabelecidos e satisfazem os requisitos acordados.

Organizações tipicamente utilizam diversas ferramentas¹ para apoiar seus processos de software. Por exemplo, ferramentas para elaboração e controle de cronograma e orçamento podem ser usadas para apoiar as atividades de gerência de projeto, ferramentas CASE (*Computer-Aided Software Engineering*) apoiam a engenharia de requisitos e ambientes de desenvolvimento apoiam a implementação e controle do código fonte. Apesar dessas ferramentas não serem normalmente concebidas para apoiar a medição de software, elas podem auxiliar a coleta e armazenamento de dados relacionados aos processos de software

¹ Neste trabalho os termos *ferramenta*, *aplicação* e *sistema* são utilizados como sinônimos.

apoiados (por exemplo, número de defeitos, tempo e custo despendido em atividades de projeto, números de linhas de código, percentual de falha dos testes, etc.).

Para que as ferramentas sejam capazes de fornecer dados consistentes e gerar informação útil para o processo de medição de software, elas devem ser integradas. Entretanto, os sistemas geralmente não são construídos com intuito de conversarem entre si. Na maioria das vezes são construídos isoladamente, em diferentes contextos, sem compartilharem um significado comum das informações manipuladas por eles. Devido a isso, tais sistemas implementam modelos de dados distintos que não são compartilhados, o que abre espaço para conflitos semânticos. A heterogeneidade entre os sistemas a serem integrados é considerada uma das maiores dificuldades no problema da integração (IZZAZ, 2009).

Os conflitos semânticos ocorrem quando o conteúdo dos itens de informação não possui o mesmo significado pretendido. Eles ocorrem sempre que dois sistemas não usam a mesma interpretação para um mesmo item de informação, ou seja, quando itens parecem possuir o mesmo significado, mas na verdade não possuem (WACHE *et al.*, 2001).

Para minimizar esses conflitos, a semântica deve ser tratada durante uma iniciativa de integração. Nesse contexto, ontologias podem ser usadas como uma interlíngua para mapear os conceitos usados pelos diferentes sistemas, de forma que sejam capazes de entender o significado dos dados e serviços (CALHAU, 2011).

Visando à qualidade, integração não deve ser feita de maneira *ad-hoc*. É importante a adoção de uma abordagem que ajude a lidar com a complexidade dessa tarefa, fornecendo, entre outros, etapas bem estabelecidas, separação de preocupações, diminuição da subjetividade e redução de problemas de entendimento e comunicação. Além disso, uma solução de integração mais completa deve integrar dados, serviços e processos.

1.2 Motivação

Apesar de as organizações utilizarem diversas ferramentas para apoiarem seus processos, um dos problemas enfrentados é a falta de ferramentas adequadas para apoiar o processo de medição (DE LUCIA; POMPELLA; STEFANUCCI, 2003). Como consequência, as organizações desenvolvem suas próprias soluções, que podem ser desde simples soluções baseadas em planilhas, até o desenvolvimento de sistemas dedicados a apoiar o processo de medição ou a integração de ferramentas com esse fim. Porém, soluções inadequadas podem comprometer a qualidade e a utilidade dos dados coletados e, conseqüentemente, o processo de medição (DUMKE; EBERT, 2007).

Integrar ferramentas para apoiar o processo de medição de software pode ser a solução adequada para várias organizações, permitindo a integração das diferentes fontes de dados e serviços, a fim de se obter informações relevantes à tomada de decisão. Entretanto, a integração de sistemas como um todo é uma tarefa complexa, que envolve diferentes tipos de preocupações. No âmbito da medição de software, uma das preocupações é garantir que as ferramentas sejam integradas de forma a prover informações realmente relevantes, ou seja, informações alinhadas aos objetivos organizacionais e dos projetos.

Um dos aspectos muitas vezes negligenciado nas iniciativas de integração é o tratamento da semântica das aplicações. Não levar em consideração a semântica durante uma iniciativa de integração é um problema sério, podendo ocasionar diversos problemas semânticos como os falsos acordos e diferenças de abstração (POKRAEV, 2009). Para tratar estes problemas, ontologias podem ser usadas a fim de estabelecer uma conceituação comum sobre um domínio, apoiando a comunicação e a integração das ferramentas (CALHAU; FALBO, 2010).

Entretanto, a solução de integração não envolve apenas a integração de conceituações de sistemas por meio de uma conceituação comum. Ela, como um tipo de produto de software, possui requisitos que devem ser levantados, analisados e satisfeitos. Além disso, a solução de integração deve ser projetada, implementada, testada e implantada. Isso sem falar que, principalmente em domínios mais complexos, pode ser que os próprios envolvidos na integração possuam visões diferentes do mesmo domínio e das tarefas sendo apoiadas (CALHAU, 2011).

Considerando-se a importância de se tratar a semântica, bem como a necessidade de orientar a integração das ferramentas, o uso de ontologias abordando a conceituação relativa ao domínio e ao processo de medição pode ser vantajoso. Ainda, levando-se em conta a existência da Ontologia de Referência de Medição de Software (ORMS) (BARCELLOS, 2009) e da Ontologia de Tarefa de Medição de Software (OTMS) (BARCELLOS; FALBO, 2013), vislumbrou-se a possibilidade de explorar o uso dessas ontologias na condução de iniciativas de integração de ferramentas visando ao apoio ao processo de medição de software.

Em (CALHAU, 2011) é proposta OBA-SI (*Ontology-Based Approach for Semantic Integration*), uma abordagem sistemática que utiliza ontologias na integração semântica de aplicações. OBA-SI pode ser aplicada para conduzir a integração de aplicações em qualquer domínio. No entanto, no contexto da medição de software, a integração de aplicações

apresenta algumas particularidades não tratadas em OBA-SI, dentre elas o fato de a integração tipicamente envolver ferramentas que não foram concebidas com o objetivo de apoiar medição, embora forneçam dados relevantes para tal. Sendo assim, torna-se necessário especializar OBA-SI para o domínio de medição de software para que a abordagem possa apoiar mais adequadamente iniciativas de integração nesse contexto.

1.3 Objetivos da Pesquisa

Este trabalho tem como *objetivo geral* **definir uma abordagem para a integração de ferramentas visando apoiar o processo de medição de software**. Esse objetivo geral pode ser detalhado nos seguintes *objetivos específicos*:

- (i) Analisar o estado da arte da integração de ferramentas no contexto de medição de software;
- (ii) Especializar OBA-SI para tratar aspectos relacionados à integração de ferramentas de apoio à medição de software;
- (iii) Aplicar a especialização de OBA-SI para realizar a integração de ferramentas visando apoiar o processo de medição de software em uma organização.

1.4 Método de Pesquisa

Este trabalho foi conduzido de acordo com as etapas a seguir:

- i) *Revisão da literatura*: nesta etapa ocorreu a aquisição de conhecimento sobre os temas relevantes para o trabalho, a saber: medição de software, ontologias e integração. Inicialmente, foi realizada uma revisão informal da literatura, onde a pesquisa por publicações relacionadas aos temas deste trabalho foi realizada de forma não sistemática, tendo sido lidos artigos, livros, dissertações, teses e relatórios técnicos considerados relevantes para o trabalho. Nesse momento não houve restrições quanto ao uso de mecanismos de busca nem ao formato das publicações, bastando o material ter reconhecimento científico. Após a revisão informal, foi realizada uma investigação formal da literatura por meio um mapeamento sistemático e uma revisão sistemática da literatura (KITCHENHAM; CHARTERS, 2007), em que foram investigadas iniciativas de integração de ferramentas de apoio à medição de software.
- ii) *Desenvolvimento da proposta*: nesta etapa ocorreu o desenvolvimento da especialização de OBA-SI, que resultou em OBA-MSI (*Ontology-Based Approach*

for Measurement Systems Integration), uma abordagem baseada em ontologias para integração de ferramentas de apoio à medição de software.

- iii) *Avaliação da proposta*: nesta etapa, a abordagem proposta foi utilizada para realizar a integração de ferramentas de apoio à medição de software em uma organização de desenvolvimento de software real, a saber o Laboratório de Extensão e Desenvolvimento de Sistemas (LEDS) do Instituto Federal do Espírito Santo (IFES) - Campus da Serra. Após a implantação, foi realizado um estudo para avaliar a percepção dos usuários sobre o resultado da integração.
- iv) *Escrita da dissertação*: os resultados obtidos durante a execução das etapas anteriores foram documentados nesta dissertação.

1.5 Organização da Dissertação

Este capítulo inicial apresentou as ideias principais desta dissertação descrevendo o contexto, motivação, objetivos e as etapas realizadas durante esta pesquisa. Além desta introdução, esta dissertação é composta pelos seguintes capítulos:

- **Capítulo 2 - Medição de Software e Integração**: aborda aspectos relevantes relacionados à medição de software e integração.
- **Capítulo 3 - Investigação sobre Integração de Ferramentas para Apoiar a Medição de Software**: apresenta os principais resultados do mapeamento sistemático e da revisão sistemática da literatura realizados.
- **Capítulo 4 - Uma Abordagem Baseada em Ontologias para Integração de Ferramentas de Apoio à Medição de Software**: apresenta OBA-MSI (*Ontology-Based Approach for Measurement Systems Integration*), a abordagem proposta neste trabalho.
- **Capítulo 5 - Aplicação da Abordagem Proposta**: apresenta os resultados da aplicação de OBA-MSI para integrar ferramentas de apoio à medição de software no LEDS. Os resultados de uma avaliação conduzida com os usuários do conjunto integrado de ferramentas também são apresentados.
- **Capítulo 6 - Conclusão**: apresenta as considerações finais deste trabalho, suas contribuições e propostas de trabalhos futuros.
- **Apêndice A - Formulários Utilizados no Estudo Experimental**: apresenta os formulários utilizados no estudo experimental realizado.

Capítulo 2

Medição de Software e Integração

Este capítulo tem como objetivo apresentar a fundamentação teórica para o trabalho realizado. A Seção 2.1 aborda Medição de Software. As ontologias de medição de software usadas neste trabalho são apresentadas na Seção 2.2. A Seção 2.3 trata aspectos relacionados à Integração e Interoperabilidade. A Seção 2.4 apresenta a abordagem para a integração semântica OBA-SI. Por fim, na Seção 2.5 são apresentadas as considerações finais do capítulo.

2.1. Medição de Software

Medição de software é o processo contínuo de definir, coletar e analisar dados relacionados aos processos e produtos de software, a fim de entendê-los, controlá-los e fornecer informação relevante para melhorá-los (SOLINGEN; BERGHOUT, 1999). Medição pode, também, ser definida como a atribuição de números ou símbolos para propriedades de entidades do mundo real com o propósito de descrevê-las (DUMKE; EBERT, 2007).

Medição de software é um dos principais processos de apoio à gerência de projetos e é crucial para a avaliação da qualidade de produtos de software e do desempenho e capacidade dos processos de software organizacionais (ISO/IEC, 2007).

No contexto da gerência de projetos, a medição auxilia no desenvolvimento de planos realísticos, no monitoramento do progresso do projeto, na identificação de problemas e no embasamento para a tomada de decisão (MCGARRY *et al.*, 2002). No contexto organizacional, a medição apoia a análise do desempenho dos processos e a identificação de ações para melhorar a competitividade das organizações (TARHAN; DEMIRÖRS, 2006).

Realizar medição de maneira efetiva contribui para o sucesso das organizações, uma vez que permite que elas entendam suas capacidades e elaborem planos alcançáveis a fim de produzir e entregar produtos e serviços. Medição também auxilia as organizações a detectarem tendências e se anteciparem a problemas, permitindo um melhor controle dos custos, reduzindo riscos, melhorando a qualidade e garantindo que os objetivos de negócio estabelecidos sejam alcançáveis (FLORAC; CARLETON, 1997).

Para obter os benefícios providos pela medição de software é necessário realizá-la adequadamente. A implementação de um programa de medição demanda esforço e envolve custos, especialmente nas fases iniciais, que requerem treinamento, elaboração dos

padrões e modelos a serem seguidos, definição dos mecanismos de coleta e implantação de ferramentas (DUMKE; EBERT, 2007).

A eficiência da medição de software está diretamente relacionada ao suporte provido pelas ferramentas adotadas (DUMKE; EBERT, 2007). Entretanto, apenas a utilização de ferramentas não garante o sucesso de um programa de medição (CARD; DOMZALSKI; DAVIES, 2008). O sucesso de um programa de medição está relacionado a dois fatores principais: (i) a coleta, análise e geração de relatórios devem estar diretamente relacionados à informação necessária para a tomada de decisão; e (ii) o processo de medição deve ser bem estruturado e repetível (MCGARRY *et al.*, 2002).

Existem diversos padrões e metodologias dedicadas a auxiliar as organizações a definir seu processo de medição de software, tais como ISO/IEC 15939 (ISO/IEC, 2007), PSM (*Practical Software Measurement*) (MCGARRY *et al.*, 2002) e IEEE Std. 1061 (IEEE, 2005). Além disso, padrões e modelos de maturidade relacionados à melhoria de processo de software, tais como a ISO/IEC 12207 (ISO/IEC, 2008), o CMMI (*Capability Maturity Model Integration*) (SEI, 2010) e o MR-MPS-SW (Modelo de Referência para a Melhoria do Processo de Software Brasileiro) (SOFTEX, 2012), também abordam o processo de medição de software. Embora haja algumas diferenças entre essas propostas, o processo de medição de software geralmente inclui: planejamento da medição, execução da medição e avaliação da medição (ISO/IEC, 2007).

Para realizar medição de software, inicialmente, uma organização deve planejá-la. Baseada em seus objetivos, a organização deve definir quais entidades (processos, produtos, etc.) devem ser consideradas para a medição de software e quais propriedades (tamanho, custo, tempo, etc.) devem ser mensuradas. A organização deve definir também quais medidas devem ser usadas para quantificar essas propriedades. Para cada medida, uma definição operacional de medida deve ser especificada, indicando, entre outras coisas, como a medida deve ser coletada e analisada. Uma vez planejada, pode-se executar a medição. A execução da medição envolve coletar dados para as medidas definidas, armazená-los e analisá-los. A análise de dados fornece informação para a tomada de decisão, apoiando a identificação de ações corretivas. Finalmente, o processo de medição e seus produtos devem ser avaliados visando identificar potenciais melhorias (BARCELLOS; FALBO; ROCHA, 2010).

Além dos padrões e metodologias que tratam do processo de medição de software como um todo, existem algumas propostas que lidam com aspectos mais específicos do processo de medição. Nesse contexto destaca-se o GQM (*Goal Question Metric*) (BASILI;

CALDIERA; ROMBACH, 1994). GQM é uma abordagem sistemática para a adaptação e integração de objetivos com processos, produtos e perspectivas de interesse de qualidade, baseado nas necessidades específicas dos projetos e da organização (BASILI; CALDIERA; ROMBACH, 1994). GQM considera três níveis:

- **Nível Conceitual (Objetivo):** Um objetivo é definido para um objeto, por uma variedade de razões, considerando vários modelos de qualidade, sob vários pontos de vista, relativos a um ambiente particular. Os objetos de medição incluem produtos (ex.: artefatos, especificações, programas), processos (ex.: manutenção, testes) e recursos (ex.: software, hardware, humano).
- **Nível Operacional (Questão):** Um conjunto de questões é usado para caracterizar o modo de avaliação do alcance a um objetivo. As questões tentam caracterizar o objeto de medição (produto, processo, recurso) em relação a critérios de qualidade selecionados e determinar sua qualidade do ponto de vista escolhido.
- **Nível Quantitativo (Medida):** Um conjunto de medidas é associado com cada questão, a fim de respondê-la de modo quantitativo.

Os níveis do GQM são organizados em uma estrutura hierárquica, iniciando-se com um objetivo. O objetivo é refinado em diversas questões, que normalmente dividem o objetivo em partes menores. Cada questão é, então, refinada em medidas. A mesma medida pode ser usada para responder diferentes questões. Vários modelos GQM podem apresentar questões e medidas em comum (BASILI; CALDIERA; ROMBACH, 1994).

Enquanto a elaboração do modelo GQM inicia-se de cima para baixo (*top-down*), os dados de medição são interpretados de baixo para cima (*bottom-up*). Uma vez que as medidas foram definidas com um objetivo explícito em mente, a informação provida por elas deve ser interpretada e analisada considerando esse objetivo, a fim de concluir se ele foi ou não alcançado (SOLINGEN; BERGHOUT, 1999). A Figura 2.1 ilustra a estrutura hierárquica do GQM.

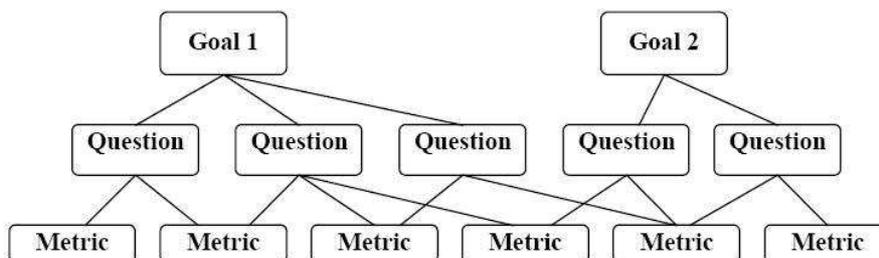


Figura 2.1 - Estrutura hierárquica do modelo GQM (BASILI; CALDIERA; ROMBACH, 1994).

Park *et al.* (1996) propõem uma variação do GQM incluindo indicador (*indicator*) como um item extra e levando ao GQ(I)M (vide Figura 2.2). O termo indicador significa a apresentação um ou mais resultados de medição projetados para comunicar ou explicar o significado dos demais resultados para o leitor (PARK; GOETHERT; FLORAC, 1996).

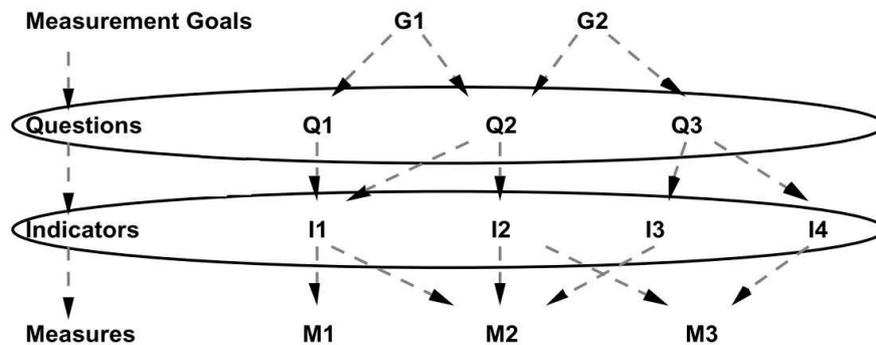


Figura 2.2 - Estrutura do GQ(I)M. Adaptado de (PARK; GOETHERT; FLORAC, 1996).

2.2. Ontologias de Medição de Software

Uma ontologia é uma especificação parcial, formal e explícita de uma conceituação compartilhada (GUARINO; STAAB; STUDER, 2009). O termo conceituação refere-se a um modelo abstrato de um fenômeno do mundo que identifica seus conceitos relevantes. Parcial diz respeito ao fato de apenas certos aspectos da realidade serem alvo de uma ontologia. Explícita significa que os conceitos usados e as restrições sobre seu uso são definidos explicitamente. Formal refere-se ao nível de expressividade da linguagem usada para descrever a ontologia. Linguagens lógicas são normalmente consideradas formais. Compartilhada reflete o fato de uma ontologia dever capturar o conhecimento consensual aceito por uma comunidade (GUARINO; STAAB; STUDER, 2009).

Na Engenharia de Software, ontologias têm sido tipicamente utilizadas para reduzir ambiguidades conceituais, tornar transparentes as estruturas de conhecimento, apoiar o compartilhamento de conhecimento e a interoperabilidade entre sistemas (USCHOLD; JASPER, 1999).

Existem diversas classificações para ontologias. Uma das mais conhecidas foi definida por Guarino (1998), que propõe quatro tipos de ontologias em uma classificação baseada no nível de generalidade: *Ontologia de Fundamentação*, *Ontologia de Domínio*, *Ontologia de Tarefa* e *Ontologia de Aplicação*. A Figura 2.3 ilustra os tipos de ontologias e os relacionamentos entre elas.

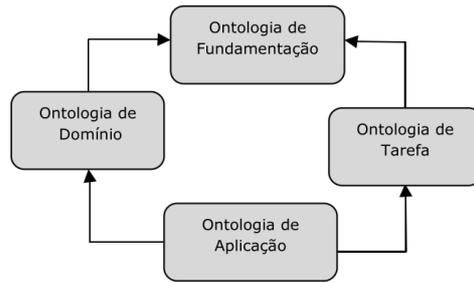


Figura 2.3 - Relacionamentos entre os tipos de ontologias (GUARINO, 1998).

Uma *Ontologia de Fundamentação* descreve conceitos gerais que são independentes de um problema ou domínio particular, definidos a partir de estudos da Filosofia, Linguística e Psicologia Cognitiva, tais como: espaço, tempo, matéria, objeto, evento e ação, dentre outros. *Ontologias de Domínio* e de *Tarefa* descrevem, respectivamente, o vocabulário relacionado a um domínio (por exemplo, medicina e turismo) ou a uma tarefa ou atividade específica (por exemplo, diagnóstico e venda) obtido pela especialização de termos introduzidos em uma ontologia de fundamentação. Por fim, uma *Ontologia de Aplicação* descreve conceitos que são dependentes de um domínio e de uma tarefa particulares e, assim, combina especializações de conceitos presentes nas ontologias de domínio e de tarefa. Esses conceitos correspondem a papéis desempenhados por entidades do domínio quando estão realizando determinada tarefa (GUARINO, 1998).

Neste trabalho foram utilizadas uma ontologia de domínio e uma ontologia de tarefa, ambas relacionadas à medição de software. Essas ontologias são descritas a seguir.

2.2.1. Ontologia de Referência para Medição de Software (ORMS)

Por ser considerada uma disciplina relativamente recente, ainda não foram estabelecidos padrões consensuais para a medição de software. Terminologias, conceitos, procedimentos e métodos de medição de software vêm sendo definidos na última década, porém, em particular, não há consenso para conceitos e terminologias, havendo duplicações e inconsistências nas propostas encontradas na literatura, inclusive nos termos mais comuns da área como medida, métrica e medição (BARCELLOS, 2009).

Buscando-se estabelecer uma conceituação de referência para o domínio de medição de software, em (BARCELLOS, 2009) foi definida a Ontologia de Referência para Medição de Software (ORMS). Uma ontologia de referência é uma ontologia de domínio construída com o objetivo de fazer a melhor descrição possível de um domínio na realidade, considerando um certo nível de granularidade e certo ponto de vista. Uma ontologia de referência de domínio é um tipo especial de modelo conceitual, representando

um modelo de consenso em uma comunidade. É uma especificação independente de solução, cujo objetivo é fornecer uma descrição clara e precisa das entidades do domínio com o propósito de comunicação, aprendizado e resolução de problemas. Idealmente, uma ontologia de referência deve ser construída considerando distinções ontológicas fundamentais, dadas por uma ontologia de fundamentação (GUIZZARDI, 2005).

O uso de ontologias de fundamentação na construção de ontologias de domínio permite tratar a ambiguidade de conceitos fundamentais, tornando explícito os compromissos ontológicos adjacentes das ontologias (GUIZZARDI, 2006). Nesse sentido, ORMS foi desenvolvida à luz da Ontologia de Fundamentação Unificada (*Unified Foundational Ontology* - UFO), uma ontologia de fundamentação desenvolvida com base em teorias das áreas de Ontologias Formais, Lógica Filosófica, Filosofia da Linguagem, Linguística e Psicologia Cognitiva (GUIZZARDI, 2005).

ORMS é subdividida em sete subontologias. Devido ao domínio de medição de software ser fortemente relacionado aos domínios de processos de software e de organizações, ORMS reutiliza conceitos das ontologias de Organização de Software (BARCELLOS; FALBO, 2009) e Processos de Software (BRINGUENTE; FALBO; GUIZZARDI, 2011). A Figura 2.4 apresenta um diagrama de pacotes com as subontologias de ORMS relevantes para este trabalho, as ontologias integradas e as relações entre elas. Em seguida, é feita uma breve descrição das subontologias. Considerando-se que a documentação de ORMS e das ontologias a ela integradas foi elaborada em inglês, optou-se por manter esse idioma nos diagramas aqui apresentados.

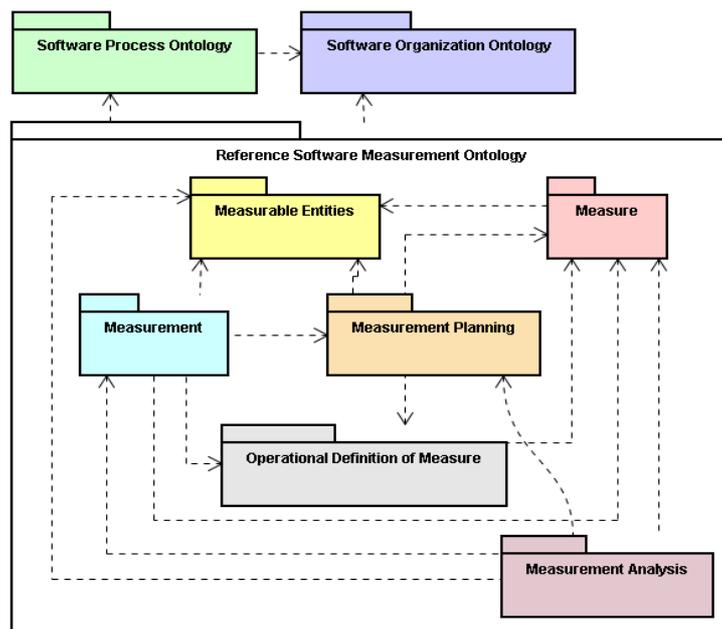


Figura 2.4 - Visão geral da Ontologia de Referência para Medição de Software.

- *Subontologia de Entidades Mensuráveis (**Measurable Entities**)*: trata das entidades que podem ser submetidas à medição e das suas propriedades que podem ser medidas.
- *Subontologia de Medida (**Measure**)*: trata da definição de medidas de software e do relacionamento das medidas com as entidades e elementos mensuráveis.
- *Subontologia de Definição Operacional de Medidas (**Operational Definition of Measure**)*: trata do detalhamento de aspectos relacionados à coleta e análise de medidas.
- *Subontologia de Planejamento de Medição (**Measurement Planning**)*: trata de aspectos relacionados ao planejamento da medição de software.
- *Subontologia de Medição (**Measurement**)*: trata da medição propriamente dita, ou seja, a coleta e armazenamento dos dados para as medidas.
- *Subontologia de Análise de Medição (**Measurement Analysis**)*: trata da análise dos dados coletados para as medidas para obtenção das informações de apoio às decisões.

ORMS é bastante extensa. Portanto, a seguir são apresentados apenas os fragmentos relevantes para este trabalho. As cores usadas nos pacotes das subontologias representados na Figura 2.4 são mantidas nos modelos conceituais apresentados ao longo de toda a dissertação com o intuito de identificar a origem dos conceitos presentes em cada modelo conceitual.

Apesar de ORMS ser baseada em UFO, discussões relacionadas à fundamentação dos conceitos estão fora do escopo deste trabalho. Assim, os fragmentos de ORMS aqui apresentados não incluem os conceitos de UFO nos quais os conceitos de ORMS foram baseados.

2.2.1.1. Ontologia de Processo de Software e Ontologia de Organização de Software

A Figura 2.5 ilustra um fragmento incluindo conceitos da Ontologia de Organização de Software e da Ontologia de Processo de Software usados em ORMS, e, em seguida, o modelo é descrito.

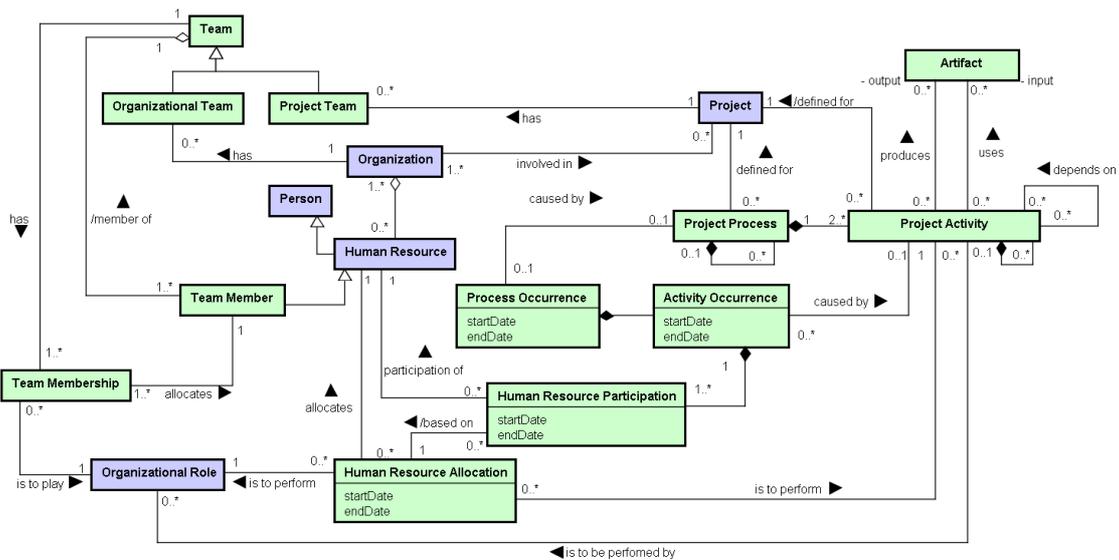


Figura 2.5 - Fragmento da Ontologia de Processo de Software e da Ontologia de Organização de Software.

Processo de Projeto (*Project Process*) é um processo definido para um Projeto (*Project*). Um Processo de Projeto é composto por Atividades de Projeto (*Project Activities*) que, por sua vez, podem ser compostas por outras. Uma Atividade de Projeto pode produzir ou usar Artefatos (*Artifact*). Uma Ocorrência de Processo (*Process Occurrence*) diz respeito a uma execução de processo e pode ser causada por um Processo de Projeto. Analogamente, uma Ocorrência de atividade (*Activity Occurrence*) é a execução de uma atividade e pode ser causada por uma Atividade de Projeto.

Recurso Humano (*Human Resource*) é uma Pessoa (*Person*) que faz parte de uma Organização (*Organization*). Por meio de uma Filiação a Equipe (*Team Membership*), um Recurso Humano passa a fazer parte de uma Equipe (*Team*) desempenhando um Papel Organizacional (*Organizational Role*) e torna-se Membro de Equipe (*Team Member*). Uma Equipe Organizacional (*Organizational Team*) é uma equipe definida para uma Organização. Uma Equipe de Projeto (*Project Team*), por sua vez, é definida para um Projeto.

Uma Alocação de Recurso Humano (*Human Resource Allocation*) diz respeito à alocação de um Recurso Humano para desempenhar um Papel Organizacional e realizar uma Atividade de Projeto. Por fim, uma Participação de Recurso Humano (*Human Resource Participation*) refere-se à participação de um Recurso Humano em uma Ocorrência de Atividade.

2.2.1.2. Subontologias de Entidades Mensuráveis, Medida e Definição Operacional de Medida

A Figura 2.6 ilustra um fragmento incluindo conceitos das subontologias de Entidades Mensuráveis, Medida e Definição Operacional de Medida. Em seguida, o modelo é descrito.

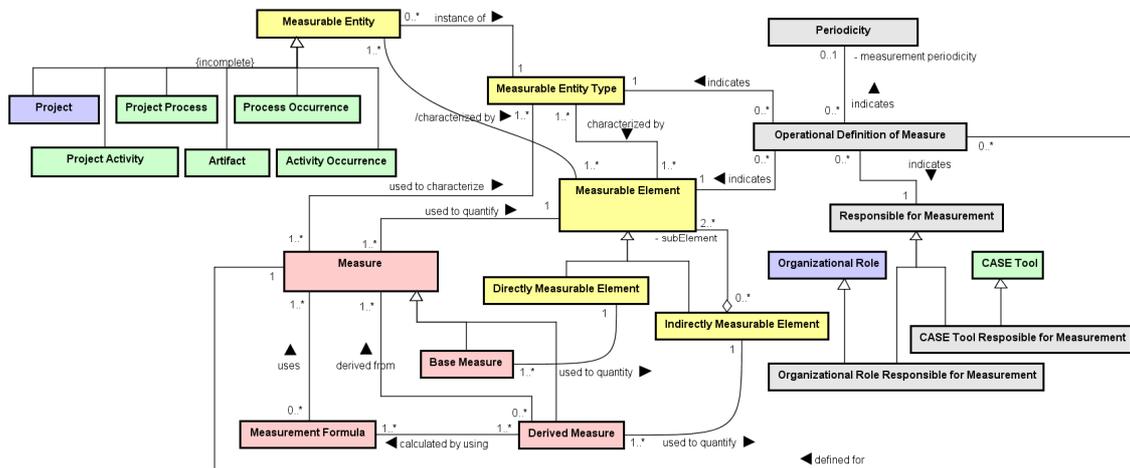


Figura 2.6 - Fragmento das subontologias Entidades Mensuráveis, Medida e Definição Operacional de Medida.

Uma Entidade Mensurável (*Measurable Entity*) é qualquer coisa que pode ser medida, como um Processo de Projeto (*Project Process*), um Artefato (*Artifact*) e um Projeto (*Project*). Entidades mensuráveis podem ser classificadas de acordo com o tipo (*Measurable Entity Type*) e são caracterizadas por Elementos Mensuráveis (*Measurable Elements*). Um Elemento Mensurável é uma propriedade de uma Entidade Mensurável que pode ser medida. *Tamanho* e *produtividade* são exemplos de elementos mensuráveis. Elementos podem ser medidos diretamente (*Directly Measurable Element*) (ex.: *tamanho*) ou indiretamente (*Indirectly Measurable Element*) (ex.: *produtividade*). Elementos Indiretamente Mensuráveis são medidos por meio de outros elementos mensuráveis, ditos seus subelementos. Entidades Mensuráveis de um mesmo tipo são caracterizadas pelos mesmos elementos.

Medidas (*Measure*) quantificam Elemento Mensuráveis. Medidas podem ser classificadas como Medidas Base (*Base Measure*), que são funcionalmente independentes de outras medidas (ex.: *número de requisitos do projeto*) e usadas para quantificar Elementos Diretamente Mensuráveis, e Medidas Derivadas (*Derived Measure*) (ex.: *taxa de alteração de requisitos, dada pela razão entre o número de requisitos alterados e o número de requisitos do projeto*), as quais são definidas em função de outras medidas e usadas para quantificar Elementos

Indiretamente Mensuráveis. Medidas Derivadas são calculadas através de fórmulas (*Measurement Formula*) que, por sua vez, utilizam outras Medidas para o cálculo.

Uma Definição Operacional de Medida (*Operational Definition of Measure*) detalha aspectos relacionados à coleta e análise de uma Medida, tais como a periodicidade da medição (*Measurement Periodicity*) e o responsável pela medição (*Responsible for Measurement*), que pode ser um Papel Organizacional (*Organizational Role Responsible for Measurement*) ou uma Ferramenta CASE (*CASE Tool Responsible for Measurement*), no caso de medições automáticas.

2.2.1.3. Subontologia Planejamento de Medição

A Figura 2.7 ilustra um fragmento da subontologia Planejamento de Medição, que é descrito em seguida.

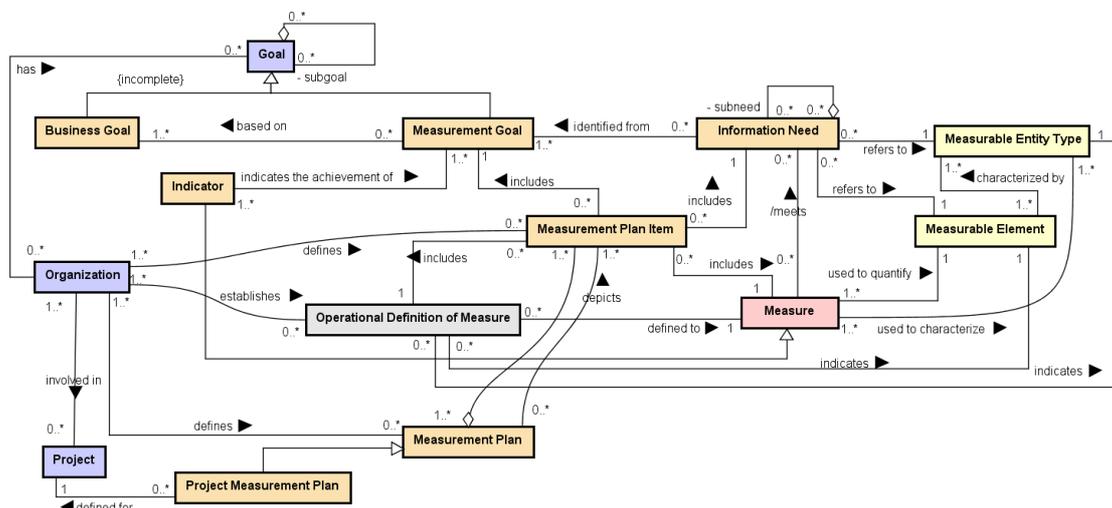


Figura 2.7 - Fragmento da subontologia Planejamento de Medição.

Um Objetivo de Negócio (*Business Goal*) é um tipo de Objetivo (*Goal*) que expressa a intenção pela qual ações estratégicas são planejadas e realizadas (por exemplo, *umentar o nível de satisfação dos clientes*). Um Objetivo de Medição (*Measurement Goal*) expressa a intenção pela qual ações relacionadas à medição são planejadas e realizadas. Objetivos de Medição devem ser definidos com base em Objetivos de Negócio (por exemplo, *reduzir o número de defeitos dos produtos entregues* é um objetivo de medição derivado do objetivo negócio *umentar o nível de satisfação dos clientes*). Uma Necessidade de Informação (*Information Need*) descreve a informação necessária para o monitoramento de Objetivos de Medição. *Qual a taxa de defeitos escapados? e qual a taxa de detecção de defeitos?* são exemplos de necessidades de informação identificadas a partir do objetivo de medição *reduzir o número de defeitos dos produtos entregues*. Necessidades de Informação dizem respeito a

Tipos de Entidades Mensuráveis e Elementos Mensuráveis (por exemplo, a necessidade de informação *quantos defeitos foram detectados após os testes?* diz respeito ao tipo de entidade mensurável *Código Fonte* e ao elemento mensurável *qualidade*). Um Indicador (***Indicator***) é o papel desempenhado por uma Medida quando ela é utilizada para monitorar o alcance a um Objetivo de Medição. Por exemplo, se a medida *taxa de defeitos escapados* é utilizada para indicar o alcance ao objetivo de medição *reduzir o número de defeitos dos produtos entregues*, ela desempenha o papel de indicador. Item de Plano de Medição (***Measurement Planning Item***) é o item básico do planejamento da medição. Associa um Objetivo de Negócio, um Objetivo de Medição, uma Necessidade de Informação, uma Medida e uma Definição Operacional de Medida. Um Plano de Medição (***Measurement Plan***) é composto por Itens de Plano de Medição. Uma Plano de Medição de Projeto (***Project Measurement Plan***) é um plano de medição definido para um Projeto.

2.2.1.4. Subontologia Medição

A Figura 2.8 ilustra um fragmento da subontologia Medição, que é descrito em seguida.

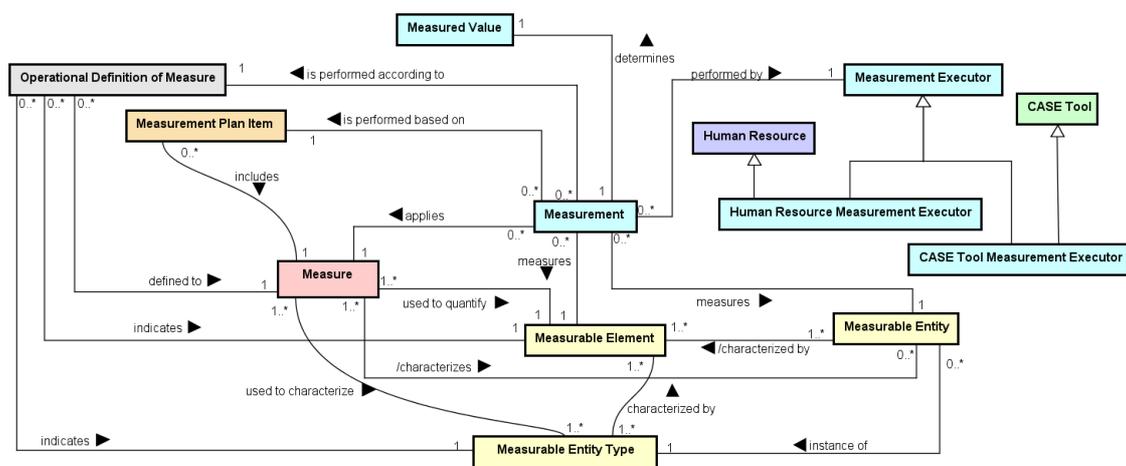


Figura 2.8 - Fragmento da subontologia Planejamento de Medição.

Uma Medição (***Measurement***) mede uma Entidade Mensurável aplicando-se uma Medida determinada em um Item de Plano de Medição e obtendo-se um Valor Medido (***Measured Value***). Por exemplo, a medição do *número de requisitos do projeto* para o *Projeto X* obtendo-se o valor medido *50*. Uma Medição é executada por um Executor de Medição (***Measurement Executor***), que pode ser um Recurso Humano (***Human Resource Measurement Executor***) ou uma Ferramenta CASE (***CASE Tool Measurement Executor***).

2.2.1.5. Subontologia Análise de Medição

A Figura 2.9 ilustra um fragmento da subontologia Análise de Medição, que é descrito em seguida.

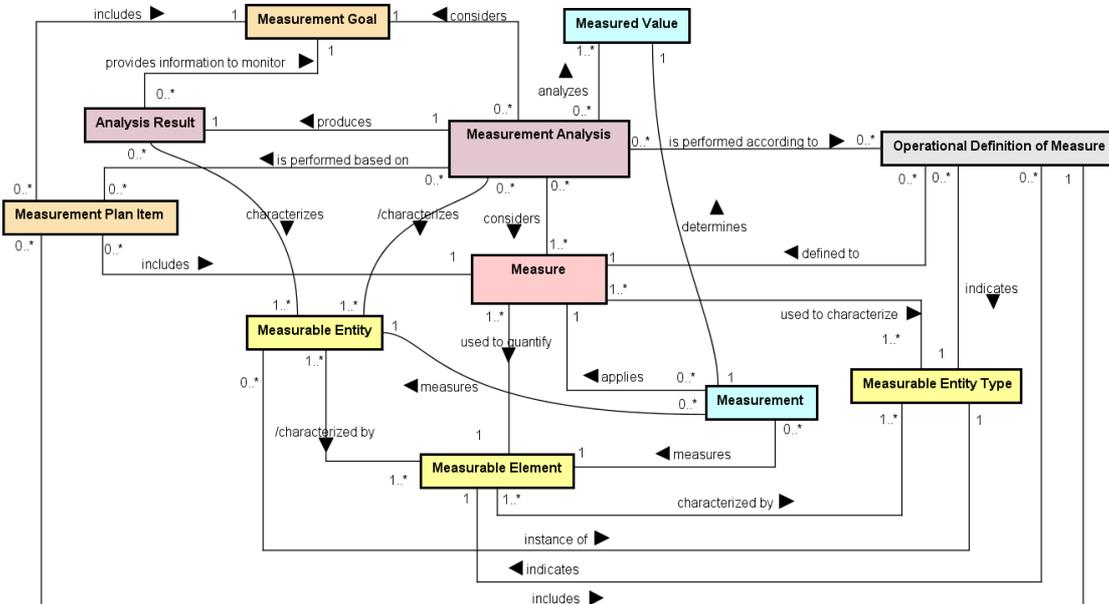


Figura 2.9 - Fragmento da subontologia Planejamento de Medição.

Uma Análise de Medição (*Measurement Analysis*) é o ato de analisar Valores Medidos para Medidas, buscando-se caracterizar Entidades Mensuráveis e obter um Resultado de Análise (*Analysis Result*) que forneça informação útil para monitorar um Objetivo de Medição. Por exemplo, a análise dos valores medidos para a medida *taxa de defeitos escapados* caracteriza o *código-fonte* e o resultado da análise (por exemplo, *há uma queda de 30% na taxa de defeitos escapados nos últimos 3 meses*) auxilia a monitorar o alcance ao objetivo *reduzir o número de defeitos dos produtos entregues*.

2.2.2. Ontologia de Tarefa de Medição de Software (OTMS)

A Ontologia de Tarefa de Medição de Software (OTMS) (BARCELLOS; FALBO, 2013) descreve as principais atividades do processo de medição de software, seus produtos e insumos. Foi desenvolvida com base em UFO (GUIZZARDI, 2005) e seu propósito é auxiliar a resolução de problemas de interoperabilidade semântica decorrentes da utilização conjunta de diferentes normas relacionadas com a medição, bem como apoiar integração semântica de aplicações que apoiam a medição de software (BARCELLOS; FALBO, 2013).

Ontologias de tarefa devem capturar dois tipos principais de conhecimento (MARTINS; FALBO, 2008): (i) o conhecimento comportamental, representando a decomposição das tarefas e fluxos de controle, e (ii) o conhecimento estrutural, que trata dos papéis desempenhados pelas entidades do domínio no cumprimento da tarefa. Dessa forma, modelos diferentes são necessários para representar tais conhecimentos. Seguindo a abordagem proposta em (MARTINS; FALBO, 2008), OTMS é representada por diagramas de atividades da UML, que tratam o aspecto comportamental da tarefa, e por diagramas de classes, que modelam os conceitos envolvidos na tarefa e seus relacionamentos.

OTMS foi desenvolvida buscando-se manter alinhamento com ORMS. Dessa forma, o modelo estrutural de OTMS é equivalente aos modelos conceituais de ORMS. Por essa razão, nesta seção são apresentados apenas os modelos comportamentais de OTMS e neles são representados apenas os conceitos relevantes para este trabalho.

Vale destacar que, embora seja usado o termo ontologia de tarefa, que é consagrado na área de ontologias, na verdade OTMS é uma ontologia de processo, uma vez que ela descreve o processo de medição como um todo e não apenas tarefas com baixo nível de granularidade (BARCELLOS; FALBO, 2013).

2.2.2.1. Modelo Comportamental Geral

O modelo comportamental geral de OTMS é apresentado na Figura 2.10.

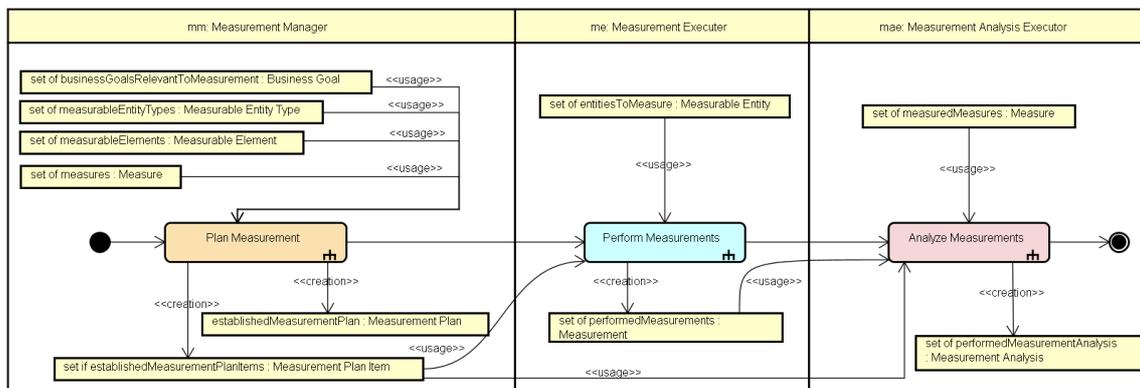


Figura 2.10 - Modelo Comportamental Geral de OTMS.

O processo de medição é composto por três atividades principais: Planejar Medição (*Plan Measurement*), Executar Medições (*Perform Measurements*) e Analisar Medições (*Analyze Measurements*). Na atividade Planejar Medição são definidos os Itens do Plano de Medição e estes são agrupados em um Plano de Medição. Em Executar Medições, os Itens do Plano de Medição são usados como base para a realização das medições. Por fim,

em Analisar Medições, as medições realizadas são analisadas a fim de se obter informações para o monitoramento de objetivos presentes nos Itens de Plano de Medição.

2.2.2.2. Modelo Comportamental de Planejar Medição

O modelo comportamental da atividade Planejar Medição é ilustrado na Figura 2.11. Na atividade Iniciar Planejamento de Medição (***Start Measurement Planning***) é definida a estrutura do Plano de Medição, dando-se início à sua elaboração. Em seguida, a partir de Objetivos de Negócio da organização são definidos Objetivos de Medição (***Establish Measurement Goals***) e estes são incluídos em Itens de Plano de Medição que compõem o Plano de Medição sendo elaborado. O próximo passo é identificar as necessidades de informação (***Identify Information Needs***) para cada objetivo de medição e incluí-las nos Itens de Plano de Medição e, conseqüentemente, no Plano de Medição. Na seqüência, para cada necessidade de informação, são identificadas as medidas (***Identify Measures***) capazes de atendê-las e estas são incluídas nos Itens do Plano de Medição. Por fim, a definição operacional de cada medida deve ser estabelecida (***Establish Operational Definition of Measure***), incluída nos respectivos Itens de Plano de Medição e a elaboração do Plano de Medição é concluída.

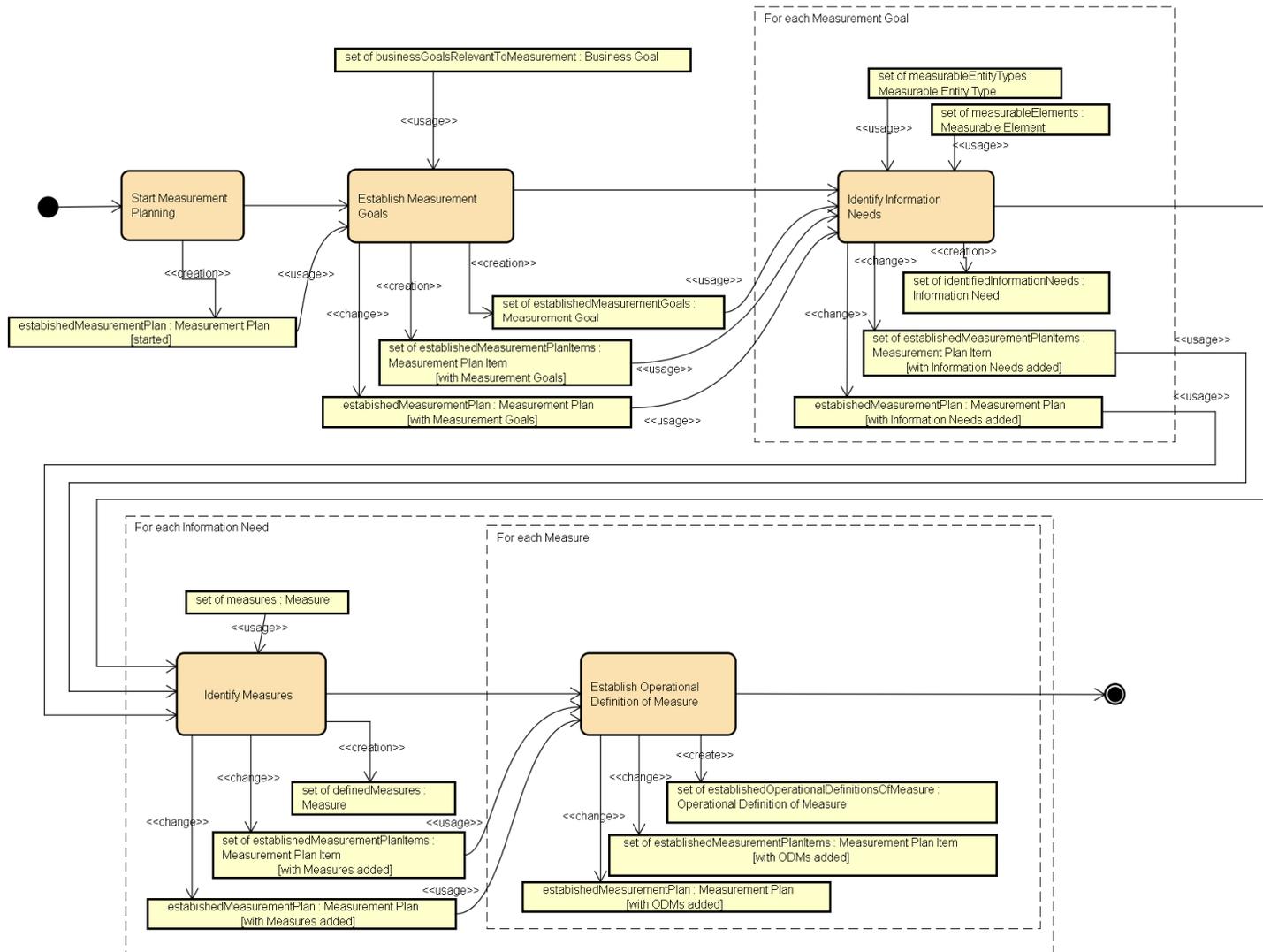


Figura 2.11 - Modelo Comportamental da atividade Planejar Medição.

2.2.2.3. Modelo Comportamental de Executar Medições

A Figura 2.12 ilustra o modelo comportamental da atividade Executar Medições.

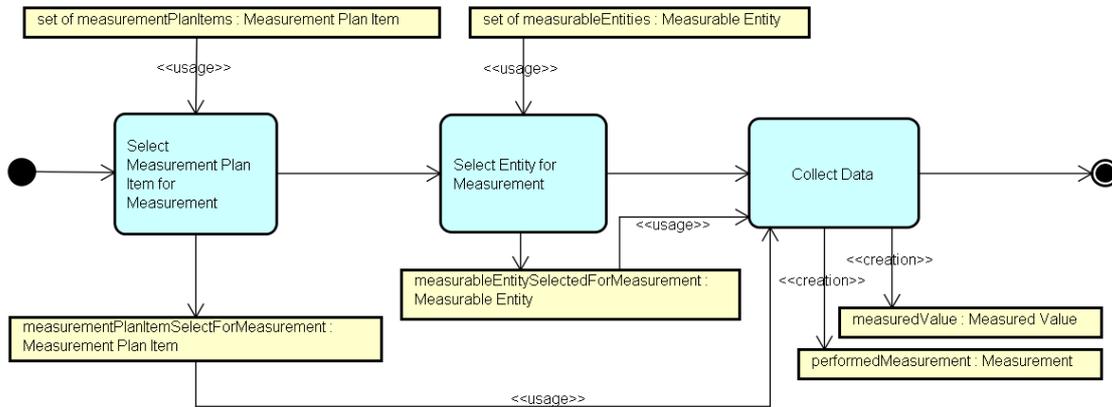


Figura 2.12 - Modelo Comportamental da atividade Executar Medições.

A primeira atividade para realizar uma medição é Selecionar Item do Plano de Medição para Medição (*Select Measurement Plan Item for Measurement*) para determinar a Medida para a qual deve ser realizada a medição. Em seguida, a Entidade Mensurável a ser medida deve ser selecionada (*Select Entity for Measurement*) e a coleta de dados pode ser realizada (*Collect Data*), obtendo-se o Valor Medido.

2.2.2.4. Modelo Comportamental de Analisar Medições

O modelo comportamental de Analisar Medições é ilustrado na Figura 2.13 é descrito em seguida.

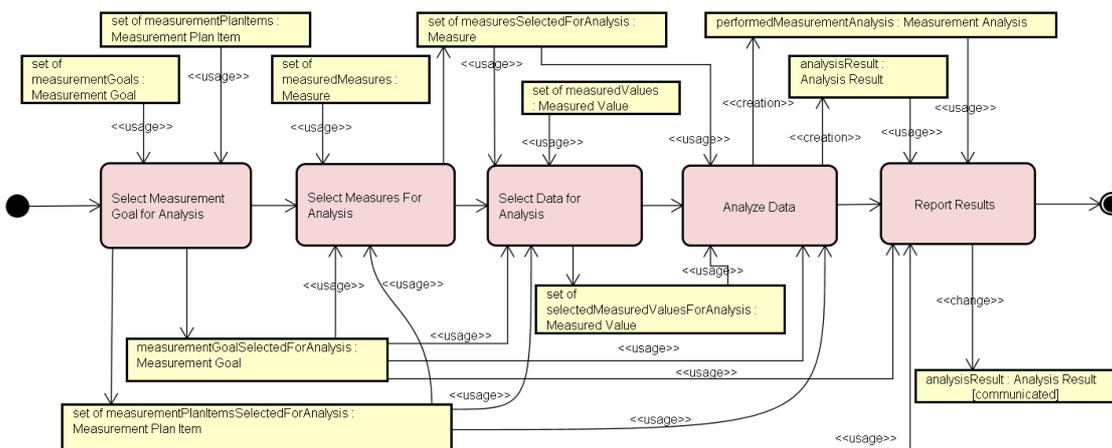


Figura 2.13 - Modelo Comportamental da atividade Analisar Medições.

Para realizar a análise de medições, primeiramente deve-se selecionar o Objetivo de Medição (*Select Measurement Goal for Analysis*) que se deseja monitorar com as

informações que serão providas pela análise a ser realizada. Esse objetivo deve ser selecionado dentre o conjunto de objetivos presentes nos Itens de Plano de Medição. Em seguida, dentre as Medidas relacionadas ao objetivo selecionado são selecionadas aquelas que serão analisadas (*Select Measures for Analysis*) e dentre os Valores Medidos para essas Medidas, deve-se selecionar aqueles que serão considerados na análise (*Select Data for Analysis*). Os valores medidos são, então, analisados (*Analyze Data*) e os resultados da análise são registrados e comunicados às partes interessadas (*Report Results*).

2.3. Integração e Interoperabilidade

Integração pode ser definida como o ato de incorporar componentes em um conjunto completo, conferindo a ele algumas propriedades esperadas. Os componentes são combinados de modo a formar um novo sistema, constituindo um todo e criando sinergia (IZZA, 2009).

Interoperabilidade, por sua vez, pode ser entendida como a habilidade de aplicações ou componentes de aplicação trocar dados e serviços (WEGNER, 1996). Ela permite que duas ou mais entidades de negócio (de uma mesma organização ou não) sejam capazes de trocar ou compartilhar informação, onde quer que estejam e a qualquer momento, e de usar funcionalidades uma da outra em um ambiente distribuído e heterogêneo. A interoperabilidade preserva os componentes de sistema como são (VERNADAT, 2007).

Devido à forte relação entre os termos integração e interoperabilidade, eles normalmente são usados como sinônimos (NARDI; FALBO; ALMEIDA, 2013). Neste trabalho, o termo integração é adotado em um sentido mais amplo, cobrindo integração e interoperabilidade.

As organizações têm usado um crescente número de aplicações para apoiar os seus processos. Em geral, essas aplicações são sistemas de software independentes, definidos de maneira isolada e operam autonomamente apoiando partes específicas de um processo de negócio (VERNADAT, 2007). Elas são baseadas em diferentes padrões, linguagens de programação, plataformas e sistemas operacionais, o que causa vários problemas de integração. Existe também a complexidade de aplicações legadas que, muitas vezes, apresentam estruturas fixas e rígidas para mensagens, interfaces e banco de dados. Além disso, existe uma lacuna de documentação. Muitos sistemas legados existem nas organizações por mais de 25 anos e sua documentação técnica não foi criada ou foi perdida no decorrer dos anos (THEMISTOCLEOUS; IRANI; LOVE, 2004).

Em resumo, as aplicações a serem integradas normalmente não foram projetadas para trabalhar em conjunto, ou seja, elas são heterogêneas, autônomas e distribuídas. Sistemas heterogêneos usam diferentes modelos de informação para capturar a semântica do domínio do negócio. Sistemas autônomos trocam dados seguindo seus protocolos de interação independentemente dos protocolos dos demais sistemas. Sistemas distribuídos não compartilham um estado em comum e usam diferentes meios para atualizar ou obter seu estado (POKRAEV, 2009).

A integração pode ser efetuada considerando diferentes dimensões. Izza (2009) propôs um *framework* que sintetiza abordagens de integração por meio de quatro principais dimensões: escopo, ponto de vista, camada e nível. A Figura 2.14 ilustra as dimensões de integração.

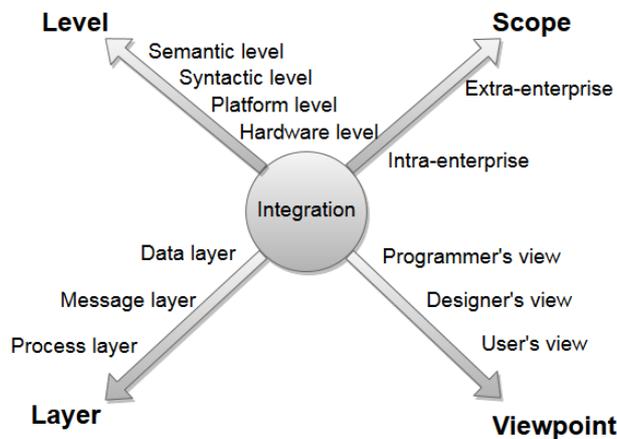


Figura 2.14 - Dimensões de integração (IZZA, 2009).

A dimensão de *escopo* distingue duas principais abordagens: intraorganizacional (*Intra-enterprise*) e interorganizacional (*Extra-enterprise*). A integração intraorganizacional trata de cenários que envolvem aplicações internas de uma mesma organização, enquanto a integração interorganizacional trata de cenários que visam conectar aplicações de diferentes organizações (IZZA, 2009).

Na dimensão de *ponto de vista*, são considerados: ponto de vista do usuário (externo) (*User's view*), o qual se preocupa com as diferentes visões dos especialistas de domínio e usuários de negócio; ponto de vista do projetista (conceitual) (*Designer's view*), o qual aborda os diferentes modelos usados durante o projeto dos sistemas de informação; e ponto de vista do programador (interno) (*Programmer's view*), que se refere à implementação dos sistemas de informação (IZZA, 2009).

A dimensão de *camadas* permite classificar uma integração em uma ou mais camadas dos sistemas de informação. A camada de dados (*Data layer*) lida com a movimentação de

dados entre múltiplas bases de dados. A integração nessa camada ignora a lógica da aplicação e manipula os dados diretamente no banco de dados, através da sua interface nativa. A integração na camada de mensagem ou serviço (*Message layer*) trata das trocas de mensagens entre as aplicações integradas. Qualquer componente de uma aplicação (por exemplo, interface de usuário, lógica de aplicação e banco de dados) pode originar ou consumir mensagens. A integração na camada de processo (*Process layer*) enxerga as organizações como um conjunto de processos inter-relacionados e é responsável por tratar o fluxo de mensagens, implementar regras e definir a execução geral do processo. Essa constitui a mais complexa abordagem de integração (IZZAZ, 2009).

A dimensão de *níveis* considera quatro níveis: hardware (*Hardware level*), plataforma (*Platform level*), sintático (*Syntactic level*) e semântico (*Semantic level*). O nível hardware cobre as diferenças entre hardwares de computadores, redes, etc. O nível plataforma engloba as diferenças nos sistemas operacionais, plataforma de banco de dados, etc. O nível sintático trata o modo como o modelo de dados e as assinaturas das operações são escritos. Por fim, o nível semântico lida com o significado pretendido dos conceitos em um esquema de dados ou assinatura da operação. Cada nível depende do nível anterior, então não é possível considerar a semântica se a sintaxe ainda não tiver sido considerada (IZZAZ, 2009).

Os desafios da integração de aplicações surgem, dentre outros, do fato de que as aplicações heterogêneas empregam diferentes modelos de dados e comportamentais, levando a conflitos semânticos. Esses conflitos ocorrem quando as aplicações são construídas com conceituações diferentes, o que pode impactar na integração de dados, serviços e processos (NARDI; FALBO; ALMEIDA, 2013).

Segundo Pokraev (2009), os problemas de interoperabilidade semântica ocorrem principalmente devido a falsos acordos ou então diferenças na abstração. O primeiro problema ocorre basicamente quando símbolos estabelecidos como iguais são usados para representar coisas diferentes no mundo real ou símbolos estabelecidos como diferentes são usados para representar a mesma coisa. Já a diferença de abstração ocorre devido à diferença de representação de um mesmo conceito ou de seus relacionamentos.

Para evitar conflitos semânticos, o significado da informação que é trocada tem que ser entendido entre os sistemas que serão integrados (WACHE *et al.*, 2001). Nesse contexto, ontologias podem auxiliar a descrever a semântica das fontes de informação e tornar o conteúdo explícito (WACHE *et al.*, 2001). Elas podem ser usadas como uma interlíngua para mapear os conceitos usados pelos diferentes sistemas, de forma que eles

entendam os dados e serviços por meio de uma ontologia compartilhada (CALHAU, 2011).

Considerando os benefícios de utilização de ontologias para evitar problemas semânticos em iniciativas de integração de sistemas, em (CALHAU, 2011) foi proposta OBA-SI (*Ontology-Based Approach for Semantic Integration*), que é brevemente apresentada na próxima seção.

2.4. *Ontology-Based Approach for Semantic Integration (OBA-SI)*

A Abordagem baseada em Ontologias para Integração Semântica de Sistemas (*Ontology-Based Approach for Semantic Integration - OBA-SI*) (CALHAU, 2011) considera o processo de integração como um tipo de processo de desenvolvimento de software, composto pelas fases de levantamento de requisitos, análise, projeto, implementação, testes e implantação. OBA-SI se concentra na fase de análise da integração, na qual a semântica deve ser definida. É importante que se tenha concordância semântica antes do projeto e implementação de qualquer solução, de modo que a integração no nível conceitual seja independente de tecnologia e de uma solução de integração específica. Para isso, modelos conceituais estruturais e comportamentais dos sistemas são comparados à luz de ontologias de domínio e de tarefa, usadas para atribuir semântica aos itens compartilhados entre os sistemas. Relações entre os modelos são obtidas por meio de mapeamentos entre seus elementos. Esse processo é apresentado na Figura 2.15.

O processo de integração de OBA-SI inicia-se com a fase de *levantamento de requisitos da integração*, quando os requisitos e objetivos devem ser estabelecidos. Nessa fase também é necessário indicar as atividades do processo de negócio que serão apoiadas, os sistemas a serem integrados para apoiá-las, os domínios de aplicação envolvidos no cenário de integração e as tarefas genéricas relacionadas à integração. Com isso, define-se o cenário de integração. Em seguida, é realizada a fase de *análise da integração*, que é responsável por analisar e modelar os requisitos da integração, especificar as funcionalidades que serão providas e os conceitos envolvidos, bem como estabelecer como será o comportamento geral do conjunto integrado de sistemas. Seu principal produto é o modelo da integração (CALHAU, 2011). Esse modelo é utilizado para estabelecer as equivalências semânticas entre os sistemas. A Figura 2.16 apresenta as atividades envolvidas nessa fase.

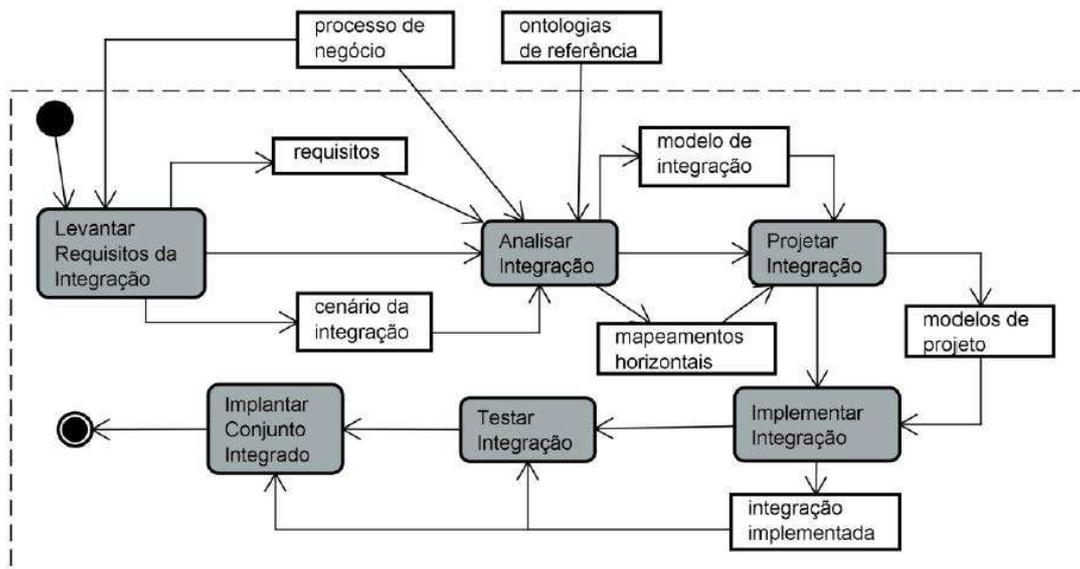


Figura 2.15 - Processo de Integração Semântica (CALHAU, 2011).

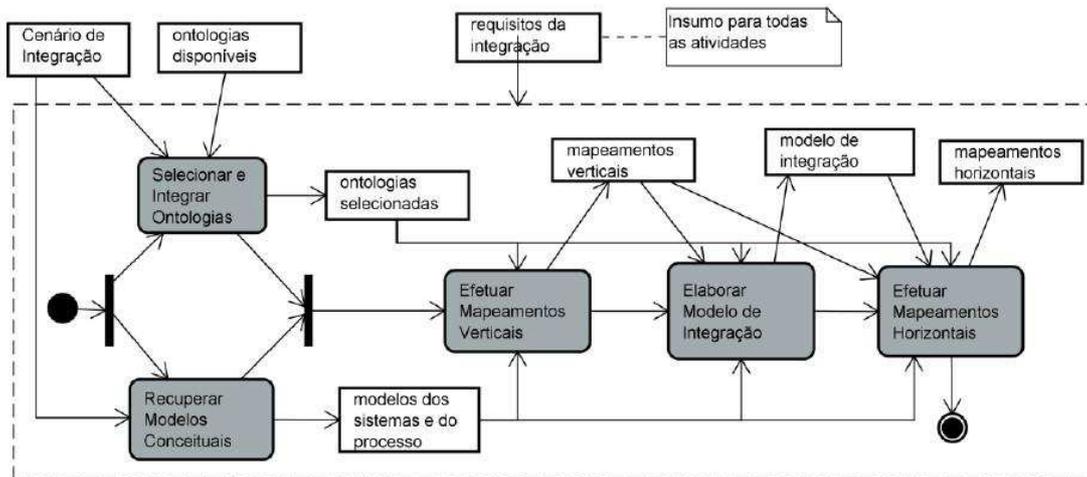


Figura 2.16 - Atividades da Fase de Análise da Integração (CALHAU, 2011).

A análise da integração se inicia com a obtenção dos modelos conceituais estruturais e comportamentais dos sistemas listados no cenário de integração. Seleciona-se uma ontologia de referência sobre o domínio do cenário de integração que, caso não esteja disponível, deve ser desenvolvida. Posteriormente, ocorre a definição dos mapeamentos verticais entre os elementos dos modelos dos sistemas e elementos da ontologia selecionada, tendo como objetivo atribuir semântica aos modelos conceituais. Inicialmente, mapeiam-se os conceitos, e em seguida, devem-se definir os mapeamentos entre as relações. Uma vez estabelecidos os mapeamentos, o modelo de integração é construído baseado na ontologia e nos modelos das ferramentas, de maneira que cada elemento dos modelos conceituais dos sistemas que estejam sem mapeamento vertical recebam um significado. Em seguida, cada elemento dos modelos conceituais dos sistemas deve ser

mapeado para um elemento correspondente do modelo de integração. Todos elementos dos modelos conceituais dos sistemas devem ser mapeados. Geram-se, assim, os mapeamentos horizontais de conceitos e relações.

Por fim, ocorrem as fases de *projeto e implementação da integração*. Há diversas maneiras de se obter uma solução de projeto e implementação de integração, e por isso, OBA-SI não se compromete com nenhuma solução específica de integração, porém propõe algumas diretrizes para que a semântica estabelecida na fase de análise se mantenha. As ferramentas podem ser integradas sem que as mesmas sejam alteradas. Nesse contexto, é proposta a utilização de mediadores responsáveis por interligar as ferramentas, de forma a ter uma visão global dos sistemas as serem integrados.

O modelo de integração é fundamental para o mediador, por fornecer as informações para projetá-lo, e os mapeamentos horizontais podem ser usados para projetar a comunicação entre as ferramentas e o mediador. A implementação dessa comunicação pode ocorrer dentro do mediador ou fora dele, por meio de adaptadores que conectam as ferramentas ao mediador (BRINGUENTE, 2011).

2.5. Considerações Finais do Capítulo

Para tratar de assuntos considerados importantes ao entendimento deste trabalho, este capítulo apresentou o conteúdo relacionado a medição de software e integração de aplicações. Foram apresentadas a Ontologia de Referência para Medição de Software e a Ontologia de Tarefa de Medição de Software, bem como OBA-SI, uma abordagem baseada em ontologias para integração semântica de aplicações. Esses três trabalhos são fundamentais no contexto da abordagem proposta neste trabalho.

No próximo capítulo são apresentados os principais resultados de uma revisão sistemática na qual foram investigadas iniciativas de integração de ferramentas para apoiar medição de software, a fim de se obter uma visão do estado da arte relacionado ao tópico de pesquisa explorado neste trabalho.

Capítulo 3

Investigação sobre Integração de Ferramentas para Apoiar Medição de Software

Este capítulo descreve os resultados da investigação formal da literatura sobre integração de ferramentas de apoio à medição de software. A Seção 3.1 apresenta uma visão geral do estudo realizado, a Seção 3.2 descreve o estudo realizado e os resultados obtidos e a Seção 3.3 apresenta as considerações finais deste capítulo.

3.1 Visão Geral do Estudo

Buscando-se investigar iniciativas que envolvam a integração de ferramentas para apoiar a medição de software, foi feita uma investigação formal da literatura, começando-se por uma revisão terciária, que foi seguida por um mapeamento sistemático e uma revisão sistemática da literatura. Uma revisão terciária é um estudo que investiga estudos secundários sobre um determinado tópico de pesquisa. Estudos secundários, por sua vez, investigam estudos primários. As revisões sistemáticas da literatura e os mapeamentos sistemáticos são exemplos de estudos secundários (KITCHENHAM; BUDGEN; BRERETON, 2011).

Um mapeamento sistemático fornece uma visão ampla de uma área de pesquisa a fim de determinar se existe evidência de pesquisa em um tópico particular (KITCHENHAM; CHARTERS, 2007). Mapeamentos sistemáticos auxiliam a identificar lacunas e sugerir áreas para pesquisas futuras, fornecendo um mapa que permite posicionar apropriadamente novas atividades de pesquisa. Além disso, os resultados de mapeamento sistemático podem indicar áreas adequadas para execução de revisões sistemáticas da literatura (KITCHENHAM; BUDGEN; BRERETON, 2011).

Revisões sistemáticas da literatura (RSL) são estudos secundários usados para encontrar, avaliar criticamente e agregar os artigos científicos relevantes a uma questão de pesquisa ou tópico de pesquisa específico (KITCHENHAM; BUDGEN; BRERETON, 2011). RSLs são consideradas meios para identificar, avaliar e interpretar a pesquisa relevante a uma questão de pesquisa particular, área tópico ou fenômeno de interesse (KITCHENHAM; CHARTERS, 2007).

Os estudos conduzidos neste trabalho seguiram a abordagem definida em (KITCHENHAM; CHARTERS, 2007), a qual é composta de três principais atividades: (i)

planejamento, quando o protocolo de pesquisa é definido com o propósito de prover reprodutibilidade ao estudo, bem como auxiliar os pesquisadores a evitar o viés durante a condução da revisão; (ii) execução, quando o protocolo é executado e os dados são extraídos, analisados e armazenados; e (iii) publicação, quando os resultados são registrados e disponibilizados para os potenciais interessados.

As seguintes bibliotecas digitais foram pesquisadas durante os estudos:

- IEEE Xplore (<http://ieeexplore.ieee.org>)
- ACM Digital Library (<http://dl.acm.org>)
- Springer Link (<http://www.springerlink.com>)
- Scopus (<http://www.scopus.com>)
- Science Direct (<http://www.sciencedirect.com>)
- Engineering Village (<http://www.engineeringvillage.com>)

Para a revisão terciária foi elaborada uma expressão de busca contendo quatro grupos de termos conectados com o operador *AND*. O primeiro grupo inclui termos relacionados à integração e interoperabilidade. O segundo grupo inclui termos relacionados à medição de software. O terceiro grupo inclui termos relacionados a ferramentas e aplicações. O quarto grupo inclui termos relacionados a mapeamentos sistemáticos e RSLs. Dentro dos grupos foi usado o operador *OR* para permitir sinônimos. A seguinte expressão de busca foi utilizada:

("integration" OR "interoperability" OR "interoperable" OR "integrated") AND ("software measurement" OR "software process measurement" OR "software project measurement" OR "software engineering measurement" OR "software product measurement") AND ("tool" OR "application" OR "system" OR "framework" OR "suite" OR "toolkit") AND ("systematic literature review" OR "systematic review" OR "systematic mapping" OR "mapping study" OR "systematic literature mapping")

A expressão de busca foi aplicada em três campos de metadados (título, resumo e palavras-chave) e 60 publicações foram encontradas. Em seguida, foi adotado o seguinte critério de seleção: a publicação apresenta uma revisão sistemática da literatura ou um mapeamento sistemático sobre integração de ferramentas para apoiar a medição de software. Contudo, nenhuma das publicações atendeu o critério.

Não tendo sido encontrado nenhum estudo secundário sobre o tópico de pesquisa investigado, decidiu-se por realizar um mapeamento sistemático da literatura, no qual 12 iniciativas envolvendo integração de ferramentas para apoiar a medição de software foram encontradas. As características gerais dessas iniciativas foram analisadas e alguns problemas foram identificados. Os principais resultados do mapeamento foram publicados em

(FONSECA; BARCELLOS; FALBO, 2015). Os problemas identificados levaram a novas questões de pesquisa, indicando a necessidade de uma investigação mais profunda das propostas selecionadas. Com o objetivo de responder às novas questões de pesquisa, uma revisão sistemática da literatura foi realizada. A mesma expressão de busca e critérios de seleção foram usados no mapeamento sistemático e na revisão sistemática da literatura. Entretanto, novas questões de pesquisa foram estabelecidas para a revisão sistemática, que é descrita na próxima seção.

3.2 Revisão Sistemática da Literatura

3.2.1. Protocolo de Pesquisa

Nesta seção é apresentado o protocolo de pesquisa usado para execução da revisão sistemática da literatura (RSL).

- a) **Objetivo da RSL:** o objetivo dessa RSL é investigar iniciativas envolvendo a integração de ferramentas para apoiar a Medição de Software.
- b) **Questões de Pesquisa:** Para alcançar o objetivo da RSL, foi definida uma questão de pesquisa principal para ser respondida: *Quais são as iniciativas de integração de ferramentas para apoiar a medição de software presentes na literatura?* A partir dessa questão principal, foram definidas quatro questões de pesquisa (QP) específicas relacionadas a três aspectos principais: Medição, Ferramentas e Integração.

QP1 (Medição) - Quais atividades do processo de Medição de Software (Planejamento da Medição, Coleta de Dados e Análise dos Dados) são apoiadas pelo conjunto integrado de ferramentas e como ocorre o apoio? O propósito desta questão é identificar quais atividades de medição são apoiadas pelas iniciativas, a fim de avaliar a cobertura do conjunto de ferramentas integradas, bem como explicar como esse apoio é fornecido. As atividades de medição consideradas são as duas primeiras atividades definidas em (ISO/IEC, 2007) (planejamento da medição e execução da medição). A execução da medição foi dividida para permitir verificar se as ferramentas apoiam tanto a coleta de dados (que envolve a coleta e o armazenamento de dados) quanto a análise dos dados, ou se apenas uma dessas atividades.

QP2 (Medição) - Quais medidas são consideradas pela iniciativa de integração e quais os principais processos de software medidos por elas? Esta questão visa identificar quais medidas são consideradas pelas iniciativas e os principais processos de software medidos por elas,

permitindo analisar o quão específico ou abrangente é o escopo da medição, bem como os principais processos focados pela iniciativa de integração.

QP3 (Ferramentas) - Quais são as ferramentas integradas e quais atividades do processo de Medição de Software elas apoiam? O objetivo desta questão é identificar as ferramentas integradas em cada iniciativa e quais atividades de medição elas apoiam.

QP4 (Integração) - Como a integração das ferramentas é feita e como ela é categorizada com relação às dimensões de escopo, ponto de vista, camada e nível? Esta questão visa descrever e categorizar cada iniciativa de integração considerando as quatro dimensões propostas pelo *framework* de Izza (2009): escopo, ponto de vista, camada e nível.

- c) **Expressão de Busca:** a expressão de busca a seguir foi aplicada nas bibliotecas digitais citadas na seção anterior. Como pode ser notado, a expressão de busca resultou da adaptação da expressão usada na revisão terciária, excluindo-se o quarto grupo de termos (relacionado a mapeamentos sistemáticos e a RSLs):

("integration" OR "interoperability" OR "interoperable" OR "integrated") AND ("software measurement" OR "software process measurement" OR "software project measurement" OR "software engineering measurement" OR "software product measurement") AND ("tool" OR "application" OR "system" OR "framework" OR "suite" OR "toolkit")

- d) **Seleção de Publicações:** a seleção foi efetuada em cinco etapas:

Etapa 1 - Seleção Primária e Catalogação: a expressão de busca foi aplicada nos mecanismos de pesquisa das bases selecionadas. O tipo de publicação foi limitado para artigos da área de Ciência de Computação e Engenharia. Ao final dessa etapa, 948 publicações foram encontradas.

Etapa 2 - Remoção de Duplicações: estudos indexados por mais de uma biblioteca digital foram identificados e as duplicações foram removidas. 85 publicações foram removidas no total, resultando em 863 estudos selecionados ao final desta etapa.

Etapa 3 - Seleção de Publicações Relevantes - 1º Filtro: o título, resumo e palavras-chave das publicações selecionadas foram analisados considerando os seguintes critérios de inclusão (CI) e exclusão (CE): (CI1) a publicação apresenta informação relacionada a integração de ferramentas, aplicações ou sistemas que apoiam a medição de software; (CE1) a publicação não tem um resumo; (CE2) a publicação está publicada apenas como um resumo; (CE3) a publicação não é um estudo primário. Como resultado dessa etapa, 24 estudos foram selecionados (uma redução de aproximadamente 97%).

Etapa 4 - Seleção de Publicações Relevantes - 2º Filtro: o texto completo das publicações selecionadas na *Etapa 3* foi lido com o propósito de identificar aqueles que forneciam

informação útil. Dessa forma, o critério de inclusão CI1 foi considerado e também os seguintes critérios de exclusão: (CE4) a publicação não está escrita em inglês; (CE5) o texto completo da publicação não está disponível; e (CE6) a publicação é uma cópia ou uma versão anterior de uma publicação já considerada. 8 estudos foram selecionados ao final dessa etapa.

Etapa 5 - Snowballing: Como sugerido em (KITCHENHAM; CHARTERS, 2007), as referências das publicações selecionadas no estudo devem ser analisadas e, se alguma delas apresentar evidência relacionada ao tópico de pesquisa, ela deve ser avaliada pelos critérios de seleção e incluída no estudo. Assim, nessa etapa, as referências das publicações selecionadas na etapa anterior foram investigadas aplicando-se o primeiro e segundo filtros. Como resultado, 4 novas publicações foram selecionadas.

A Figura 3.1 ilustra o processo seguido para a seleção das publicações, que resultou em 12 publicações selecionadas.

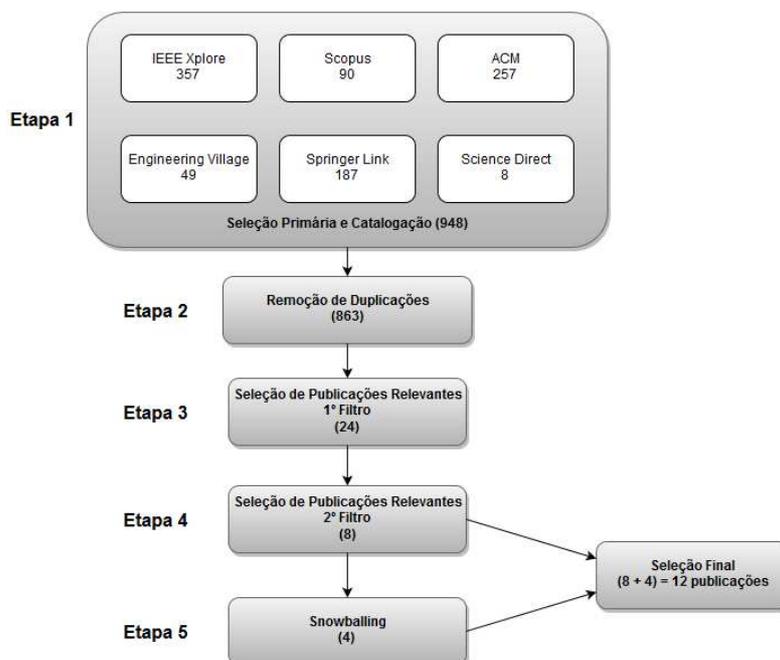


Figura 3.1 - Processo de seleção de publicações.

3.2.2. Síntese dos Dados

Nessa seção são apresentados os principais resultados obtidos considerando-se cada questão de pesquisa (QP).

QP Principal - Quais são as iniciativas de integração de ferramentas para apoiar a medição de software presentes na literatura?

A Tabela 3.1 apresenta as doze iniciativas identificadas pela RSL, respondendo à questão principal.

Tabela 3.1 - Iniciativas de integração de ferramentas que apoiam medição de software.

Proposta	Descrição
TAME (BASILI; ROMBACH, 1988)	TAME (<i>Tailoring A Measurement Environment</i>) é um ambiente integrado de engenharia de software composto por vários componentes integrados. TAME integra três ferramentas de medição que capturam dados de código Ada e geram medidas.
Tool support for SM (TIAN; TROSTER; PALMA, 1997)	Esta iniciativa usa um conjunto de ferramentas integradas a fim de apoiar a medição de software e a melhoria da qualidade. Uma ferramenta que apoia a análise baseada em modelagem de árvores (S-PLUS) é a ferramenta central. As ferramentas são conectadas à S-PLUS como consumidoras ou fornecedoras de informações.
GQM Tool (LAVAZZA, 2000)	Esta proposta apresenta a ferramenta GQM tool, que apoia a definição de medidas, coleta de dados, análise e <i>feedback</i> . Ela possui uma interface com um sistema de gerência de configuração e outras ferramentas de medição.
MetriFlame (KOMI-SIRVIO; PARVIAINEN; RONKAINEN, 2001)	MetriFlame é uma ferramenta para automação da medição baseada em GQM que usa dados registrados durante o processo de desenvolvimento de software. Ela possui componentes para coletar dados de medição de várias ferramentas, planilhas e bases de dados.
DSS (CHULANI <i>et al.</i> , 2003)	DSS (<i>Decision Support System</i>) é um sistema de suporte à decisão desenvolvido pela IBM para usar medidas de software com o objetivo de permitir os executivos tomarem decisões mais fundamentadas. Ela captura (de diferentes sistemas) dados relacionados ao suporte ao cliente, situações críticas e satisfação do cliente, integrando esses dados em um <i>data warehouse</i> .
SM in a CI Environment (MOREIRA <i>et al.</i> , 2010)	Esta abordagem usa um motor de Integração Contínua (IC) para automatizar a extração dos dados de medição. Ela segue a área de processo <i>Measurement and Analysis</i> do CMMI e conceitos do GQ(I)M para selecionar as medidas mais relevantes. A coleta dos dados é feita por várias ferramentas.
SOFAS (GHEZZI; GALL, 2011)	SOFAS é uma plataforma que oferece serviços de análise de medidas a fim de permitir a interoperabilidade entre ferramentas de análise. Ela é constituída de três principais componentes: <i>Software Analysis Web Services</i> , que fornece um catálogo de serviços para análise de dados; <i>Software Analysis Broker</i> , que age como o gerenciador de serviços e faz interface entre os serviços e os usuários; e o <i>Software Analysis Ontologies</i> , que define e representa os dados consumidos e produzidos pelos diferentes serviços.
Dione (CAGLAYAN <i>et al.</i> , 2012)	Dione é uma aplicação Java web cujas principais funcionalidades são: i) construir um repositório de medição que contenha medidas de produto e de processo bem como informações sobre componentes de software com defeitos; ii) analisar medidas para identificar tendências e problemas usando configurações de gráficos e relatórios; e iii) construir e calibrar modelos de predição de defeitos customizados para predizer a propensão a defeitos de uma versão do produto de software. Dione coleta dados de várias ferramentas e usa <i>smart clients</i> para se conectar com os artefatos de desenvolvimento de software e extrair medidas automaticamente. Ela também suporta a integração com outras ferramentas através de serviços web.
QualitySpy (JURECZKO; MAGOTT, 2012)	QualitySpy é um <i>framework</i> para monitoramento do processo de desenvolvimento de software. Ele coleta dados brutos de várias ferramentas integradas, bem como de código fonte, e fornece relatórios de análise.
3C (JANUS <i>et al.</i> , 2012)	3C Approach é uma extensão à prática de IC que adiciona <i>Continuous Measurement</i> e <i>Continuous Improvement</i> como atividades subsequentes à IC. Diversas ferramentas Java e um sistema de controle de versão foram integrados no motor de IC Cruise Control, permitindo a coleta de medidas relacionadas à código fonte e cobertura de testes.
ASSIST (KESER; IYIDOGAN; OZKAN, 2013)	ASSIST é uma ferramenta integrada desenvolvida por uma organização CMMI nível 3. Adota a abordagem GQ(I)M e é conectado com softwares comerciais para gerência de projetos, rastreamento de problemas e planejamento de recurso empresarial (ERP).
DePress (MADEYSKI; MAJCHRZAK, 2014)	DePress é um <i>framework</i> de código aberto e extensível para medição de software e integração de dados que podem ser usados para predição (ex.: predição de defeitos, predição de esforço) e análise de mudanças de software (ex.: alterações da versão, estatísticas de defeitos). Suporta o uso integrado (através do <i>framework</i> KNIME) dos sistemas de rastreamentos de problema JIRA e Bugzilla, dos sistemas de gerência de configuração Subversion e Git, e das ferramentas de medição Judy, JaCoCo, EclipseMetrics, CheckStyle e PMD.

QP1 (Medição) - Quais atividades do processo de Medição de Software (Planejamento da Medição, Coleta de Dados e Análise dos Dados) são apoiadas pelo conjunto integrado de ferramentas e como ocorre o apoio?

A atividade de *planejamento da medição* é apoiada por quatro dos doze estudos analisados (*TAME*, *GQM Tool*, *MetriFlame* e *ASSIST*), sendo que todos usam GQM ou GQ(IM). *TAME* fornece *templates* para a definição e refinamento de objetivos em questões e medidas. *GQM Tool* permite a edição dos objetivos definidos através de formulários predefinidos e é capaz de verificar a consistência estrutural dos planos de medição (por exemplo, verificando se cada questão está conectada com um objetivo e foi refinada em medidas). *ASSIST* usa GQ(IM) e inclui um conjunto de objetivos-questões-indicadores-medidas bem estruturados que podem ser reutilizados. *MetriFlame* não apresenta detalhes sobre a utilização do GQM, mas seus autores argumentam que a ferramenta é capaz de gerenciar planos de medição baseados em GQM. Em *SM in a CI Environment*, o planejamento da medição é realizado pelo uso de GQ(IM), entretanto, é feito manualmente antes do uso do conjunto integrado. Portanto, considerou-se que essa iniciativa não apoia o planejamento da medição.

Diferente do planejamento da medição, a atividade de *coleta de dados* é apoiada por todos os estudos. Todas as iniciativas apoiam a coleta de dados através das ferramentas integradas, as quais funcionam como ferramentas para entrada de dados. *Tool Support for SM*, *GQM Tool*, *MetriFlame*, *DSS*, *SM in a CI Environment*, *SOFAS*, *QualitySpy* e *3C* obtêm dados de medição apenas de ferramentas externas. *TAME*, *Dione* e *ASSIST* também fornecem mecanismos para a entrada manual de dados como resultado do apoio integrado ao processo de medição.

A *análise de dados* também é apoiada por todos os estudos. As iniciativas apresentam funcionalidades para a análise que variam de relatórios simples a ferramentas de análise sofisticadas. *TAME*, *MetriFlame*, *QualitySpy*, *3C* e *DePress* apresentam ferramentas de relatórios simples, isto é, incluem um módulo ou uma ferramenta que gera gráficos limitados e fixos ou relatórios para visualizar os resultados das medições. *Tool Support for SM*, *GQM Tool*, *DSS*, *SM in a CI Environment*, *SOFAS*, *Dione* e *ASSIST*, por sua vez, apresentam ao menos uma ferramenta de análise sofisticada, fornecendo visualizações, gráficos e relatórios flexíveis e dinâmicos para os dados de medição coletados.

QP2 (Medição) - Quais medidas são consideradas pela iniciativa de integração e quais os principais processos de software medidos por elas?

A Tabela 3.2 lista as medidas consideradas em cada proposta e os principais processos de software medidos. Algumas propostas focam em um único processo, enquanto outras focam em mais de um. Existem também iniciativas nas quais os processos medidos não são definidos previamente e dependem dos dados disponíveis no conjunto integrado de ferramentas.

Tabela 3.2 - Medidas e os Principais Processos de Software Medidos.

Proposta	Medidas	Principais Processos de Software
TAME	Linhas de código, medidas de complexidade estrutural, medidas de ligação de dados, cobertura de testes.	Codificação
Tool Support for SM	Número de defeitos corrigidos, complexidade de código, medidas internas, medidas de código de ligações de modelagem preditiva, chegada de falhas, tempo de execução, tempo das instâncias de falhas, números de testes executados, número de transações processadas, confiabilidade estimada (número de sucessos sobre o número de testes executados), taxa de sucesso prevista, medidas relacionadas a <i>design</i> , tamanho, mudanças, defeitos, testes e transações.	Codificação
GQM Tool	Número total de falhas; para cada falha: prioridade, tipo, momento da detecção, momento de conclusão; Número total de defeitos; para cada defeito: severidade, tipo, fase em que originou, momento da detecção, momento da correção; número ciclomático; número de classes; métodos por classe; LOC; linhas de comentários; tamanho; complexidade média e tamanho por componente.	Codificação
MetriFlame	Depende dos dados disponíveis nas ferramentas, banco de dados e planilhas integradas.	Depende dos dados disponíveis.
DSS	Natureza do problema, Severidade, Solução do problema fornecida, Capacidade, Facilidade de uso, Desempenho, Confiabilidade, Facilidade de instalação, Manutenibilidade, Documentação, Serviço/Suporte, Satisfação geral.	Gestão de Clientes
SM in a CI Environment	Número de linhas de código, Número de instruções, Número de métodos, Número de campos, Complexidade Ciclométrica, Complexidade Ciclométrica II, <i>Ranking</i> de tipos, Falta de coesão dos métodos, Número de filhos, Profundidade da árvore de herança, Número de linhas de comentários, Percentual de comentários, Acoplamento aferente no nível de tipo, Acoplamento eferente no tipo de nível, Associação entre classes, Percentual de cobertura (de testes unitários).	Codificação
SOFAS	Fan-In e Fan-Out de classes, métodos e pacotes; Complexidade ciclométrica de McCabe para classes, métodos e pacotes; Número de chamadas no sistema; Tamanho da árvore de herança de classes; Média da altura da hierarquia; Média de derivados; Número de subclasses diretas; Número de métodos sobrescrevendo qualquer método da superclasse; Número de classes; Número de pacotes; Número de atributos (estáticos e não estáticos) de classes e pacotes; Número de métodos (estáticos e não estáticos) de classes e pacotes; Números de parâmetros dos métodos.	Codificação
Dione	Complexidade ciclométrica por mês, tamanho em LOC do projeto no decorrer do tempo, Quantidade de defeitos e densidade de defeitos por mês, Média da complexidade ciclométrica por mês.	Codificação
QualitySpy	19 medidas relacionadas a código calculadas pela ferramenta CKJM extended.	Codificação
3C	Número de testes, Cobertura de testes, Taxa de Crescimento dos Testes, Número de <i>builds</i> quebradas, Total de linhas, Linhas efetivas, Violações <i>Checkstyle</i> , violações de regras de prioridade 1/2/3 de <i>Findbugs</i> , violações de regras de prioridade 1/2/3 de <i>PMD</i> .	Codificação e Testes

Tabela 3.2 - Medidas e os Principais Processos de Software Medidos (cont.).

Proposta	Medidas	Principais Processos de Software
ASSIST	Medidas relacionadas a projetos (ex.: valores atuais e estimados), a produtos (ex.: tamanho funcional, tamanho de código, propriedades técnicas), ao processo de desenvolvimento (ex.: medidas da equipe de desenvolvimento) e outras medidas definidas durante o GQ(I)M, de acordo com os dados disponíveis.	Gerência de Projetos e Codificação
DePress	Número de defeitos, Número de defeitos únicos por arquivo, Defeitos após a entrega, CK Java Metrics, Cobertura de código, Tempo gasto nas atividades atribuídas.	Gerência de Configuração e Codificação

A maioria das medidas consideradas pelas iniciativas são medidas relacionadas a código fonte. Como consequência, a Codificação é o processo de software focado pela maioria das propostas. Algumas das iniciativas consideram apenas medidas relacionadas a código (ex.: *SOFAS*). Outras consideram também medidas relacionadas a outros processos (ex.: *TAME* e *SM in a CI Environment* também apresentam medidas relacionadas a Testes), mas o principal processo medido é o de Codificação. Ainda, algumas propostas dividem o foco entre Codificação e outro processo de software. *ASSIST* mede principalmente os processos de Gerência de Projetos e Codificação, *DePress* foca em Codificação e Gerência de Configuração, e *3C* foca em Codificação e Testes.

MetriFlame considera medidas relacionadas ao processo de desenvolvimento de software, contudo as medidas não foram especificadas pelos autores, uma vez que elas dependem dos dados disponíveis pelas ferramentas integradas. Conseqüentemente, não é possível identificar quais processos relacionados ao desenvolvimento de software (ex.: Codificação, Gerência de Projetos) são medidos, porque isso depende das medidas consideradas. *ASSIST* também permite definir medidas de acordo com os dados disponíveis. Nesse sentido, dependendo do contexto de aplicação, *ASSIST* poderia medir outros processos além daqueles apontados na Tabela 3.2.

As medidas consideradas são usadas para diversos propósitos. Muitas delas são usadas para apoiar a melhoria do processo de software. Isso ocorre em *TAME*, *GQM Tool*, *SM in a CI Environment*, *QualitySpy* e *ASSIST*. Em *Tool Support for SM*, as medidas permitem a avaliação da qualidade de software baseada em modelos de crescimento de confiabilidade. Em *DSS*, medidas relacionadas a clientes são usadas para fornecer uma visão do cliente para os executivos do negócio responsáveis por múltiplos produtos de software. *SOFAS* aplica medidas para analisar a qualidade de serviços. Em *Dione* e *DePress*, medidas são usadas para apoiar a predição de defeitos. Em *3C*, medidas apoiam a garantia da qualidade de produtos de software desenvolvidos usando métodos ágeis. Por fim, *MetriFlame* usa medidas para avaliar os processos de desenvolvimento de software e produtos.

QP3 (Ferramentas) - Quais são as ferramentas integradas e com quais atividades do processo de Medição de Software elas apoiam?

A Tabela 3.3 apresenta as ferramentas envolvidas em cada iniciativa e a respectiva atividade de medição apoiada. Após a Tabela 3.3, o apoio à medição fornecido pelas ferramentas em cada proposta é descrito.

Tabela 3.3 - Ferramentas de Medição e Atividades de Medição de Software Apoiadas.

Proposta	Ferramentas de Medição e Atividades de Medição de Software Apoiadas		
	Planejamento da Medição	Coleta de Dados	Análise de Dados
TAME	TAME	TAME, ferramenta de cobertura, ferramenta de ligação de dados, ferramenta de medição de código	TAME
Tool Support for SM	-	IDSS, CMVC, TestLog, SlaveDriver, REFINE, W-Analyzer	S-PLUS, SMERFS, SAS, TreeBrowser
GQM Tool	GQM tool	GQM tool, Oracle, PCMS, Krakatau, Resource Standard Metrics	GQM tool, MS Access
MetriFlame	MetriFlame	MetriFlame, Lotus Notes, Paradox, dBASE, IBM DB/2, Informix, Interbase, MS Access, MS SQL Server, Oracle, Sybase, FoxPro, Microsoft Project, Microsoft Excel	MetriFlame
DSS	-	Ferramentas legadas	Decision Support System
SM in a CI Environment	-	Cruise Control.Net, Nant, NUnit, PartCover, NDepend, MS Access	MS Excel, MS SQL Server Analysis Services
SOFAS	-	SOFAS, CVS, Subversion, Git, Bugzilla, Google Code, Trac, SourceForge	SOFAS
Dione	-	Dione, CVS, Subversion, Git, Mercurial, Clearcase, Bugzilla, Jira	Dione
QualitySpy	-	QualitySpy, CKJM extended, Selenium, Jira, Subversion, Hudson	QualitySpy
3C	-	Subversion, Findbugs, Checkstyle, PMD, Cobertura, Cruise Control, JUnit	Cockpit
ASSIST	ASSIST	ASSIST, sistema ERP, ferramenta de rastreamento de problemas, ferramenta de gerência de projetos	ASSIST
DePress	-	Jira, Bugzilla, Subversion, GIT, Judy, JaCoCo, EclipseMetrics, CheckStyle, PMD, DePress	DePress, KNIME Report Designer

TAME: Nesta iniciativa, *TAME* é a ferramenta principal e apoia as três atividades de medição. A arquitetura de *TAME* é constituída de vários componentes. Os componentes *GQM Model Selection* e *GQM Model Generation* apoiam o planejamento da medição ao permitirem a criação e reuso de modelos GQM. Os componentes *Measurement Scheduling*, *Measurement Tools* e *Data Entry and Validation* apoiam a coleta de dados. Eles permitem, respectivamente, agendar a coleta automática de dados, coletando dados de processo e produto automaticamente de três ferramentas (ferramenta de cobertura,

ferramenta de ligação de dados e ferramenta de medição de código), e a entrada de dados manualmente. Os componentes *GQM Analysis and Feedback* e *Report Generator* apoiam a análise de dados. O primeiro permite realizar análises de acordo com um modelo de GQM específico e o segundo oferece uma variedade de relatórios.

Tool Support for SM: Essa proposta não apoia o planejamento da medição. Ela usa ferramentas da IBM (IDSS - *Integrated Development Support System* e CMVC - *Configuration Management/Version Control*) e aplicações internas (*TestLog* e *SlaveDriver*) para apoiar a coleta de dados dos bancos de dados de projetos. *REFINE*, um kit de engenharia reversa, é usado para calcular medidas de projeto e de complexidade do código fonte e uma ferramenta chamada *W-Analyzer* computa medidas de produtos. A análise de dados é apoiada por quatro ferramentas: *S-PLUS*, uma ferramenta para análise de modelagem em árvore que age como a ferramenta de análise central; *SMERFS*, que é usada quando modelos de análise adicionais são necessários; *SAS*, um pacote estatístico comercial que é usado para modelagem estatística em geral; e *TreeBrowser*, uma ferramenta interna que facilita a análise através de exploração de árvores.

GQM Tool: *GQM Tool* apoia o planejamento da medição permitindo a criação de planos GQM. Para auxiliar a coleta de dados, as ferramentas *PCMS*, *Krakatau* e *Resource Standard Metrics* agem como provedores de dados, e *MS Access* e *Oracle* são usados para armazenar os dados. A análise dos dados é apoiada por *GQM Tool*, que usa o poder computacional do sistema de busca do *MS Access* para executar pesquisas associadas com os itens do plano GQM e exibe os resultados.

MetriFlame: Essa proposta usa a ferramenta *MetriFlame* como a ferramenta central. Ela apoia a criação de planos GQM (planejamento da medição) e coleta dados de várias fontes de dados (coleta de dados): *Lotus Notes*, *Paradox*, *dBASE*, *IBM DB/2*, *Informix*, *Interbase*, *MS Access*, *MS SQL Server*, *Oracle*, *Sybase*, *FoxPro*, *MS Project* e *MS Excel*. *MetriFlame* permite a análise de dados, suporta a representação de dados e visualização de resultados, a identificação de tendências e comparação entre resultados anteriores e atuais.

DSS: *Decision Support System* consiste de um *data warehouse*, um motor OLAP (*On-Line Analytical Processing*) e um *front-end web*. A coleta de dados é feita de três fontes de dados de sistemas legados (suporte ao cliente, situações críticas e satisfação do cliente) que são integradas no *data warehouse*. Para apoiar a análise de dados, o motor analítico analisa os dados do *data warehouse* e os resultados são apresentados pelo *front-end web*.

SM in a CI Environment: Diversas ferramentas são usadas para apoiar a coleta de dados: *PartCover* é usada para fornecer dados de cobertura de testes; *NDepend* fornece

medidas de software; *Cruise Control.Net* age como um motor de CI (*Continuous Integration*) conectando as demais ferramentas; *Nant* automatiza as tarefas de *build* e fornece informações sobre as *builds*; *JUnit* executa os testes unitários e relata os resultados dos testes. Para apoiar a análise de dados, um *data warehouse* foi criado e um processo de ETL (*Extract, Transform and Load*) consolida os dados em um cubo do *data warehouse*, permitindo operações OLAP.

SOFAS: A arquitetura de *SOFAS* é construída de três principais componentes: *Software Analysis Web Services* (SA-WS), *Software Analysis Broker* (SA-B) e *Software Analysis Ontologies* (SA-Ontos). Tanto a coleta quanto a análise de dados são apoiadas por SA-WS, que coleta dados através de serviços de *CVS*, *Subversion*, *Git*, *Bugzilla*, *Google Code*, *Trac* e *SourceForge*. Para apoiar a análise de dados, SA-B fornece um compositor de serviços que permite que os dados trafeguem entre os serviços, gerando resultados combinados das análises.

Dione: Para apoiar a coleta de dados, *Dione* usa uma tecnologia de *smart clients* para conectar sistemas de controle de versão (*CVS*, *Subversion*, *Git*, *Mercurial*, *Clearcase*) e repositórios de defeitos (*Bugzilla*, *Jira*). Os *smart clients* extraem automaticamente dados de medidas de produtos e processos. A análise de dados é apoiada por um módulo de relatórios web, que permite a customização de relatórios para diferentes partes interessadas.

QualitySpy: *QualitySpy* tem dois grupos principais de funcionalidades. O primeiro grupo trata a aquisição de dados e apoia a coleta de dados. Esse grupo se baseia em conectores para coletar dados de classes Java, *Subversion*, *Jira* e *Hudson*. Medidas das classes Java são calculadas por um conector que envelopa a ferramenta *CKJM extended*. Medidas do *Jira* são coletadas através da interface do usuário usando Selenium. A análise de dados é apoiada pelo segundo grupo de funcionalidades (relatórios), que é operado através de um navegador web e fornece relatórios que podem ser predefinidos ou customizados pelo usuário.

3C: Para apoiar a coleta de dados, *Cruise Control* é usado para extrair dados de várias ferramentas: *Subversion*, para dados relacionados a código fonte; *Findbugs* para defeitos de programação; *Checkstyle*, para violações de padrões de codificação; *PMD*, como uma versão híbrida das ferramentas mencionadas anteriormente; *Cobertura*, para cobertura de testes; *JUnit*, para testes unitários; e *Cruise Control* para dados relacionados a *builds*. Uma ferramenta chamada *Cockpit* foi desenvolvida para apoiar a análise de dados. Os resultados de medição são colocados em gráficos que exibem as mudanças nas medidas ao decorrer do tempo.

ASSIST: *ASSIST* é composta de vários módulos. As atividades de medição são apoiadas principalmente pelo módulo *Measurement*, que engloba três submódulos: *Metrics*, *Services* e *Projects*. O planejamento da medição é apoiado pelo submódulo *Metrics*, que fornece funcionalidades para definir objetivos estratégicos, questões, indicadores e medidas, permitindo estabelecer relacionamentos entre eles. O planejamento da medição também é apoiado pelo submódulo *Projects*, que é usado para criar os planos de medição dos projetos. A coleta de dados, por sua vez, é apoiada pelo submódulo *Services*, que fornece serviços para a coleta de dados manual e automática para popular o banco de dados de medição. A funcionalidade de coleta de dados automática extrai dados de ferramentas de gerência de projetos, rastreamento de defeitos e de um ERP comercial. A análise de dados é apoiada pelo submódulo *Metrics*, que permite a criação/customização de relatórios para os indicadores selecionados e a construção de tabelas pivô que suplementam os indicadores. Os resultados das análises (representações gráficas/tabulares e interpretações ou comentários feitos pelo usuário) são armazenadas. Embora o módulo principal de apoio ao processo de medição seja *Measurement*, o módulo *Project Management* também auxilia atividades de medição. Ele fornece uma infraestrutura para a coleta e armazenamento de dados de projetos e é usado para a estimativa de projetos.

DePress: *DePress* utiliza um *framework* de código aberto chamado *KNIME* para apoiar a coleta e análise de dados. A coleta de dados é apoiada por um conjunto de extensões de plugins do *KNIME* que retornam dados de sistemas de gerenciamento de configuração de software (*Subversion* e *Git*), ferramentas de rastreamento de defeitos (*Jira* e *Bugzilla*) e leitores de medidas (*Judy*, *JaCoCo*, *EclipseMetrics*, *CheckStyle* e *PMD*). Dados também podem ser entrados manualmente. *DePress* suporta a análise de dados com a ajuda do *KNIME Report Designer*, que é baseado em uma ferramenta *BIRT* (*Business Intelligence Reporting Tool*).

QP4 (Integração) - Como a integração das ferramentas é feita e como ela é categorizada com relação às dimensões de escopo, ponto de vista, camada e nível?

A Tabela 3.4 sumariza a abordagem de integração adotada em cada proposta. Em seguida, as iniciativas são classificadas de acordo com o *framework* proposto por Izza (2009). É importante ressaltar que algumas publicações descrevem a abordagem de integração em mais detalhes do que outras. Dessa forma, a informação relacionada às abordagens de integração é heterogênea e limitada ao conteúdo das publicações.

Tabela 3.4 - Visão geral da integração de ferramentas nas abordagens.

Proposta	Descrição
TAME	A integração em TAME ocorre no componente <i>Measurement Tools</i> . Esse componente é integrado com três ferramentas de medição que coletam dados de código fonte Ada e os disponibilizam para TAME em um banco de dados relacional.
Tool Support for SM	A abordagem de integração é baseada em adotar regras externas para conteúdos e formatos de dados (nos quais todas as partes envolvidas precisam estar de acordo), usando ferramentas comuns para múltiplos propósitos e programas utilitários para converter dados para a interoperabilidade de ferramentas. Existe uma ferramenta central de análise (S-PLUS) e as outras ferramentas integradas (internas e comerciais) apoiam a captura, análise e representação dos dados. Dados de defeitos são armazenados em banco de dados, dados de testes em relatórios, e dados relacionados a medidas de código fonte são calculados dinamicamente. Programas utilitários são usados para extrair dados brutos de fontes de dados e convertê-los em um formato adequado para a análise.
GQM Tool	Nessa iniciativa, a integração entre a ferramenta principal (GQM Tool) e o sistema de gerência da configuração (PCMS) é feita através de ligações criadas entre os bancos de dados das ferramentas. Uma vez que as ligações sejam estabelecidas, os dados criados e armazenados no banco de dados da PCMS são disponibilizados automaticamente para o banco de dados da GQM Tool. Para integrar as ferramentas leitoras de medidas (Kaakatau e Resource Standard Metrics) com o banco de dados da GQM Tool, outra estratégia é usada pois os leitores de medidas gravam os dados de medição em arquivos HTML. Um DTD (<i>Document Type Definition</i>) foi definido e um tradutor para transformar dados em formato nativo em XML foi desenvolvido para cada ferramenta leitora de medidas. GQM Tool estava equipada com um interpretador capaz de ler os arquivos XML e armazena os dados em seu banco de dados.
MetriFlame	MetriFlame integra dados de várias fontes de dados, tais como sistemas de controle de versão e ferramenta de gerência de projetos, com a ajuda de um agendador. O agendador fornece um conjunto de tarefas (ex.: obtenção de dados brutos de fontes de dados externas, cálculo de medidas) que podem ser escolhidas pelo usuário e agendadas de acordo com as condições estabelecidas. Quando as condições são cumpridas, o agendador ativa MetriFlame para efetuar as tarefas agendadas.
DSS	Em DSS, os dados armazenados em três fontes de dados legado são integrados em um <i>data warehouse</i> central que é usado posteriormente por um <i>front-end</i> web para analisar e exibir informações de qualidade sobre serviços e satisfação do cliente. Detalhes sobre como os dados são carregados no <i>data warehouse</i> não são discutidos na publicação.
SM in a CI Environment	Nesta proposta, a integração é feita com a assistência do motor de CI Cruise Control.Net. Ele extrai dados das outras ferramentas e consolida-os em arquivos XML. Um programa desenvolvido extrai as medidas dos arquivos XML e armazena-os em um banco de dados relacional. Um processo de ETL consolida os dados do banco de dados relacional em um cubo do <i>data warehouse</i> , permitindo análises OLAP em cima do cubo.
SOFAS	Em SOFAS existe um conjunto de serviços web para análise de software que fornecem diferentes tipos de análises dos dados armazenados em diversas ferramentas. A abordagem de integração é baseada em fluxos de trabalho de serviços. Os usuários podem selecionar os serviços para compor os fluxos de trabalho, e SOFAS transforma os fluxos de trabalho em processos executáveis e os executa. A composição dos serviços permite os dados a trafegarem entre os serviços e gerar resultados combinados de análise.
Dione	Dione integra os dados usando <i>smart clients</i> que são pequenos programas Java capazes de serem executados diretamente da interface web de Dione. Eles coletam medidas dos artefatos de software (ex.: repositórios de código fonte e sistemas de gerência de defeitos) e envia-os para o servidor. A integração com aplicações de qualidade de software também é possível através de serviços web.
QualitySpy	QualitySpy usa quatro conectores para coletar dados de diferentes fontes. Três deles são usados para coletar dados brutos (dados armazenados de forma textual sem transformações) e um para calcular medidas de software de classes Java. Os dados coletados são armazenados em um repositório central e são disponibilizados para investigação posterior. O usuário pode definir medidas em cima dos dados brutos usando uma interface do módulo de relatórios implementada como cliente web que comunica com o servidor usando a arquitetura <i>Representational State Transfer</i> (REST).

Tabela 3.4 - Visão geral da integração de ferramentas nas abordagens (cont.).

Proposta	Descrição
3C	Nesta proposta, o motor de CI Cruise Control é responsável pela integração. Ele ativa as ferramentas de medição para coletar dados de teste (de JUnit e Cobertura) e dados de código fonte (de Findbugs, Checkstyle e PMD). Então, os resultados da medição são colocados em gráficos (Cockpit).
ASSIST	ASSIST é integrada com uma ferramenta comercial de gerência de projetos, uma ferramenta de rastreamento de defeitos e com um ERP, os quais dependem de um banco de dados relacional. Uma estratégia baseada em SQL (<i>Structured Query Language</i>) é usada para evitar desenvolvimento/modificações de código do lado das ferramentas comerciais. Uma sintaxe baseada em SQL e um interpretador foram desenvolvidos de forma que expressões complexas e parametrizadas podem ser escritas e os usuários podem definir construtos de medição, pesquisas, relatórios e serviços para coleta de dados através da interface de usuário.
DePress	DePress integra-se com outras ferramentas através da estrutura de plugins de KNIME. Para cada nova ferramenta a ser integrada, um novo plugin deve ser implementado. Para coletar dados, plugins podem trabalhar de duas formas: <i>online</i> e <i>offline</i> . O modo <i>online</i> significa acesso direto aos dados através da API fornecida pelas ferramentas. O modo <i>offline</i> significa que os dados são exportados pelas ferramentas e importados em DePress. DePress usa um formato tabular para trocar dados. O plugin verifica apenas se os dados recebidos contém as colunas requeridas ou não.

Escopo: A Tabela 3.5 apresenta a classificação dos estudos considerando a dimensão de escopo. A coluna *Intraorganizacional* refere-se a propostas que apresentam um escopo intraorganizacional, onde apenas uma organização está envolvida na abordagem de integração. A coluna *Interorganizacional* refere-se a propostas que envolvem mais do que uma organização na abordagem de integração, apresentando um escopo interorganizacional. A coluna *Indefinido* refere-se a propostas nas quais não foi possível identificar se uma ou mais organizações estavam envolvidas na iniciativa de integração. 8 propostas foram classificadas como escopo intraorganizacional (*TAME, Tool Support for SM, GQM Tool, MetriFlame, DSS, SM in a CI Environment, 3C, ASSIST*) e 4 propostas foram classificadas como indefinido (*SOFAS, Dione, QualitySpy, DePress*).

Tabela 3.5 - Classificação da integração de acordo com a dimensão de escopo.

Proposta	Intraorganizacional	Interorganizacional	Indefinido
TAME	X		
Tool Support for SM	X		
GQM Tool	X		
MetriFlame	X		
DSS	X		
SM in CI Environment	X		
SOFAS			X
Dione			X
QualitySpy			X
3C	X		
ASSIST	X		
DePress			X

Ponto de Vista: Os três pontos de vista desta dimensão são cobertos por todas as propostas. O ponto de vista do programador é coberto por todas as propostas, uma vez que esse ponto de vista refere-se à implementação dos sistemas e todas as iniciativas são funcionais. O ponto de vista do projetista é tratado em todas as iniciativas visto que esse ponto de vista preocupa-se com a representação de modelos da integração e todas as iniciativas apresentam ao menos um modelo descrevendo a integração. Em geral, os modelos considerados são modelos arquiteturais ou conceituais. Os detalhes sobre os modelos apresentados em cada estudo estão na Tabela 3.6. O ponto de vista do usuário refere-se à integração considerando a perspectiva do usuário. Todas as propostas fornecem funcionalidades da integração para os usuários.

Tabela 3.6 - Classificação da integração de acordo com o ponto de vista do projetista.

Proposta	Tipo de Modelo	Modelo apresentado na publicação
TAME	Arquitetural	Projeto arquitetural do sistema TAME.
Tool Support for SM	Conceitual	Um modelo abstrato representa o fluxo de informações e as conexões entre as ferramentas integradas.
GQM Tool	Conceitual	Esquemas de banco de dados e um modelo considerando a integração de GQM Tool com o sistema de gerência de configuração e ferramentas de medidas.
MetriFlame	Conceitual	Modelos representam o ambiente de integração e o agendador.
DSS	Arquitetural	Modelos arquiteturais e de projeto do sistema integrado.
SM in CI Environment	Conceitual	O esquema do <i>data warehouse</i> representa o cenário da integração.
SOFAS	Arquitetural	Modelo arquitetural de SOFAS.
Dione	Arquitetural	Modelo arquitetural de Dione.
QualitySpy	Arquitetural	Um modelo representa a arquitetura de alto nível de QualitySpy.
3C	Conceitual	Um modelo considera o fluxo de mensagens entre as ferramentas.
ASSIST	Conceitual	Um modelo representa o módulo responsável pela coleta de dados.
DePress	Conceitual	Modelos ilustram a integração de DePress com outras ferramentas e um fluxo de trabalho dos plugins.

Camada: A classificação da integração relacionada à dimensão de camada é apresentada na Tabela 3.7. A camada de processo não é considerada por nenhuma proposta, 7 propostas tratam apenas a camada de dados (*GQM Tool*, *MetriFlame*, *DSS*, *SM in a CI Environment*, *QualitySpy*, *ASSIST*, *DePress*) e 4 propostas tratam apenas a camada de serviços (*TAME*, *SOFAS*, *Dione*, *3C*). *Tool Support for SM* trata tanto as camadas de dados e serviços. Nessa proposta, os dados são acessados diretamente de sua fonte (camada de dados) e mensagens guiam o fluxo de informações (camada de serviços) da captura até as ferramentas de análise e apresentação.

Tabela 3.7 - Classificação da integração de acordo com a dimensão de camadas.

Proposta	Dados	Serviço	Processo
TAME		X	
Tool Support for SM	X	X	
GQM Tool	X		
MetriFlame	X		
DSS	X		
SM in CI Environment	X		
SOFAS		X	
Dione		X	
QualitySpy	X		
3C		X	
ASSIST	X		
DePress	X		

Nível: Com exceção de *SOFAS*, o nível mais elevado considerado por todas as propostas é o nível sintático. Esse nível engloba o modo como os modelos de dados e as assinaturas das operações são escritos (IZZA, 2009). Propostas classificadas nesse nível preocupam-se essencialmente em capturar dados independentemente da semântica dos mesmos ou dos serviços envolvidos. *SOFAS* é a única proposta considerando o nível semântico. *SOFAS* usa ontologias OWL para atribuir uma semântica clara aos dados consumidos e produzidos pelos serviços.

3.2.3. Discussões

Nesta seção são apresentadas algumas discussões em relação aos resultados obtidos.

Com relação às atividades de medição, o planejamento da medição é apoiado por quatro estudos, enquanto a coleta e a análise de dados são tratadas em todos eles. Uma explicação possível é que o planejamento da medição é altamente dependente do julgamento humano e não é propenso à automatização (KOMI-SIRVIO; PARVIAINEN; RONKAINEN, 2001). É possível notar que todas as propostas que apoiam a atividade de planejamento da medição (*TAME*, *GQM Tool*, *MetriFlame*, *ASSIST*) são baseadas no paradigma GQM (BASILI; CALDIERA; ROMBACH, 1994). Como GQM vem sendo adotado com sucesso pelas iniciativas de medição de software há anos, a sua utilização pelas propostas que apoiam o planejamento da medição era esperada.

Sobre a coleta de dados, as propostas apoiam essa atividade de diferentes formas. Existem tecnologias de *smart clients* (*Dione*), acesso direto ao banco de dados (*GQM Tool*, *ASSIST*), motores de CI (*SM in a CI Environment*, *3C*), chamadas de serviços web (*SOFAS*), plugins (*DePress*) e agendadores (*MetriFlame*). Todas as propostas são focadas na coleta

automática de dados. As propostas *TAME*, *Dione* e *ASSIST* também permitem a entrada manual de dados. Os autores da proposta *MetriFlame* ressaltam que o processo de medição deve ser automatizado sempre que possível, tornando a definição, coleta, cálculo e análise de dados mais fácil e com o menor esforço possível. Desse modo, a automatização da medição aumenta a visibilidade dos resultados e contribui para uma maior consciência sobre o raciocínio por trás da coleta de dados e de seu uso pelas organizações (KOMISIRVIO; PARVIAINEN; RONKAINEN, 2001). Entretanto, automatizar o processo de medição não significa suprimir a coleta manual de dados. Dessa forma, uma proposta híbrida (isto é, automática e manual) é preferida, pois facilita a migração de um processo essencialmente baseado na coleta manual de dados para um processo de medição automatizado. Além disso, permite a coleta de dados de medição que ainda não estão disponíveis nas ferramentas ou não são propensos para a coleta automatizada.

Em relação à análise de dados, alguns estudos focam em uma perspectiva específica como a análise da confiabilidade de software (*Tool Support for SM*), satisfação do cliente (*DSS*) e predição de defeitos (*Dione*, *DePress*). Entretanto, a maioria dos estudos (*TAME*, *GQM Tool*, *MetriFlame*, *SM in CI Environment*, *Dione*, *3C*) adota uma perspectiva mais geral, permitindo analisar se os objetivos estabelecidos foram alcançados. Pode ser destacado que todas as propostas que apoiam o planejamento de medição consideram essa perspectiva geral, uma vez que elas adotam o GQM, que é um paradigma orientado a objetivos.

Analisando-se as ferramentas integradas, existem propostas integrando ferramentas comerciais (ex.: *Tool Support for SM*, *MetriFlame*, *ASSIST*), ferramentas de código aberto (ex.: *SM in a CI Environment*, *SOFAS*, *QualitySpy*, *3C*, *DePress*) e ferramentas desenvolvidas internamente (ex.: *TAME*, *GQM Tool*, *ASSIST*). Algumas propostas focam em integrar ferramentas visando prover um ambiente mais completo (com novas funcionalidades de apoio à medição que não estavam presentes em nenhuma das ferramentas) no qual ferramentas externas são usadas essencialmente para a coleta de dados (ex.: *TAME*, *ASSIST*). Outras (ex.: *Tool Support for SM*, *SM in a CI Environment*, *SOFAS*, *QualitySpy*, *3C*) integram ferramentas visando melhorar funcionalidades de apoio à medição presentes em algumas delas (por exemplo, com a inclusão de dados fornecidos a partir das ferramentas integradas, funcionalidades de análise já existentes em uma ferramenta tornam-se mais abrangentes). 8 das 12 propostas (75%) adotam essa última abordagem. Também é possível notar que em algumas iniciativas (*Tool Support for SM*, *GQM Tool*, *MetriFlame*, *SM in a CI Environment*, *SOFAS*, *ASSIST*, *DePress*) apoiar o processo de medição foi a motivação principal para integrar ferramentas, enquanto em outras (*TAME*, *DSS*, *Dione*, *QualitySpy*,

3C) o apoio à medição foi alcançado como uma consequência da integração das ferramentas. Por exemplo, em *QualitySpy*, as ferramentas são integradas para apoiar o monitoramento do processo de desenvolvimento de software e, como consequência da integração, a medição de software também é apoiada.

Existe uma variedade de ferramentas que, apesar de não serem específicas para o domínio da medição de software, podem ser usadas para apoiar a medição. Isso aumenta a relevância da integração nesse domínio, pois as organizações podem escolher as ferramentas que são mais adequadas para as suas necessidades e trabalhar na integração delas. Embora exista uma diversidade de ferramentas sendo integradas, destaca-se a predominância de ferramentas relacionadas a código. Diversas ferramentas de medição de código, rastreamento de defeitos e sistemas de gerência de configuração são integradas nas propostas analisadas. Isso pode ser uma consequência do fato de que esses tipos de ferramentas são propícios para a coleta automática de medidas. Não obstante, algumas delas dependem de outras para fornecer informação. Por exemplo, como o código fonte normalmente é armazenado em um sistema de gerência de configuração, a presença de uma ferramenta de medição de código normalmente implica na presença de um sistema de gerência de configuração.

Considerando-se que ferramentas relacionadas a código foram integradas na maioria das propostas, era esperado que medidas relacionadas a código (ex.: complexidade ciclomática, número de métodos) fossem consideradas pela maioria das propostas. 10 dos 12 estudos tratam medidas de código. Levando em consideração as medidas e os tipos das ferramentas integradas, com exceção de *MetriFlame* e *ASSIST*, que possuem um escopo mais abrangente, as iniciativas de integração normalmente tratam de um escopo de medição específico (ex.: codificação, suporte ao cliente).

Como consequência da predominância de ferramentas relacionadas a código, a Codificação foi o principal processo medido, embora outros processos como testes, gerência de configuração e gerência de projetos também tenham sido medidos.

Em relação à integração, considerando as dimensões de classificação propostas por Izza (2009), alguns pontos podem ser destacados. Primeiro, a grande maioria dos estudos possui um escopo intraorganizacional. De fato, nenhum trabalho fez menção explícita a envolver um escopo interorganizacional. Segundo, com relação aos pontos de vista da integração (programador, projetista e usuário), todos eles foram considerados por todas as propostas, mas de maneiras diferentes. Para o ponto de vista do programador, cada uma das iniciativas é funcional e foi implementada de maneira *ad-hoc*, como descrito na QP3. O

ponto de vista do projetista é coberto, em geral, pela apresentação de modelos arquiteturais ou conceituais. O ponto de vista do usuário é considerado através das funcionalidades disponíveis para os usuários devido à integração. Terceiro, todas as propostas tratam até o nível sintático, exceto uma (*SOFAS*) que também trata o nível semântico. Por fim, em relação às camadas tratadas pela integração, os resultados mostraram alguma diversidade. 7 propostas tratam a camada de dados, 4 a camada de serviços e 1 ambas as camadas. A camada de processo não é tratada em nenhuma das propostas. Acredita-se que isso seja devido ao fato de que a integração na camada de processo (também conhecida como *Business Process Integration*) constitui a mais complexa abordagem de integração (IZZA, 2009). Ela vê a organização como um conjunto de processos de negócio e não meras ilhas de informação. A integração de processo lida com o fluxo de mensagens, regras e execução do processo.

A camada de serviços é considerada, mas apenas em poucas propostas. A integração na camada de serviços requer a comunicação por meio de troca de mensagens entre as ferramentas. Ferramentas que fornecem uma API (*Application Programming Interface*) de serviços encorajam a integração de serviços. Entretanto, se as ferramentas integradas não são capazes de se comunicar através de mensagens, a integração nessa camada demanda esforço extra, especialmente se as ferramentas não foram desenvolvidas pelo mesmo grupo que está efetuando a integração (que é o caso da maioria das propostas). Uma alternativa é o desenvolvimento de envelopadores (*wrappers*) para expor as funcionalidades das ferramentas como serviços, permitindo a integração nessa camada.

3.3 Considerações Finais do Capítulo

Este capítulo apresentou uma revisão sistemática da literatura (RSL) na qual foram investigadas as iniciativas sobre a integração de ferramentas para apoiar a medição de software. Nessa revisão, foram investigados aspectos relacionados à medição (medidas consideradas e atividades de medição apoiadas), ferramentas (ferramentas envolvidas e apoio fornecido) e dimensões de integração (de acordo com o *framework* de Izza (2009)).

Em resumo, todas as propostas analisadas apoiam a execução da medição (coleta e análise de dados), mas a maioria não apoia o planejamento da medição. A maioria das propostas trata medidas relacionadas a código e foca no processo de codificação. A cobertura do escopo, pontos de vista e nível de integração é similar entre as propostas (a maioria delas é intraorganizacional, cobre todos os pontos de vista e considera apenas aspectos sintáticos). Com relação às camadas, a integração de dados é a mais comum,

embora algumas propostas lidem com a integração na camada de serviços. Por fim, apenas uma proposta considera aspectos semânticos.

Os resultados dessa RSL apontam alguns aspectos importantes no contexto da integração de ferramentas para apoiar a medição de software: (i) existe um apoio limitado ao planejamento da medição; (ii) o escopo da medição não tem sido abrangente, limitando-se principalmente a medidas relacionadas a código e ao processo de codificação; (iii) a semântica não tem sido uma preocupação; (iv) arquiteturas orientadas a serviços não têm sido exploradas, resultando em uma integração limitada na camada de serviços; (v) uma visão holística do processo de software não tem sido considerada, levando à ausência de integração na camada de processo. No próximo capítulo é apresentada a abordagem proposta neste trabalho, a qual busca tratar algumas dessas lacunas.

Capítulo 4

Uma Abordagem Baseada em Ontologias para Integração de Ferramentas de Apoio à Medição de Software

Este capítulo descreve a abordagem OBA-MSI (Ontology-Based Approach for Measurement Systems Integration). A Seção 4.1 descreve a visão geral de OBA-MSI, a Seção 4.2 apresenta o processo de OBA-MSI e suas atividades. A Seção 4.4 apresenta as considerações finais do capítulo.

4.1 Visão Geral da Abordagem Proposta

Conforme já discutido neste trabalho, organizações enfrentam dificuldades para estabelecer e implementar um processo de medição eficiente. Dentre elas destaca-se a carência de ferramentas capazes de apoiar o processo de medição como um todo (FONSECA; BARCELLOS; FALBO, 2015). Uma alternativa para lidar com essa limitação é realizar a integração de ferramentas, no entanto, integrar ferramentas não é uma tarefa trivial (IZZA, 2009).

Em (CALHAU, 2011) é proposta OBA-SI (*Ontology-Based Approach for Semantic Integration*) que, conforme apresentado no Capítulo 2, considera que o processo de integração pode ser pensado como um processo de desenvolvimento de software e, dessa forma, é composto pelas fases: levantamento de requisitos, análise, projeto, implementação, testes e implantação. O foco de OBA-SI é a análise da integração. A fase de levantamento de requisitos, que antecede a fase de análise, é trata apenas superficialmente. Segundo OBA-SI, nessa fase deve-se definir o cenário de integração identificando-se os objetivos da integração, as atividades do processo de negócio a serem apoiadas e os sistemas a serem integrados. No entanto, OBA-SI não fornece orientações sobre como essas definições devem ser realizadas.

No âmbito da integração de ferramentas para apoiar medição de software a fase de levantamento de requisitos é particularmente importante, pois a integração das ferramentas deve ser realizada de modo a atender os objetivos e as necessidades de informação de medição da organização. Assim, para auxiliar na integração de ferramentas para apoiar o processo de medição de software, neste trabalho propõe-se uma especialização de OBA-SI na qual a fase de levantamento de requisitos é detalhada, sendo conduzida utilizando-se uma abordagem baseada em objetivos. A especialização de OBA-SI proposta neste

trabalho é chamada OBA-MSI (*Ontology-Based Approach for Measurement Systems Integration*) e visa auxiliar organizações a estabelecer um processo de medição adequado, bem como apoio ferramental para ele, através do uso de ontologias de medição de software e da integração de ferramentas.

OBA-MSI advoga pelo uso da Ontologia de Referência de Medição de Software (ORMS) e da Ontologia de Tarefa de Medição de Software (OTMS) como base para a definição do processo de medição e condução da integração das ferramentas. Essas ontologias proveem conhecimento acerca de medição de software, considerando aspectos estruturais (conceitos e relações) e comportamentais (atividades, produtos e insumos). Em linha com essas ontologias, OBA-MSI também se baseia nos princípios do GQM, orientando a medição de software alinhada aos objetivos da organização. Uma vez que essas ontologias foram desenvolvidas com base nos principais padrões de medição de software presentes na literatura, ao considerar o uso do conhecimento por elas provido, OBA-MSI incorpora as melhores práticas de medição. Na próxima seção, o processo para aplicação de OBA-MSI é apresentado e suas atividades são detalhadas.

4.2 Processo de OBA-MSI

OBA-MSI é uma especialização de OBA-SI, portanto, é baseado nas fases de integração definidas por OBA-SI. OBA-MSI apresenta diferenças em relação a OBA-SI nas duas primeiras fases: *Levantar Requisitos da Integração* e *Analisar Integração*. Na Figura 4.1 é apresentada uma visão geral dessas fases do processo de OBA-MSI. As demais fases do processo de OBA-MSI (*Projetar Integração*, *Implementar Integração*, *Testar Integração* e *Implantar Conjunto Integrado*) não são apresentadas na figura, pois são idênticas às fases do processo de OBA-SI.

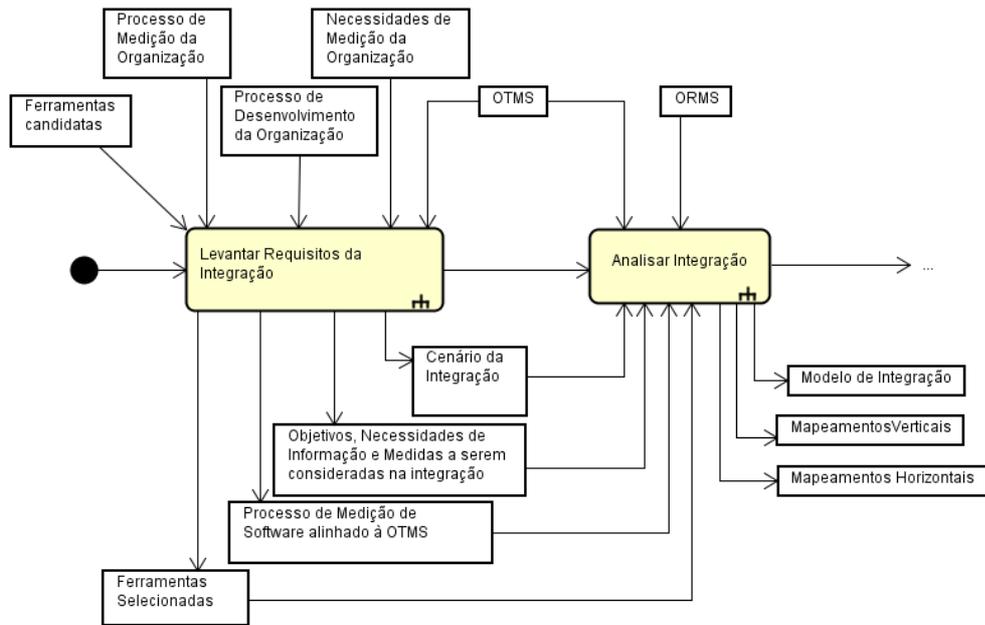


Figura 4.1 - Primeiras fases do processo de OBA-MSI.

4.2.1. Levantar Requisitos da Integração

Para realizar o levantamento de requisitos da integração, OBA-MSI propõe o uso de uma abordagem orientada a objetivos. Além disso, OBA-MSI considera que o processo de medição de software da organização deve estar adequadamente definido para ser apoiado pela integração. Dessa forma, o processo de medição da organização deve estar alinhado ao processo de medição de software estabelecido na OTMS. A fase de levantamento de requisitos de OBA-MSI é composta por quatro atividades: (i) Identificar e Alinhar o Processo de Medição; (ii) Identificar Objetivos, Necessidades de Informação e Medidas relevantes para a Organização; (iii) Identificar Ferramentas Fornecedoras de Dados e Serviços para a Medição de Software; e (iv) Registrar Cenário de Integração. A Figura 4.2 ilustra as atividades da fase *Levantar Requisitos da Integração*, que são descritas em seguida.

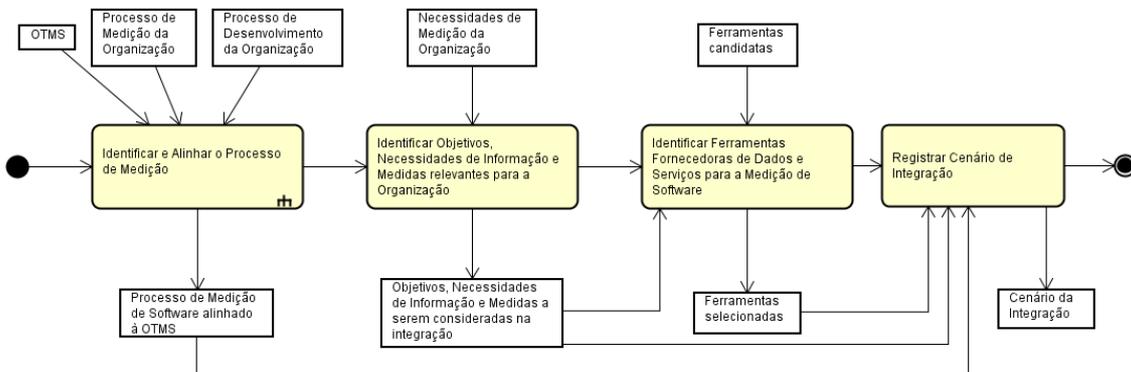


Figura 4.2 - Detalhamento de Levantar Requisitos da Integração.

4.2.1.1. Identificar e Alinhar o Processo de Medição

A primeira atividade de *Levantar Requisitos da Integração* trata do alinhamento do processo de medição da organização com o processo de medição estabelecido na OTMS. Esse alinhamento visa garantir que o processo de medição de software da organização inclua todas as atividades necessárias para que a medição seja realizada de maneira apropriada e que essas atividades estejam adequadamente definidas.

Em OBA-SI o alinhamento entre o processo de negócio a ser apoiado pela integração e a ontologia de tarefa sendo usada é feito na atividade *Analisar Integração* de OBA-SI. Contudo, seguindo-se OBA-SI, o processo de medição da organização não seria afetado pela iniciativa da integração. Ao trazer o alinhamento entre o processo de medição e a ontologia de tarefa para o início do processo de integração semântica, pretende-se não apenas definir um conjunto integrado de ferramentas para apoiar o processo de medição da organização, mas também melhorar esse processo, fazendo com que a iniciativa de integração (e o conjunto integrado de ferramentas resultante dela) seja também um agente de mudanças na organização, abrindo espaço para a melhoria do processo de medição da organização ou para a sua clara identificação.

A subatividade *Identificar e Alinhar o Processo de Medição* é composta pelas seguintes subatividades: (i) Verificar Existência de Processo de Medição de Software na Organização; (ii) Identificar Atividades de Medição no Processo de Desenvolvimento da Organização; e (iii) Alinhar Processo de Medição com OTMS. A Figura 4.3 ilustra as subatividades de *Identificar e Alinhar o Processo de Medição*.

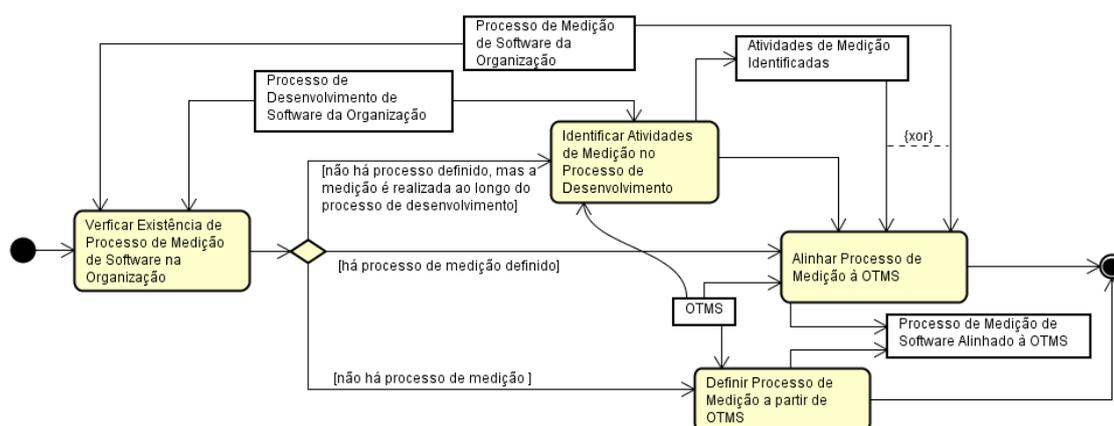


Figura 4.3 - Subatividades de Identificar e Alinhar o Processo de Medição.

Para realizar a identificação do processo de medição da organização, o primeiro passo é *Verificar a Existência de Processo de Medição da Organização*. Algumas organizações possuem um processo de medição definido, no entanto, há organizações que, embora

realizem algumas atividades de medição, não têm um processo explicitamente definido e as atividades de medição são realizadas implicitamente ao longo do processo de desenvolvimento. Há, ainda, organizações que não possuem um processo definido, pois não realizam medição de software e darão início a essa prática a partir da iniciativa de integração de ferramentas.

Caso a organização possua processo de medição definido, deve-se *Alinhar o Processo de Medição à OTMS*. Para isso, é necessário realizar um mapeamento entre as atividades do processo de medição da organização e as atividades do processo de medição definido na OTMS, o que pode ser feito com o auxílio de uma tabela no formato apresentado na Tabela 4.1. Após a realização dos mapeamentos é possível identificar se há alguma atividade faltando no processo de medição da organização ou se alguma atividade definida é desnecessária. Assim, com base no mapeamento realizado, o processo de medição da organização deve ser adequado (por exemplo, inserindo-se novas atividades) para ficar alinhado à OTMS.

Tabela 4.1 – Tabela para registro do mapeamento entre atividades do processo de medição da organização e da OTMS.

Atividades de Medição da Organização	Atividades de Medição de OTMS	Comentários
<<atividade de medição presente no processo de medição da organização>>	<<atividade de medição de OTMS (Planejar Medição, Executar Medições, Analisar Medições ou alguma de suas subatividades >>	<<informações sobre a equivalência (total ou parcial) entre as atividades mapeadas>>

Caso a organização não possua processo de medição definido e ainda não realize medição de software, deve-se *Definir o Processo de Medição a partir de OTMS*. Para isso, o processo estabelecido em OTMS deve ser usado para estabelecer as atividades e subatividades do processo de medição da organização, bem como os principais artefatos utilizados (insumos) e produzidos (produtos). Cabe ressaltar que podem ser realizados alguns ajustes para que o processo fique alinhado à organização (por exemplo, podem-se fazer ajustes em nomes de artefatos produzidos ou atividades, para que fiquem consistentes com os demais processos da organização).

Por fim, caso a organização não possua um processo de medição definido, mas realize medição ao longo de seu processo de desenvolvimento, devem-se identificar as atividades em que a medição ocorre. A coleta de dados é uma das atividades do processo de medição que, muitas vezes, é realizada de maneira implícita pelas organizações através da coleta de dados por ferramentas que apoiam outros processos de software. Por exemplo, uma ferramenta de apoio à gerência de projetos captura e registra dados

referentes à duração planejada e realizada para as atividades dos projetos, ou seja, há execução de medição: valores medidos são atribuídos às medidas duração estimada e duração real que medem a atividade (entidade mensurável) à qual estão relacionadas. Assim, mesmo que a organização não tenha definido explicitamente que esta atividade é uma atividade de medição, ela o é. De maneira similar, quando dados são analisados (por exemplo, quando um gerente compara valores de duração planejados e realizados para averiguar a aderência do projeto ao cronograma estabelecido), uma atividade de análise de medição é realizada, mesmo que ela não esteja explicitamente definida como tal.

Assim, durante a atividade *Identificar Atividades de Medição no Processo de Desenvolvimento*, deve-se analisar criteriosamente o processo de desenvolvimento da organização, buscando-se identificar entre suas atividades aquelas onde ocorre alguma atividade do processo de medição. Durante essa atividade, sugere-se que sejam utilizadas algumas técnicas, tais como entrevistas e reuniões, para que sejam obtidas as informações acerca do processo de desenvolvimento que permitam a identificação das atividades relacionadas à medição.

Para realizar esta atividade, idealmente, o processo de desenvolvimento da organização deve estar formalmente definido (textualmente ou em um diagrama, por exemplo). Caso a organização não tenha o processo definido, orienta-se que isso seja feito para que seja possível realizar a identificação das atividades de medição. Após a identificação das atividades onde ocorre medição, pode ser necessário rever a definição do processo de desenvolvimento, para que essas atividades fiquem mais explícitas. Por exemplo, se na atividade *Codificação* do processo de desenvolvimento de uma organização ocorre a coleta de dados para medidas referentes ao código fonte (tais como número de linhas de código, número de classes e número de métodos), ela poderia ser dividida em *Codificação* e *Coleta de Dados sobre Código-Fonte*. Dessa forma, a atividade *Coleta de Dados sobre Código-Fonte*, que é uma atividade relacionada à medição, ficaria explícita no processo de desenvolvimento.

Para a operacionalização desta atividade, sugere-se que o processo da organização seja definido graficamente e que as atividades onde ocorre medição sejam destacadas no diagrama. Por exemplo, a Figura 4.4 ilustra um processo de desenvolvimento de software fictício onde são destacadas as atividades relacionadas à medição. As atividades identificadas com a letra D representam atividades relacionadas apenas ao processo de desenvolvimento de software (ou a processos de apoio, como gerência de projetos). As atividades identificadas com a letra M são atividades dedicadas apenas à medição e as atividades com as letras D e M são atividades que relacionadas tanto ao desenvolvimento

de software quanto à medição. Para essas, conforme discutido no parágrafo anterior, sugere-se a decomposição para que as atividades relacionadas apenas à medição fiquem explícitas no processo de desenvolvimento. A Figura 4.5 ilustra o processo de desenvolvimento após a decomposição da atividade *Codificar e Testar Sistema*.

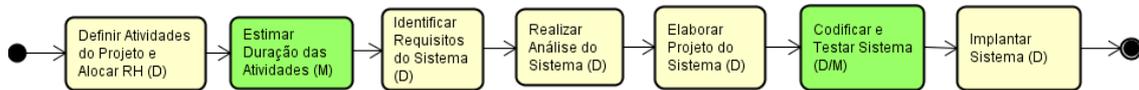


Figura 4.4 - Processo de Desenvolvimento de Software fictício com destaque para as atividades onde ocorre a medição de software.

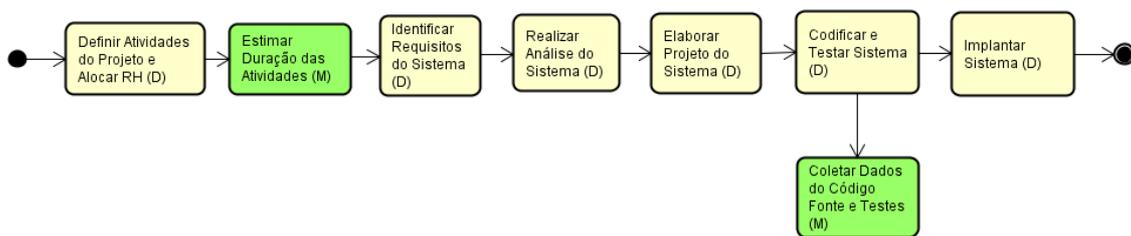


Figura 4.5 - Processo de Desenvolvimento de Software alterado para explicitar atividades relacionadas à medição.

Após a identificação das atividades, deve-se *Alinhar o Processo de Medição à OTMS*, como descrito anteriormente. Ou seja, as atividades de medição identificadas devem ser mapeadas para atividades do processo estabelecido em OTMS e o processo de medição deve ser adequado com base nos resultados do mapeamento. Para o exemplo apresentado na Figura 4.5, as atividades *Estimar Duração das Atividades* e *Coletar Dados do Código Fonte* são equivalentes à atividade *Executar Medições* de OTMS, porém considerando a medição de algumas medidas específicas. Não há no processo de desenvolvimento atividades equivalentes a *Planejar Medição* e *Analisar Medições* de OTMS. Dessa forma, é necessário incluir essas atividades no processo de medição da organização que está sendo explicitado.

Para a definição do processo de medição alinhado à OTMS, pode ser utilizado um diagrama de atividades da UML, onde cada partição representa uma atividade definida no processo de medição da OTMS. A Figura 4.6 ilustra o processo de medição definido a partir das atividades de medição identificadas (Figura 4.5) e OTMS. As atividades em destaque são as atividades identificadas a partir do processo de desenvolvimento. As demais são atividades definidas na OTMS.

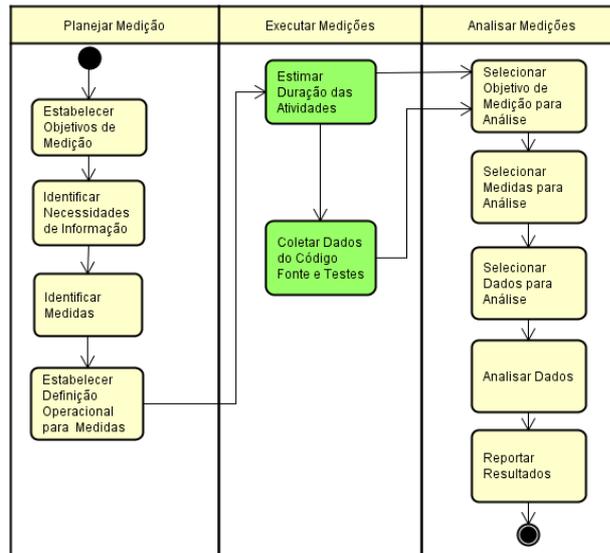


Figura 4.6 - Exemplo de processo de medição alinhado à OTMS .

4.2.1.2. Identificar Objetivos, Necessidades de Informação e Medidas relevantes para a Organização

Esta atividade de OBA-MSI defende o princípio de que a medição de software deve ser baseada em objetivos, para que as informações providas sejam realmente úteis. Realizar medição não orientada a objetivos pode implicar no desperdício de recursos e desmotivação para a realização da medição, uma vez que, não se obtendo os benefícios desejados a partir da medição, os esforços empregados não teriam retorno. Dessa forma, para realizar a integração de ferramentas de modo a agregar valor para a organização, os objetivos que se deseja monitorar devem ser previamente estabelecidos.

Para o estabelecimento dos objetivos de medição que deverão ser considerados na integração, deve-se, inicialmente, identificar os objetivos estratégicos da organização relevantes para a medição. A partir deles, os objetivos de medição podem ser derivados e darão origem às necessidades de informação a serem satisfeitas e às medidas necessárias.

A identificação dos objetivos, necessidades de informação e medidas deve ser realizada adotando-se o paradigma *Goal-Question-Metric* (GQM) (BASILI; CALDIERA; ROMBACH, 1994), com uma pequena adaptação, uma vez que em OBA-MSI devem ser identificados tanto objetivos estratégicos quanto objetivos de medição. A Figura 4.7 ilustra um exemplo de uso do GQM adaptado para a identificação de objetivos, necessidades de informação (questões) e medidas em OBA-MSI.

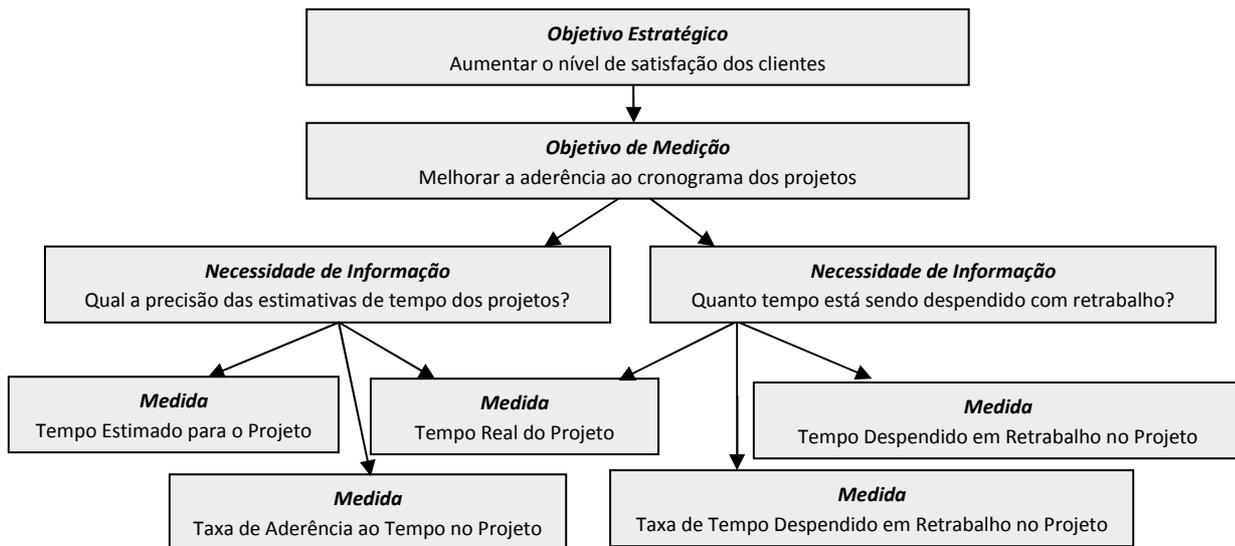


Figura 4.7 - Aplicação de GQM para a identificação de Objetivos, Necessidades de Informação e Medidas em OBA-MSI.

Uma vez que em OBA-MSI a identificação dos objetivos, necessidades de informação e medidas é realizada visando à integração de ferramentas, os objetivos estabelecidos devem ser passíveis de ser monitorados por medidas que possam ser obtidas a partir de ferramentas a serem integradas. Dessa forma, esta atividade de OBA-MSI deve ser realizada de maneira iterativa com a atividade *Identificar Ferramentas Fornecedoras de Dados e Serviços para a Medição de Software*. Assim, consegue-se garantir que os objetivos, necessidades de informação e medidas definidos estejam em consonância com as ferramentas que serão integradas.

4.2.1.3. Identificar Ferramentas Fornecedoras de Dados e Serviços para a Medição de Software

Nesta atividade são identificadas as ferramentas a serem integradas. Conforme mencionado, esta atividade deve ser conduzida iterativamente com a atividade anterior, uma vez que as ferramentas integradas devem ser capazes de atender às necessidades de informação fornecendo as medidas necessárias ao monitoramento dos objetivos.

Para selecionar as ferramentas a serem integradas, devem ser analisadas ferramentas candidatas à integração, ou seja, ferramentas que sejam capazes de apoiar o monitoramento dos objetivos estabelecidos fornecendo as medidas necessárias. Devem ser consideradas tanto ferramentas que a organização já utilize (desenvolvidas *in-house* ou adquiridas), bem como outras ferramentas não utilizadas até então pela organização. Dentre as ferramentas que a organização utiliza, devem ser consideradas aquelas que apoiam atividades

relacionadas ao processo de medição, mesmo que elas não sejam ferramentas dedicadas a este fim.

As ferramentas desenvolvidas na própria organização (*in-house*) usualmente têm a vantagem de facilitar a obtenção de informações detalhadas, uma vez que a equipe de desenvolvimento é da própria organização e a documentação (caso exista) está disponível. O fácil acesso aos desenvolvedores da ferramenta também contribui, caso seja necessário fazer adaptações na ferramenta para a integração. Além disso, essas ferramentas tendem a ser mais simples de integrar, uma vez que o código fonte está disponível e os dados, funcionalidades e processos que essas ferramentas apoiam são de conhecimento da organização.

Ferramentas não desenvolvidas pela organização também podem ser selecionadas para o fornecimento de dados e serviços de apoio à medição. No caso de ferramentas comerciais, usualmente há alguma documentação disponível e tem sido comum a disponibilização de APIs (*Application Programming Interface*) de serviços, que facilitam identificar os serviços que a ferramenta é capaz de fornecer. A desvantagem é que, geralmente, o código fonte dessas ferramentas não está disponível, o que dificulta o desenvolvimento de serviços para fornecer novas medidas.

Ferramentas disponíveis para uso via web também devem ser consideradas para a seleção. Algumas ferramentas fornecem serviços online através de abordagens como *Software as a Service* (SaaS) e *cloud computing* (computação em nuvem). Por serem utilizadas via Internet, essas ferramentas atendem a um público amplo e heterogêneo e normalmente fornecem uma documentação completa com a descrição dos serviços disponíveis em suas APIs. Existem catálogos de software online onde ferramentas desse tipo podem ser encontradas, tais como o catálogo da *Targetware*², o catálogo do *Software Público Brasileiro*³ e o catálogo da *Apache Software Foundation*⁴. Algumas desvantagens do uso desse tipo de ferramenta estão relacionadas ao armazenamento externo dos dados da organização e a principal desvantagem é que se o serviço da ferramenta estiver indisponível, a organização perde o acesso aos dados. Outra desvantagem está relacionada com a garantia da segurança da informação: dados confidenciais podem ser acessados por indivíduos não autorizados, caso não existam controles de segurança apropriados, visto que os dados também trafegam pela Internet e não apenas pela rede de computadores da organização.

² <http://www.software.com.br/>

³ <http://softwarepublico.gov.br/>

⁴ <http://www.apache.org/>

Embora não seja possível cobrir todas as possibilidades no contexto da seleção de ferramentas, OBA-MSI provê algumas diretrizes para auxiliar na seleção de ferramentas fornecedoras de dados e serviços para a medição, descritas a seguir.

- **Ferramentas de apoio específico à medição de software:** Ferramentas concebidas para apoiar a medição de software podem ser usadas como repositório centralizado de medição, uma vez que permitem o armazenamento de dados relacionados à medição de software sem se restringir ao processo de software no qual a medição é aplicada. Assim, sugere-se que seja selecionada uma ferramenta de apoio específico à medição de software para a integração.
- **Ferramentas de apoio aos demais processos de software na organização:** As ferramentas utilizadas para apoiar processos de software geralmente capturam dados a respeito da execução do processo que são úteis para a medição de software. Além disso, quando são ferramentas já utilizadas pela organização, esta já está acostumada às ferramentas e tem conhecimento sobre elas, incluindo os dados que elas proveem e que são úteis à tomada de decisão, facilitando a identificação dos objetivos passíveis de serem monitorados e medidas disponíveis. Assim, ferramentas de apoio aos processos que fornecem dados de medidas identificadas na atividade anterior devem ser selecionadas, dando-se preferência, inicialmente, às ferramentas em uso na organização.
- **Ferramentas com API de serviços:** Ferramentas que possuem API de serviços facilitam a integração, uma vez que a API representa uma interface para fornecimento e consumo de dados. Assim, preferem-se ferramentas que proveem APIs de serviços às que não o fazem.
- **Ferramentas com documentação:** O fato de haver documentação disponível auxilia a identificação de medidas que podem ser obtidas a partir da ferramenta, seja através de funcionalidades internas que podem ser envelopadas pela criação de serviços ou através de consulta direta ao banco de dados. Além disso, a documentação facilita obter conhecimento sobre os modelos estruturais e comportamentais da ferramenta, que são necessários na fase de análise da integração. Dessa forma, deve-se dar preferência a ferramentas com documentação disponível.

- **Ferramentas desenvolvidas *in-house*:** A maior vantagem de se usar uma ferramenta desenvolvida *in-house* é a facilidade para obter informações sobre ela e realizar alterações, caso seja necessário. Dessa forma, quando possível, deve-se dar preferência a essas ferramentas.

4.2.1.4. Registrar Cenário de Integração

Nesta atividade, o cenário de integração definido a partir dos resultados das atividades anteriores deve ser registrado. Para isso, assim como em OBA-SI, sugere-se o uso de uma tabela no formato da Tabela 4.2, onde devem ser explicitados os sistemas a serem integrados, o(s) domínio(s) da integração (domínios nos quais a medição será aplicada) e as atividades do processo de medição que serão apoiadas pela integração.

Tabela 4.2 – Tabela para registro do Cenário de Integração.

Sistemas	<<sistemas a serem integrados>>
Domínio	<<domínio(s) da integração>>
Atividades	<<atividades do processo de medição a serem apoiadas pela integração >>

4.2.2 Analisar Integração

Uma vez identificados o processo de medição a ser apoiado, os objetivos, necessidades de informação e medidas necessários e as ferramentas a ser integradas, deve-se realizar a análise da integração. Conforme descrito no Capítulo 2, para realizar a integração de ferramentas aplicando-se OBA-SI, deve ser selecionada ou desenvolvida a ontologia a ser utilizada para garantir o compartilhamento da semântica entre as aplicações integradas. Assim, em OBA-SI as ontologias são selecionadas de acordo com o domínio das aplicações sendo integradas. Em OBA-MSI, independente das ferramentas envolvidas na integração ou do domínio para o qual foram concebidas, as ontologias a serem utilizadas são a Ontologia de Referência para Medição de Software (ORMS) e a Ontologia de Tarefa de Medição de Software (OTMS), uma vez que o que se deseja apoiar é o processo de medição de software. Assim, ao invés de selecionar as ontologias a serem usadas na integração, em OBA-MSI devem ser selecionados os fragmentos de ORMS e OTMS relevantes à integração. Vale ressaltar que essas ontologias podem ser estendidas para tratar algum aspecto específico do escopo da integração.

As ontologias selecionadas durante a análise da integração servem para atribuir semântica aos elementos compartilhados na integração. Para isso, conforme sugerido em OBA-SI, devem ser feitos mapeamentos verticais, nos quais os conceitos e relações da

ontologia são mapeados para conceitos e relações das ferramentas integradas, resultando em um modelo integrador que é usado como base para projetar a solução de integração.

Durante os mapeamentos verticais é possível identificar os conceitos e relações da ontologia presentes em cada ferramenta. Alguns conceitos e relações da ontologia podem estar presentes em todas as ferramentas envolvidas na integração, outros podem estar presentes em apenas algumas delas. A partir dos mapeamentos verticais é possível identificar quais ferramentas serão responsáveis por tratar quais conceitos e relações na solução de integração.

Quando há equivalência semântica entre um conceito/relação da ontologia com elementos (classes, atributos, relações, etc.) de todas as ferramentas (i.e., o conceito/relação da ontologia é mapeado para elementos de todas as ferramentas envolvidas), elementos de todas as ferramentas deverão ser integrados para tratar o conceito/relação na solução de integração. Por exemplo, caso em um cenário de integração o conceito Projeto, presente na ontologia, seja tratado em todas as ferramentas integradas, será necessário integrar os elementos semanticamente equivalentes a Projeto em todas as ferramentas para que o conceito Projeto seja tratado adequadamente na solução de integração. Dessa forma, todas as ferramentas são capazes de lidar com esse conceito individualmente e todas serão responsáveis por tratá-lo na integração.

Por outro lado, quando um conceito/relação da ontologia possui equivalência semântica apenas com elementos de uma ferramenta (ou algumas), essa será a ferramenta responsável por lidar com o conceito na integração. Por exemplo, caso o conceito Equipe, presente na ontologia, seja tratado por apenas uma das ferramentas envolvidas em um cenário de integração, apenas essa ferramenta será a responsável por lidar com esse conceito na solução de integração e apenas ela é capaz de individualmente (fora da integração) tratar esse conceito. Assim, essa será a ferramenta que proverá dados ou serviços que permitirão integrar, por exemplo, equipes a projetos.

Uma vez realizados os mapeamentos horizontais, obtém-se o modelo integrador que, conforme orienta OBA-SI, é definido a partir do modelo conceitual da ontologia, acrescentando-se conceitos presentes nas ferramentas a serem integradas e que não foram mapeados com conceitos da ontologia, mas são relevantes para a integração. Cabe destacar que para atender requisitos da integração, novos conceitos necessários à integração e que não fazem parte nem da ontologia nem das ferramentas integradas podem ser inclusos no modelo integrador.

Nos mapeamentos verticais, o modelo integrador é mapeado para as ferramentas a fim de se identificar relações não explicitadas nos mapeamentos horizontais. O mapeamento de um conceito do modelo integrador para um elemento de uma ferramenta, de maneira análoga ao que ocorre nos mapeamentos horizontais, significa que aquela é a ferramenta que irá lidar com o conceito na integração. Quando um conceito do modelo integrador não possui mapeamento com elemento de nenhuma ferramenta, tem-se que aquele conceito deverá ser tratado pela solução de integração propriamente dita (por exemplo, por uma aplicação mediadora que fará a mediação da integração entre as ferramentas).

Os mapeamentos horizontais ou verticais podem ser feitos de maneira direta, onde, por exemplo, um conceito da ontologia é mapeado para uma classe de uma ferramenta ou uma relação da ontologia é mapeada para um relacionamento do modelo conceitual de uma ferramenta. No entanto, é importante observar que há situações onde o mapeamento não ocorre tão diretamente, sendo possível haver mapeamentos, por exemplo, entre conceitos da ontologia e atributos de classe, instâncias, instâncias em que um atributo possua determinado valor ou instâncias em que determinado relacionamento exista. Dessa forma, durante a realização dos mapeamentos, é preciso atentar-se às possíveis equivalências semânticas entre elementos da ontologia e elementos das ferramentas.

4.2.2.1. Integração nas Camadas de Serviço e Processo

Conforme citado anteriormente, a integração de ferramentas para apoiar o processo de medição envolve ferramentas não concebidas para apoiar o processo medição de software, mas sim outros processos de software, como a gerência de projetos, gerência de configuração, gerência de requisitos, codificação e testes. Usualmente, essas ferramentas não apresentam um modelo comportamental sequencial bem definido, parecido com um *workflow*. Como consequência, a abordagem para realizar a integração semântica nas camadas de serviço e de processo definida em OBA-SI precisa ser ajustada para lidar com a integração de ferramentas quando essas características estiverem presentes.

Segundo OBA-SI, para realizar a integração semântica nas camadas de serviço e processo, uma ontologia de tarefa deve ser usada para atribuir semântica às funcionalidades dos sistemas e às atividades do processo. Dessa forma, os elementos comportamentais dos sistemas (funcionalidades, produtos e insumos) devem ser analisados para que seja atribuída semântica a eles por meio de mapeamentos entre esses elementos e os elementos da ontologia de tarefa. Essa abordagem é adequada para sistemas que apresentam um

modelo comportamental sequencial bem definido, similar a um processo. Por exemplo, o modelo comportamental de um sistema para compra de passagens é sequencial, o que favorece a identificação da sequência em que as funcionalidades ocorrem e o mapeamento destas com atividades de uma ontologia de tarefa. No entanto, vários sistemas não têm um modelo comportamental sequencial bem definido, sendo, na verdade, um conjunto de funcionalidades que podem ser realizadas sem uma ordem preestabelecida, existindo apenas algumas relações de dependência entre algumas delas. Nesses casos, a abordagem sugerida em OBA-SI não é adequada para realizar a integração semântica nas camadas de serviço e processo.

Assim, para realizar a integração semântica nessas camadas, inicialmente devem ser identificados os serviços disponíveis nos sistemas que são necessários para apoiar o processo de medição de software. Em seguida, deve ser feito o mapeamento desses serviços e as atividades da OTMS. Porém, diferentemente do que sugere OBA-SI, nesse caso, o mapeamento entre um serviço e uma atividade não indica equivalência semântica entre eles, mas sim que o serviço *apoiar* a realização da atividade. Para realizar esse mapeamento pode ser utilizada uma tabela com o formato sugerido na Tabela 4.3.

Tabela 4.3 – Tabela para Mapeamento entre Serviços das Ferramentas Integradas e Atividades de OTMS.

Atividades OTMS	Serviços	Descrição	Ferramenta
Planejar Medição			
<<subatividade de Planejar Medição em OTMS>>	<<serviços/funcionalidades que apoiam a realização da atividade>>	<<descrição do apoio provido pelo serviço à execução da atividade>>	<<ferramenta que disponibiliza o serviço/funcionalidade>>
Executar Medições			
<<subatividade de Executar Medições em OTMS>>			
Analisar Medições			
<<subatividade de Analisar Medições em OTMS>>			

Em seguida, como sugere OBA-SI, para atribuir semântica ao processo de medição de software da organização, deve ser feito o mapeamento de suas atividades com as atividades estabelecidas na OTMS. Cabe notar que em OBA-MSI esse mapeamento já foi feito (pelo menos parcialmente) durante a atividade *Identificar e Alinhar Processo de Medição*, nas subatividades *Alinhar Processo de Medição à OTMS* ou *Definir Processo de Medição a partir de OTMS*.

Com o mapeamento entre os serviços providos e as atividades do processo de medição estabelecido na OTMS e deste com as atividades do processo de medição da organização, os serviços mapeados para as atividades da OTMS podem, por transitividade, ser mapeados para as atividades do processo de medição da organização, indicando que

eles apoiam essas atividades do processo. Assim, uma vez realizada a integração das ferramentas considerando-se os serviços identificados, obtém-se apoio integrado ao processo de medição definido para a organização.

4.3 Considerações Finais do Capítulo

Durante a investigação da literatura feita no contexto deste trabalho e descrita no Capítulo 3, foram identificadas algumas lacunas no âmbito da integração de ferramentas par apoiar a medição de software. Dentre elas, estão: (i) a falta de uma visão holística do processo de software; (ii) o apoio limitado ao planejamento da medição; (iii) a não preocupação com aspectos semânticos, e (iv) não cobertura da camada de processo. Buscando-se tratar essas lacunas, este capítulo apresentou uma abordagem especializada a partir de OBA-SI para apoiar iniciativas de integração de ferramentas no contexto de medição de software.

Em relação a (i), OBA-MSI considera uma visão holística do processo de medição de software, uma vez que utiliza a Ontologia de Tarefa de Medição de Software (OTMS) como um de seus pilares e requer que o processo de medição da organização a ser apoiado pela integração esteja alinhado à OTMS. Para tratar (ii), OBA-MSI tem uma atividade (*Identificar Objetivos, Necessidades de Informação e Medidas*) para lidar com aspectos do planejamento da medição, os quais serão apoiados pela integração das ferramentas. Em relação a (iii), OBA-MSI é uma especialização de OBA-SI para tratar a integração de ferramentas de apoio à medição de software. Logo, uma vez que OBA-SI trata a integração no nível semântico, esta lacuna também é tratada em OBA-MSI. De maneira análoga, em relação a (iv), considerando-se OBA-MSI uma especialização de OBA-SI, ela também permite a integração nas camadas de dados, serviços e processo. Para tratar particularidades de alguns cenários de integração, OBA-MSI sugere uma abordagem alternativa a OBA-SI para a integração semântica nas camadas de serviço e processo.

OBA-MSI é uma especialização de OBA-SI centrada no levantamento dos requisitos de integração e que define um processo sistemático para a integração de ferramentas visando apoiar o processo de medição de software. Nas iniciativas de medição investigadas na revisão sistemática da literatura realizada, notou-se que nenhuma das publicações mencionou o uso de uma abordagem sistemática para conduzir a integração.

Em OBA-MSI é explorado o uso de uma ontologia de domínio (ORMS) e uma ontologia de tarefa (OTMS) na integração de ferramentas de apoio à medição de software. OTMS é utilizada como base para o alinhamento do processo de medição da organização

ao processo de medição definido com base em diversos padrões que tratam o processo de medição de software, bem como para atribuição de semântica aos serviços envolvidos na integração, apoiando a integração nas camadas de serviço e processo. ORMS, por sua vez, fornece a conceituação relacionada à medição de software e provê o modelo estrutural de OTMS. ORMS é fundamental para a atribuição de semântica aos termos e dados envolvidos na integração, apoiando a integração na camada de dados.

Para avaliar a viabilidade de uso de OBA-MSI, a abordagem foi utilizada para conduzir a integração de ferramentas para apoiar o processo de medição do Laboratório de Extensão em Desenvolvimento de Sistemas (LEDS). Os resultados obtidos são apresentados no próximo capítulo.

Capítulo 5

Aplicação da Abordagem Proposta

Este capítulo apresenta a aplicação de OBA-MSI no Laboratório de Extensão em Desenvolvimento de Sistemas (LEDS). Na Seção 5.1 são apresentados os principais resultados da execução de cada atividade de OBA-MSI para realizar a integração de ferramentas visando apoiar o processo de medição de software no LEDS. A Seção 5.2 apresenta um survey realizado com os membros do LEDS para avaliar os resultados da integração de ferramentas realizada. Por fim, na Seção 5.3 são feitas as considerações finais do capítulo.

5.1 Integração de Ferramentas para Apoiar Medição de Software no LEDS

Para avaliar a abordagem proposta neste trabalho, OBA-MSI foi utilizada para realizar a integração de ferramentas visando apoiar o processo de medição no Laboratório de Extensão em Desenvolvimento de Sistemas (LEDS). O LEDS, localizado no Instituto Federal do Espírito Santo (IFES), campus Serra, é um espaço experimental, semelhante a uma empresa de desenvolvimento de sistemas, que faz a ponte entre a teoria e a prática, propiciando a integração entre docentes e discentes com foco na resolução de problemas da sociedade. No LEDS alunos e professores trabalham em conjunto no desenvolvimento de projetos para atender demandas apresentadas pela sociedade (LEDS, 2015).

Atualmente, o LEDS conta com equipes desenvolvendo diversos projetos de software seguindo os princípios do SCRUM (PHAM, 2011) e adota ferramentas para apoiar atividades de gerência de projetos e desenvolvimento de software. O coordenador do LEDS relatou a necessidade de obter dados para auxiliar a monitoração dos projetos e a qualidade do software produzido. Segundo o coordenador, as ferramentas utilizadas no LEDS são capazes de prover alguns dados, mas, além de os dados estarem dispersos em diferentes ferramentas, o que dificulta seu acesso, manipulação e utilização, outros dados derivados dos fornecidos diretamente pelas ferramentas também seriam necessários.

Considerando-se esse cenário, decidiu-se por aplicar OBA-MSI para integrar ferramentas visando apoiar o processo de medição no LEDS. A seguir são apresentados os principais resultados produzidos em cada atividade de OBA-MSI.

5.1.1 Definição do Processo de Medição alinhado à Ontologia de Tarefa de Medição de Software (OTMS)

Conforme apresentado no Capítulo 4, a primeira atividade de OBA-MSI consiste em alinhar o processo de medição da organização à OTMS. No momento da aplicação da abordagem, o LEDS não possuía um processo de medição formalmente definido. Dessa forma, seguindo as orientações de OBA-MSI, analisou-se o processo de desenvolvimento da organização e foi feita a identificação das atividades onde ocorrem atividades relacionadas à medição de software. A Figura 5.1 apresenta uma visão geral do processo de desenvolvimento adotado no LEDS. A Figura 5.2 apresenta o processo em detalhes, destacando-se as atividades onde ocorre medição de software. Em seguida, na Tabela 5.1, uma breve descrição do processo é apresentada e na Tabela 5.2 são disponibilizadas informações sobre as atividades relacionadas à medição que ocorrem nas atividades destacadas na Figura 5.2.

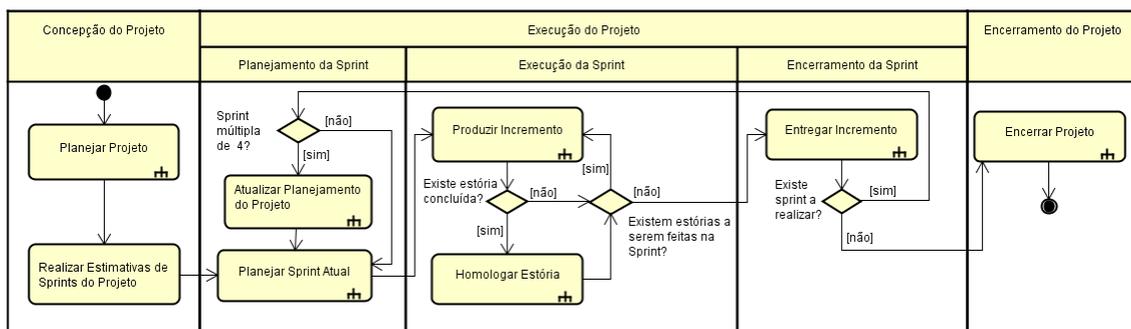


Figura 5.1 - Visão Geral do Processo de Desenvolvimento de Software adotado no LEDS

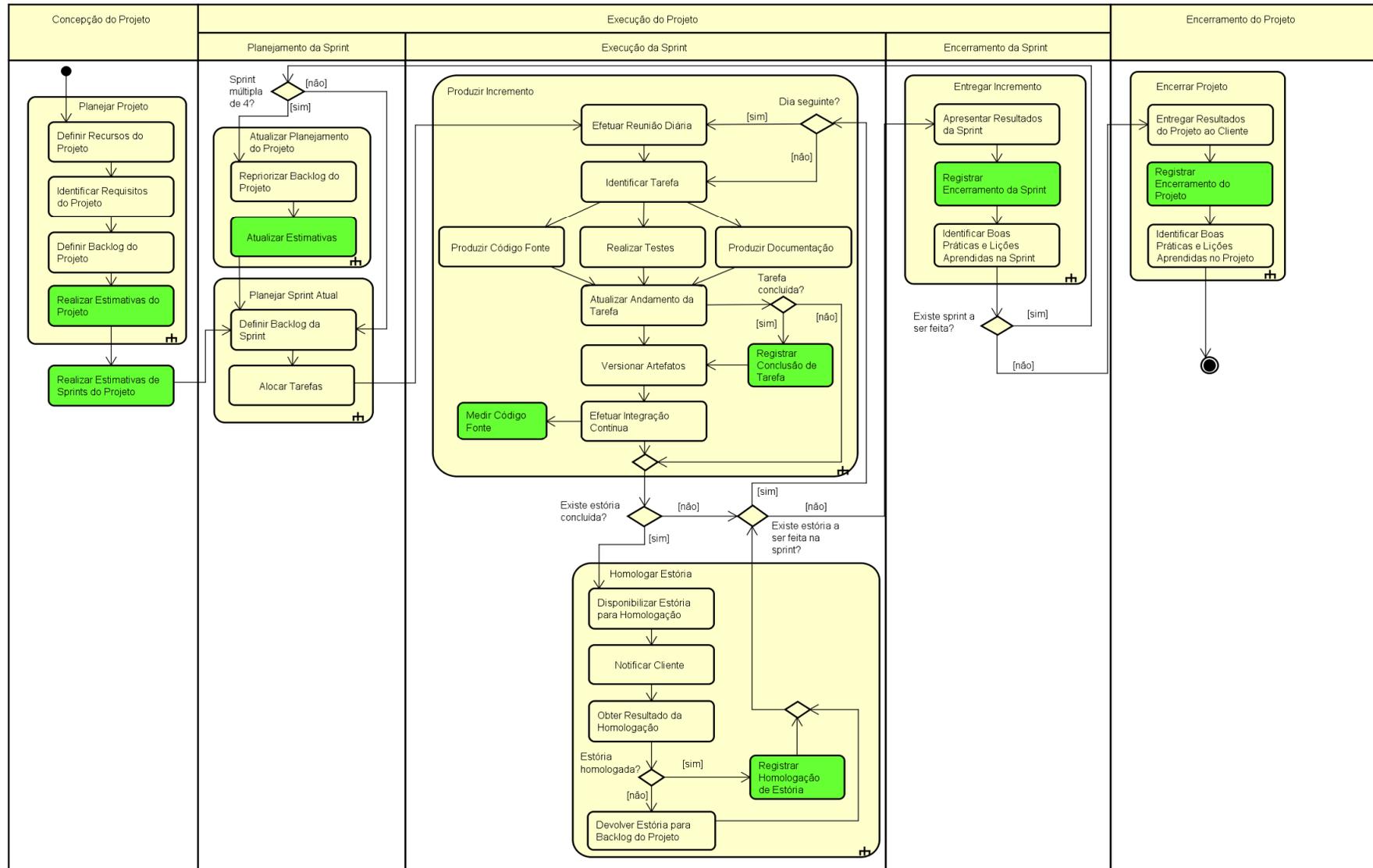


Figura 5.2 - Visão Detalhada do Processo de Desenvolvimento de Software adotado no LEDSS, destacando-se atividades onde ocorre medição de software.

Tabela 5.1 - Descrição das atividades do Processo de Desenvolvimento adotado no LEDS.

Atividade/subatividade	Descrição
Planejar Projeto	Nesta atividade é realizado o planejamento geral do projeto. Envolve:
<i>Definir Recursos do Projeto</i>	Os recursos necessários (hardware, software e recursos humanos) são alocados ao projeto.
<i>Identificar Requisitos do Projeto</i>	Os requisitos do projeto são levantados junto ao cliente.
<i>Definir Backlog do Projeto</i>	Os requisitos identificados são transformados em estórias, que são registradas e priorizadas no <i>backlog</i> do projeto.
<i>Realizar Estimativas do Projeto</i>	O tamanho das estórias (em pontos de estória), os pontos de estória do projeto (somatório dos pontos de todas as estórias) e a velocidade da equipe (quantos pontos de estória a equipe é capaz de desenvolver em uma <i>sprint</i>) são estimados.
Realizar Estimativas de Sprints do Projeto	Nesta atividade são realizadas as estimativas do número de <i>sprints</i> do projeto (o projeto é subdividido em <i>sprints</i> levando-se em consideração a velocidade da equipe e os pontos de estória do projeto) e da quantidade de estórias para as <i>sprints</i> (levando-se em consideração a velocidade da equipe e o tamanho estimado para as estórias do projeto, estima-se o número de estórias das <i>sprints</i>).
Planejar Sprint Atual	Nesta atividade é realizado o planejamento da <i>sprint</i> a ser executada. Envolve:
<i>Definir Backlog da Sprint</i>	Estórias do <i>backlog</i> do projeto são selecionadas para serem produzidas durante a <i>sprint</i> . A seleção das estórias ocorre de acordo com a velocidade da equipe, de forma que o total de pontos de estórias não ultrapasse a velocidade estimada para a equipe.
<i>Alocar Tarefas</i>	São definidas tarefas (pequenas unidades de trabalho as quais um membro da equipe seja capaz de realizar) para o desenvolvimento das estórias e membros da equipe são alocados às tarefas.
Produzir Incremento	Nesta atividade é executado o trabalho definido para a <i>sprint</i> a fim de se produzir um incremento de software. O trabalho é dividido em ciclos diários. Envolve:
<i>Efetuar Reunião Diária</i>	Reunião diária cujo objetivo é verificar o que foi feito no dia anterior, o que será feito no dia atual e se existe algum impedimento para que as tarefas do dia sejam executadas.
<i>Identificar Tarefa</i>	A próxima tarefa a ser realizada por cada membro da equipe é identificada e o ambiente para realizá-la é preparado (ex.: ferramentas). As tarefas a serem realizadas pelos membros do projeto são tarefas relacionadas a implementação, testes e documentação do software.
<i>Produzir Código Fonte</i>	O código fonte do software é produzido.
<i>Realizar Testes</i>	O software é testado.
<i>Produzir Documentação</i>	Documentos, diagramas e modelos são elaborados.
<i>Atualizar Andamento da Tarefa:</i>	O andamento da tarefa realizada por cada membro é atualizado de acordo com o trabalho feito.
<i>Registrar Conclusão de Tarefa</i>	A conclusão de uma tarefa é registrada.
<i>Efetuar Integração Contínua</i>	Os artefatos produzidos são submetidos ao controle de versão.

Tabela 5.1 - Descrição das atividades do Processo de Desenvolvimento adotado no LEDS (cont.).

Atividade/subatividade	Descrição
<i>Versionar Artefatos</i>	O código fonte e os testes são submetidos ao motor de Integração Contínua. É produzido um executável do software e o código de teste é executado. Caso todos os testes sejam bem-sucedidos, o executável é implantado em produção. Caso algum teste não seja bem-sucedido, a construção do executável não é finalizada e a equipe é notificada para que as correções necessárias sejam realizadas. As tarefas relacionadas às correções necessárias são atribuídas aos membros na atividade <i>Identificar Tarefa</i> .
<i>Medir Código Fonte</i>	Dados relacionados ao código fonte produzido são disponibilizados.
Homologar Estória	Nesta atividade uma estória concluída é homologada pelo cliente. Envolve:
<i>Disponibilizar Estória para Homologação</i>	A versão do software contendo a estória concluída é disponibilizada para homologação pelo cliente.
<i>Notificar Cliente</i>	O cliente é notificado de que há uma nova versão do software contendo uma nova estória, disponível para sua homologação.
<i>Obter Resultado da Homologação</i>	A aprovação ou reprovação da estória pelo cliente é informada e avaliada pela equipe.
<i>Registrar Homologação de Estória</i>	Caso a estória seja homologada, ela é registrada como concluída.
<i>Devolver Estória para Backlog do Projeto</i>	Caso a estória não seja aprovada pelo cliente, ela é retornada ao <i>backlog</i> do projeto para ser tratada em uma nova <i>sprint</i> .
Entregar Incremento	Nesta atividade é feita a entrega do incremento de software produzido durante a execução da <i>sprint</i> . Envolve:
<i>Apresentar Resultados da Sprint</i>	Os resultados produzidos na <i>sprint</i> são apresentados e entregues para o cliente em uma reunião. É criado um marco no repositório de controle de versão contendo o software e os artefatos entregues.
<i>Registrar Encerramento da Sprint</i>	A <i>sprint</i> é registrada como concluída e os dados relacionados à execução da <i>sprint</i> são disponibilizados.
<i>Identificar Boas Práticas e Lições Aprendidas na Sprint</i>	Boas práticas e lições aprendidas na <i>sprint</i> são identificadas e avaliadas pela equipe, sendo registradas para posterior consulta e reutilização.
Atualizar Planejamento do Projeto	Após a entrega do incremento produzido em uma <i>sprint</i> , a próxima <i>sprint</i> é executada até não restarem mais <i>sprints</i> . Caso a próxima <i>sprint</i> a ser executada seja uma <i>sprint</i> múltipla de quatro (por exemplo, quarta e oitava <i>sprints</i> do projeto), o planejamento do projeto deve ser atualizado. Envolve:
<i>Repriorizar Backlog do Projeto</i>	As estórias do <i>backlog</i> do projeto são repriorizadas de acordo com a necessidade do cliente.
<i>Atualizar Estimativas</i>	O tamanho das estórias, os pontos de estória do projeto e a velocidade da equipe são atualizados.
Encerrar Projeto	Caso não haja mais <i>sprints</i> a serem executadas ou o cliente solicite o encerramento do projeto, o projeto é encerrado. Envolve:
<i>Entregar Resultados do Projeto</i>	Os resultados produzidos durante o projeto são apresentados e entregues ao cliente.
<i>Registrar Encerramento do Projeto</i>	O projeto é registrado como concluído e os dados relacionados à sua execução são disponibilizados.
<i>Identificar Boas Práticas e Lições Aprendidas no Projeto</i>	Boas práticas e lições aprendidas no projeto são identificadas e avaliadas pela equipe, sendo registradas para posterior consulta e reutilização.

Tabela 5.2 - Atividades de medição de software que ocorrem no contexto de atividades do processo de desenvolvimento adotado no LEDS.

Atividades LEDS	Atividades de Medição de Software (segundo OTMS)	Comentários
Realizar Estimativas do Projeto	Executar Medições	Ao realizar as estimativas para o projeto, são atribuídos valores para medidas relacionadas às estimativas para a velocidade da equipe, tamanho das estórias do projeto e quantidade de pontos de estória do projeto.
Realizar Estimativas de <i>Sprints</i> do Projeto	Executar Medições	A estimativa do número de <i>sprints</i> do projeto envolve a atribuição de um valor à medida relacionada ao tamanho do projeto em <i>sprints</i> .
Atualizar Estimativas	Executar Medições	Novos valores são atribuídos às medidas que quantificam a velocidade da equipe e o tamanho das estórias do projeto.
Registrar Conclusão de Tarefa	Executar Medições	Ao concluir uma tarefa medições relacionadas à quantidade de tarefas são realizadas (por exemplo, o número de tarefas concluídas pelo membro da equipe é atualizado, bem como o número de tarefas concluídas na <i>sprint</i> e no projeto).
Medir Código Fonte	Executar Medições	Durante a integração contínua, são atribuídos valores a medidas relacionadas ao tamanho, duplicações e complexidade ciclomática do código fonte.
Registrar Homologação de Estória	Executar Medições	Ao concluir uma estória, é atribuído um novo valor para a velocidade da equipe e medidas relacionadas ao número de estórias concluídas são atualizadas.
Registrar Encerramento da <i>Sprint</i>	Executar Medições	O encerramento da <i>sprint</i> leva à atualização de medidas relacionadas ao desempenho na <i>sprint</i> e no projeto.
Registrar Encerramento do Projeto	Executar Medições	O encerramento do projeto leva à atualização de medidas relacionadas ao desempenho do projeto.

Analisando-se a Tabela 5.2, percebe-se que no contexto das atividades do processo de desenvolvimento do LEDS ocorre a atividade *Executar Medições* do processo de medição de software. Entretanto, embora não esteja explícito nas atividades do processo de desenvolvimento, as atividades *Planejar Medição* e *Analisar Medições* também são realizadas no LEDS. Por exemplo, ao se iniciar um projeto, são planejadas algumas metas que são definidas por meio de medidas (por exemplo, são definidos valores desejados e aceitáveis para a taxa de duplicações no código produzido no projeto) e no encerramento de uma *sprint* ou do projeto o alcance a essas metas é verificado (por exemplo, analisam-se os valores coletados para a taxa de duplicações no código produzido no projeto e verifica-se se os valores estabelecidos foram alcançados).

Ainda que o LEDS realize atividades relacionadas à medição, o processo de medição não era explícito. Para tornar o processo de medição explícito, tomou-se como base as atividades do processo de desenvolvimento do LEDS onde ocorre medição e as atividades do processo de medição de software estabelecido pela OTMS e definiu-se o processo de medição do LEDS. O resultado é apresentado na Figura 5.3. As atividades em destaque são atividades oriundas do processo de desenvolvimento e as demais atividades foram incluídas com base na OTMS.

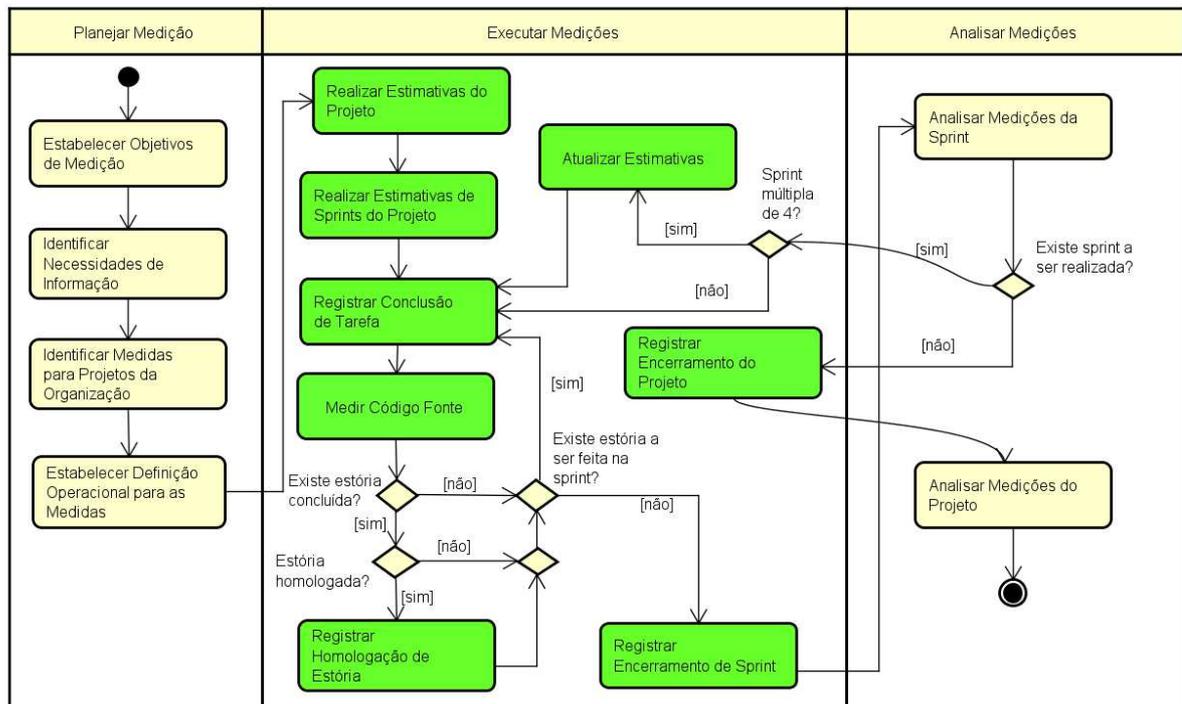


Figura 5.3 - Processo de Medição definido a partir do Processo de Desenvolvimento de Software do LEDS e da OTMS.

Embora o processo de medição definido não seja igual ao processo estabelecido pela OTMS, há alinhamento entre eles. As três atividades principais de OTMS são representadas nas partições do processo de medição do LEDS. Para a atividade *Planejar Medição*, foram definidas as mesmas subatividades estabelecidas na OTMS.

Para tratar a atividade *Executar Medições*, foram mantidas as atividades do processo de desenvolvimento nas quais ocorre execução de medições. Cada atividade presente na partição *Executar Medições* pode ser entendida como uma representação da atividade *Executar Medições* definida na OTMS, porém cada atividade na partição considera medidas específicas. Por exemplo, em *Realizar Estimativas do Projeto* são realizadas medições para medidas relacionadas a velocidade da equipe, tamanho das histórias do projeto e quantidade de pontos de história do projeto.

Por fim, para tratar a atividade *Analisar Medições*, de maneira similar a *Executar Medições*, foram incluídas atividades para lidar com análise de medições específicas. Assim, cada atividade da partição *Analisar Medições* pode ser entendida como uma representação da atividade *Analisar Medições* definida na OTMS, porém considerando medições relacionadas a uma *sprint* do projeto ou ao projeto como um todo.

Embora, por questões de simplificação, não tenham sido definidas explicitamente no processo de medição do LEDS as subatividades de *Executar Medições* e *Analisar Medições* presentes na OTMS, elas estão implícitas nas atividades definidas. Por exemplo, em *Realizar Estimativas do*

Projeto, ao se selecionar a medida a ser coletada (por exemplo, velocidade da equipe) é executada a atividade *Selecionar Item do Plano de Medição para Medição*, ao se identificar o projeto para o qual será estimada a velocidade da equipe é executada a atividade *Selecionar Entidade Mensurável para Medição*, e ao se estimar o valor da velocidade da equipe é executada a atividade *Coletar Dados*. Analogamente, em *Analisar Medições da Sprint*, ao se definir que se deseja avaliar o desempenho de um membro do projeto na *sprint* é executada a atividade *Selecionar Objetivo de Medição para Análise*, ao se selecionar as medidas é executada a atividade *Selecionar Medidas para Análise*, quando são selecionados os dados referentes à *sprint* é executada a atividade *Selecionar Dados para Análise*, quando os dados são analisados é executada a atividade *Analisar Dados* e quando os resultados são apresentados para a equipe é executada a atividade *Reportar Resultados*.

5.1.2 Identificação de Objetivos, Necessidades de Informação e Medidas Relevantes para a Organização

Conforme sugere OBA-MSI, a identificação dos objetivos, necessidades de informação e medidas foi realizada de maneira iterativa com a atividade de identificação das ferramentas a serem integradas, que é apresentada na Subseção 5.1.3, a fim de garantir que as medidas identificadas fossem passíveis de serem obtidas a partir das ferramentas selecionadas para integração. Inicialmente, foram definidos os objetivos estratégicos da organização, que serviriam como motivadores para a realização da medição. Foram identificados dois objetivos estratégicos (OE): *Melhorar o gerenciamento dos projetos de software da organização* e *Melhorar a qualidade do código fonte produzido*. A partir desses objetivos foram identificados os objetivos de medição (OM), as necessidades de informação (NI) e as medidas (ME), apresentados na Tabela 5.3.

Tabela 5.3 - Objetivos, Necessidades de Informação e Medidas identificadas para o LEADS.

<p><i>OE - Melhorar o gerenciamento dos projetos de software da organização</i></p> <p>OM – Monitorar a conclusão de pontos de estória nos projetos</p> <p>NI - Quantos pontos de estória foram planejados para o projeto? ME - Pontos de Estória Planejados para o Projeto (PEPP)</p> <p>NI - Quantos pontos de estória foram concluídos no projeto? ME - Pontos de Estória Concluídos no Projeto (PECP)</p> <p>NI – Qual a taxa de conclusão de pontos de estória no projeto? ME – Taxa de Conclusão de Pontos de Estória no Projeto (TCPEP)</p> <p>OM – Monitorar a realização de sprints nos projetos</p> <p>NI - Quantas sprints foram planejadas para o projeto? ME - Número de Sprints Planejadas para o Projeto (NSPP)</p> <p>NI - Quantas sprints foram realizadas no projeto?</p>

Tabela 5.3 - Objetivos, Necessidades de Informação e Medidas identificadas para o LEADS (cont.).

<p>ME - Número de Sprints Realizadas no Projeto (NSRP)</p> <p>NI - Qual a taxa de conclusão de sprints no projeto? ME - Taxa de Conclusão de Sprints no Projeto (TCSP)</p> <p>OM – Monitorar desempenho na sprint</p> <p>NI – Quantas estórias foram planejadas para a sprint? ME - Número de Estórias Planejadas para a Sprint (NEPS)</p> <p>NI - Quantas estórias foram concluídas na sprint? ME - Número de Estórias Concluídas na Sprint (NECS)</p>
--

- NI – Qual a taxa de conclusão de estórias na sprint?
ME - Taxa de Conclusão de Estórias na Sprint (TCES)
- NI - Quantos pontos de estória foram planejados para a sprint?
ME - Pontos de Estória Planejados para a sprint (PEPS)
- NI - Quantos pontos de estória foram concluídos na sprint?
ME – Pontos de Estória Concluídos na sprint (PECS)
- NI – Qual a taxa de conclusão de pontos de estórias na sprint?
ME - Taxa de Conclusão de Pontos de Estórias na Sprint (TCPES)
- NI - Quantas tarefas foram planejadas para a sprint?
ME - Número de Tarefas Planejadas para a Sprint (NTPS)
- NI - Quantas tarefas foram concluídas na sprint?
ME - Número de Tarefas Concluídas na Sprint (NTCS)
- NI – Qual a taxa de conclusão de tarefas na sprint?
ME - Taxa de Conclusão de Tarefas na Sprint (TCTS)

OM – Monitorar desempenho no projeto

- NI - Quantas sprints foram realizadas no projeto?
ME - Número de Sprints Realizadas no Projeto (NSRP)
- NI - Quantas estórias foram concluídas para o projeto?
ME - Número de Estórias Concluídas para o Projeto (NECP)
- NI - Quantos pontos de estória foram concluídos no projeto?
ME - Pontos de Estória Concluídos no Projeto (PECP)
- NI - Qual o número médio de estórias concluídas por sprint no projeto?
ME - Média de Estórias Concluídas por Sprint do Projeto (MECSP)
- NI - Qual o número médio de pontos de estórias concluídos por sprint no projeto?
ME - Velocidade da Equipe no Projeto (VEP)

OM – Monitorar quantidade de doses de Iocaine nas sprints

- NI - Quantas *doses de Iocaine* ocorreram na sprint?
ME - Número de Doses de Iocaine na Sprint (NDIS)
- NI - Quantas tarefas foram concluídas na sprint?
ME - Número de Tarefas Concluídas na Sprint (NTCS)
- NI – Qual a taxa de *doses de Iocaine* na sprint?
ME - Taxa de Doses de Iocaine na Sprint (TDIS)

OM – Monitorar o desempenho dos membros da equipe no projeto

- NI - Quantas tarefas foram atribuídas a um membro da equipe do projeto?
ME - Número de Tarefas Atribuídas a Membro do Projeto (NTAM)
- NI - Quantas tarefas foram concluídas por um membro da equipe do projeto?
ME - Número de Tarefas Concluídas pelo Membro do Projeto (NTCM)
- NI – Qual a taxa de conclusão de tarefas de um membro da equipe do projeto?
ME - Taxa de Conclusão de Tarefas de Membro do Projeto (TCTM)
- NI - Quantos pontos de estória foram atribuídos a um membro da equipe do projeto?
ME - Número de Pontos de Estória Atribuídos a Membro do Projeto (NPEAM)
- NI - Quantos pontos de estória foram concluídos por um membro da equipe do projeto?
ME - Número de Pontos de Estória Concluídos pelo Membro do Projeto (NPECM)
- NI – Qual a taxa de conclusão de pontos de estória de um membro da equipe do projeto?
ME - Taxa de Conclusão de Pontos de Estória de Membro do Projeto (TCPEM)
- NI - Quantas *doses de Iocaine* foram atribuídas a um membro da equipe do projeto?
ME - Número de Doses de Iocaine Atribuídas a Membro do Projeto (NDIAM)
- NI - Qual a taxa de *doses de Iocaine* de um membro da equipe do projeto?
ME - Taxa de Doses de Iocaine de Membro do Projeto (TDIM)

OE - Melhorar a qualidade do código fonte produzido nos projetos

OM – Monitorar a qualidade do código fonte produzido nos projetos

- NI - Qual a complexidade ciclomática média por método?
ME - Média da Complexidade Ciclomática por Método (MCCM)
- NI - Qual a taxa de duplicação de código?
ME - Taxa de Duplicação de Código (TDC)
- NI - Qual o percentual da dívida técnica?
ME - Percentual da Dívida Técnica (PDT)

5.1.3 Identificação das Ferramentas Fornecedoras de Dados/Serviços para a Medição de Software

Para selecionar as ferramentas a serem integradas, seguindo as diretrizes de OBA-MSI, inicialmente foram identificadas as ferramentas utilizadas pela organização e que apoiam as atividades do processo de desenvolvimento onde ocorre alguma atividade de medição. Neste momento foram selecionadas as ferramentas Taiga⁵ e SonarQube⁶. Em seguida, observou-se que ambas as ferramentas possuem uma API de serviços e a documentação completa da API está disponível no site da empresa fabricante, estando em linha com as recomendações para seleção de ferramentas de OBA-MSI.

Taiga é uma ferramenta de apoio ao gerenciamento ágil de projetos baseado no Scrum. Permite o controle das atividades do projeto e utiliza conceitos do Scrum como *sprint*, *estória* e *tarefas*, entre outros.

SonarQube é uma ferramenta de análise estática, específica para o controle de qualidade do código fonte. A partir de um conjunto de regras e padrões, o SonarQube coleta diversos dados sobre o código fonte de um software.

A Figura 5.4 representa o processo de desenvolvimento do LEDES incluindo as ferramentas e as atividades apoiadas por cada uma. As ferramentas são representadas por ilustrações de bases de dados. Os fluxos numerados na figura representam fluxos de dados de medição entre as ferramentas e as atividades. O detalhamento dos dados presentes em cada fluxo (identificado pelo número na figura) é descrito na Tabela 5.4.

As ferramentas utilizadas no LEDES apoiam a medição através da captura e armazenamento de dados relacionados à medição. No entanto, nenhuma delas é uma ferramenta desenvolvida para apoiar o processo de medição e, dessa forma, não proveem funcionalidades para o planejamento da medição e a análise de medições. Então, seguindo-se as diretrizes de OBA-MSI, foi necessário selecionar uma ferramenta de medição capaz de prover o apoio necessário a essas atividades, tendo sido selecionada a ferramenta SoMeSPC⁷ (*Software Measurement & Statistical Process Control*).

A seleção de SoMeSPC foi motivada pelas seguintes razões: (i) ela provê apoio a todas as atividades do processo de medição de software; (ii) foi desenvolvida com base em uma Arquitetura de Referência de Medição de Software que teve como base a primeira versão da Ontologia de Referência para Medição de Software (BARCELLOS, 2009); (iii) possui

⁵ <https://taiga.io/>

⁶ <http://www.sonarqube.org/>

⁷ <https://github.com/nemo-ufes/SoMeSPC>

documentação de seus modelos conceituais; *(iv)* possui código fonte aberto; e *(v)* foi desenvolvida em (MARETTO, 2013), trabalho realizado do NEMO, o que facilita contato com o desenvolvedor da ferramenta.

Vale comentar que, embora SoMeSPC seja uma ferramenta concebida para apoiar o processo de medição, ela foi desenvolvida como uma prova de conceito e apresenta alguns problemas de usabilidade. O registro dos dados em SoMeSPC deve ser realizado manualmente, o que demanda considerável esforço. Além disso, algumas funcionalidades providas pela ferramenta requerem que o usuário tenha conhecimento em medição de software para que os dados corretos sejam fornecidos. Antes de iniciar a integração de ferramentas para o LEDS, SoMeSPC foi apresentada e o feedback recebido foi consistente com uma avaliação da ferramenta feita anteriormente (MARETTO, 2013). Embora a ferramenta apoie o processo de medição e tenha funcionalidades muito úteis, seu uso no dia a dia de uma organização é desmotivado devido ao esforço necessário para a entrada de dados. Dessa forma, foi necessário evoluir SoMeSPC para permitir sua utilização em ambientes organizacionais reais (CASTRO, 2015).

Tabela 5.4 - Fluxos de dados de medição entre ferramentas e atividades.

<p>1) <i>Taiga</i> → <i>Realizar Estimativas do Projeto</i>: velocidade da equipe em projetos anteriores (média de pontos de histórias concluídos por <i>Sprint</i>), tamanho das histórias em projetos anteriores. <i>Realizar Estimativas do Projeto</i> → <i>Taiga</i>: tamanho estimado para as histórias do projeto, quantidade de pontos de história planejados para o projeto, velocidade estimada para a equipe no projeto.</p> <p>2) <i>Taiga</i> → <i>Realizar Estimativas de Sprints do Projeto</i>: velocidade estimada para a equipe no projeto, quantidade de pontos de história planejados para o projeto. <i>Estimar Número de Sprints do Projeto</i> → <i>Taiga</i>: número de <i>sprints</i> planejadas para o projeto, número de histórias planejadas para cada <i>sprint</i>.</p> <p>3) <i>Taiga</i> → <i>Atualizar Estimativas</i>: velocidade real da equipe no projeto até o momento, tamanho planejado para as histórias do projeto, tamanho real das histórias concluídas até o momento. <i>Atualizar Estimativas</i> → <i>Taiga</i>: velocidade estimada para a equipe no projeto atualizada, tamanho estimado para as histórias do projeto atualizado.</p> <p>4) <i>SonarQube</i> → <i>Medir Código Fonte</i>: duplicações (taxa de duplicação de código), complexidade ciclomática (média da complexidade ciclomática por método) e dívida técnica (percentual da dívida técnica).</p> <p>5) <i>Registrar Conclusão de Tarefa</i> → <i>Taiga</i>: novos valores para as medidas relacionadas a número de tarefas concluídas.</p> <p>6) <i>Registrar Homologação de História</i> → <i>Taiga</i>: novos valores para as medidas relacionadas a número de histórias concluídas, velocidade da equipe no projeto atualizada.</p> <p>7) <i>Registrar Encerramento da Sprint</i> → <i>Taiga</i>: novos valores para as medidas relacionadas a número de <i>sprints</i> concluídas.</p> <p>8) <i>Registrar Encerramento do Projeto</i> → <i>Taiga</i>: novos valores para as medidas relacionadas ao projeto.</p>

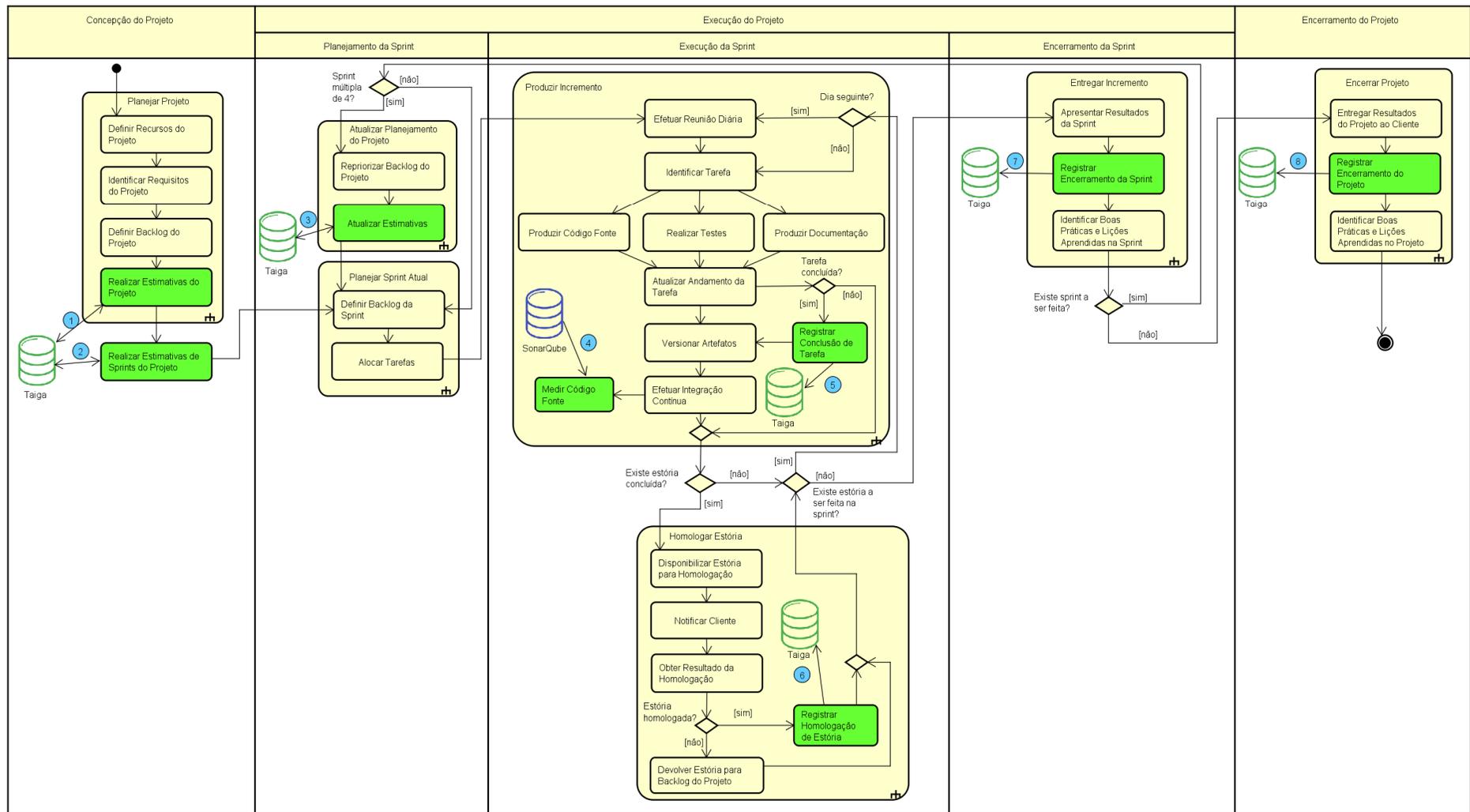


Figura 5.4 - Ferramentas que apoiam atividades onde ocorre medição.

5.1.4 Registro do Cenário de Integração

O objetivo da integração é apoiar a realização do processo de medição de software adotado pelo LEDES, considerando os objetivos, necessidades de informação e medidas identificadas. Dessa forma, as atividades a serem apoiadas pela integração das ferramentas são as atividades do processo de medição de software. A Tabela 5.5 apresenta o cenário da integração.

Tabela 5.5 - Cenário de Integração.

Sistemas	SoMeSPC, Taiga e SonarQube
Domínio	Medição de Software aplicada à Gerência de Projetos Ágeis e Codificação
Atividades	<ul style="list-style-type: none">• Planejar Medição• Executar Medições• Analisar Medições

5.1.5 Análise da Integração

Segundo OBA-SI, para realizar a análise da integração, é necessário: selecionar a ontologia a ser utilizada, recuperar os modelos conceituais das ferramentas a serem integradas, realizar mapeamentos verticais, elaborar modelo de integração e realizar mapeamentos horizontais.

5.1.5.1 Selecionar Ontologia para a Integração

O primeiro passo para realizar a análise da integração é selecionar a ontologia a ser utilizada. Conforme discutido no Capítulo 4, em OBA-MSI advoga-se pelo uso das ontologias Ontologia de Referência para Medição de Software (ORMS) e Ontologia de Tarefa de Medição de Software (OTMS), apresentadas no Capítulo 2, devendo ser identificados os fragmentos relevantes para a integração.

Conforme descrito no Capítulo 2, ORMS reutiliza conceitos da Ontologia de Processos de Software (OPS) (BRINGUENTE; FALBO; GUIZZARDI, 2011). Considerando-se que o domínio da integração inclui a gerência de projetos ágeis e que OPS não tratava aspectos relacionados a processos ágeis, foi necessário estendê-la. A Figura 5.5 apresenta o fragmento de OPS estendido. Os conceitos incluídos para tratar processos ágeis estão em branco na figura.

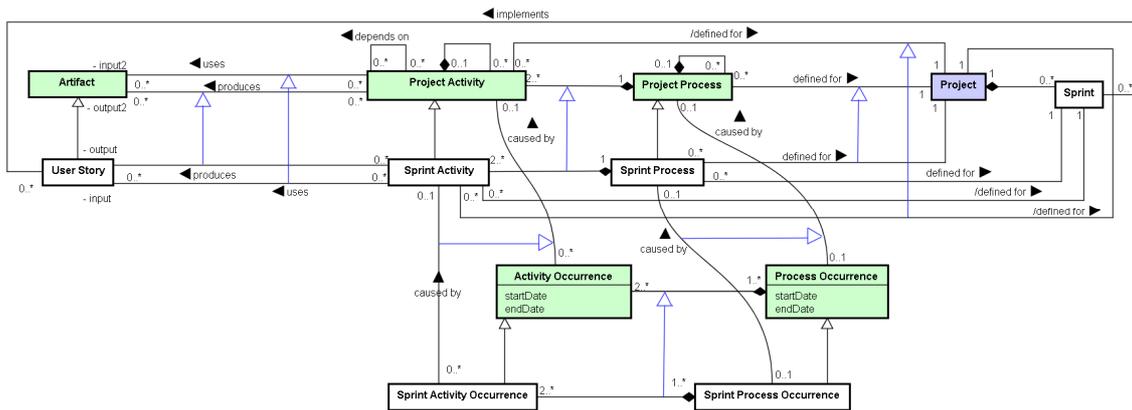


Figura 5.5 - Fragmento de OPS estendido para tratar processos ágeis.

Um *Project* (Projeto) pode ser composto por *Sprints* (Sprint). *Sprint Process* (Processo de Sprint) é um tipo de *Project Process* (Processo de Projeto) que é definido para uma *Sprint*. De forma análoga, *Sprint Activity* (Atividade de Sprint) é um tipo de *Project Activity* (Atividade de Projeto) que é definida para uma *Sprint*. *Sprint Activities* podem produzir ou usar *User Stories* (Estórias de Usuário), que são um tipo de *Artifact* (Artefato). *User Stories* produzidas em *Sprint Activities* definidas para uma dada *Sprint* são *User Stories* implementadas nessa *Sprint*. *Sprint Process Occurrence* (Ocorrência de Processo de Sprint) é um tipo de *Process Occurrence* (Ocorrência de Processo) que ocorre no contexto de uma *Sprint* e pode ser causada por um *Sprint Process*. Analogamente, uma *Sprint Activity Occurrence* (Ocorrência de Atividade de Sprint) é um tipo de *Activity Occurrence* (Ocorrência de Atividade) que ocorre no contexto de uma *Sprint* e pode ser causada por uma *Sprint Activity*.

Após a extensão de OPS, foi necessário estender ORMS para considerar os conceitos inclusos em OPS relevantes para a integração e identificar o fragmento da ORMS relevante para a integração. A Figura 5.6 apresenta esse fragmento.

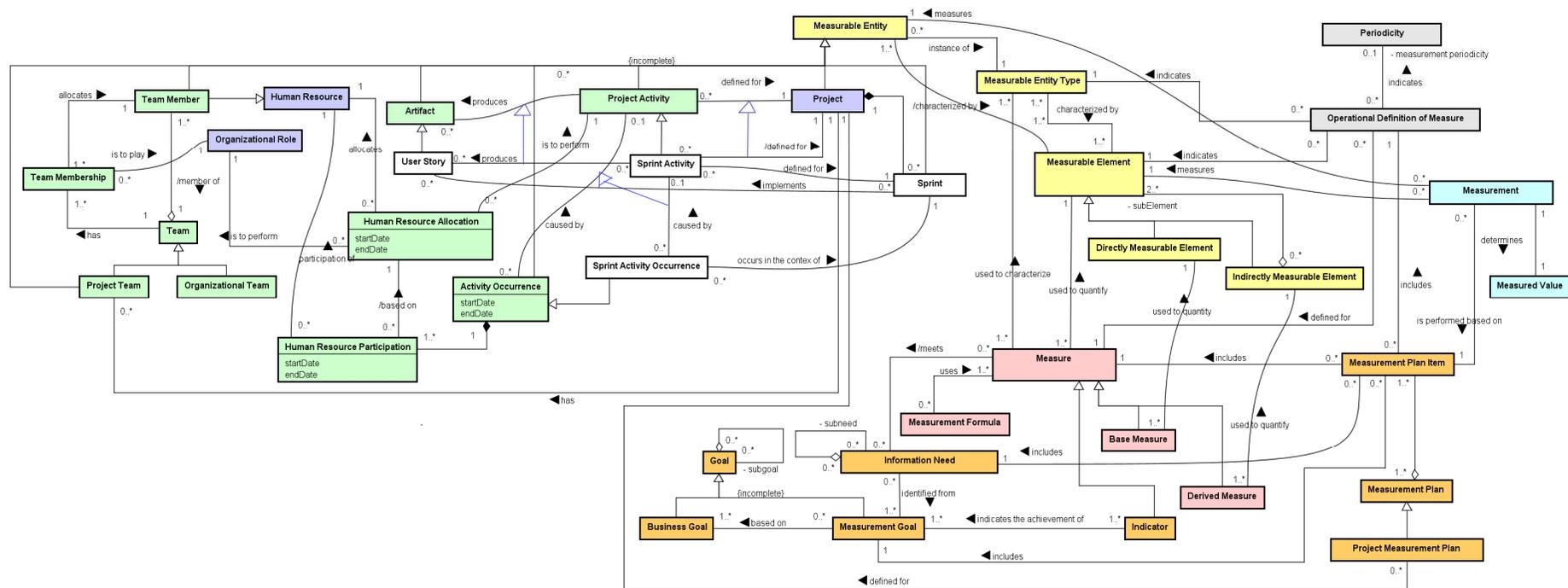


Figura 5.6 - Fragmento de ORMS considerado na integraço.

5.1.5.2 Modelos Conceituais das Ferramentas

Os fragmentos dos modelos estruturais das ferramentas relevantes à integração foram obtidos a partir de análise da documentação da ferramenta, no caso de SoMeSPC, e de escavação de modelos conceituais estruturais, nos casos de Taiga e SonarQube. A Figura 5.7 apresenta o fragmento do modelo estrutural de Taiga relevante à integração. Em seguida é apresentada uma breve descrição do modelo.

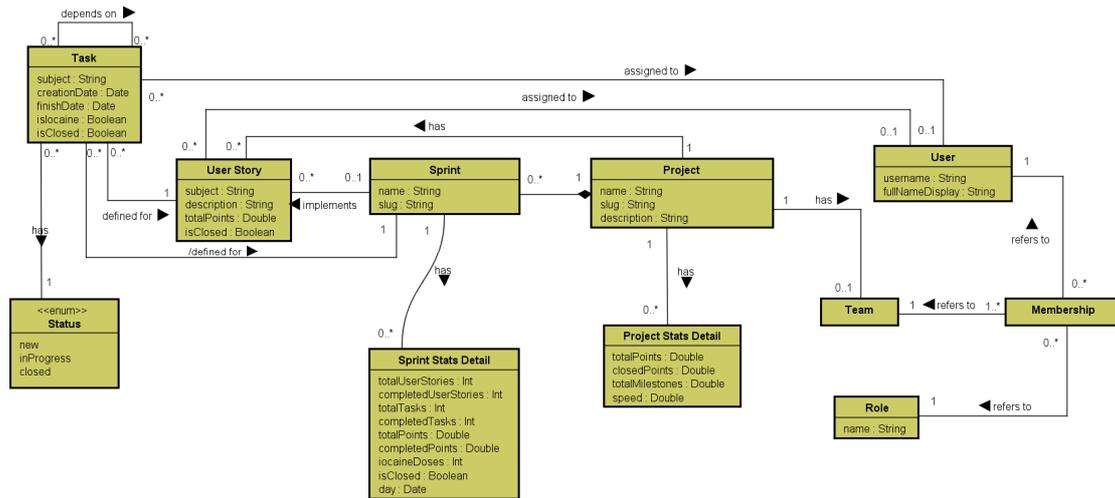


Figura 5.7 - Fragmento de Taiga relevante à integração.

Um **Project** (Projeto) é composto por **Sprints** (*Sprint*) nas quais são implementadas **User Stories** (Estórias de Usuário) do Projeto. **Tasks** (Tarefas) são definidas para implementar **User Stories** e são atribuídas a **Users** (Usuários). **Tasks** possuem **status** (situação), que pode ser **new** (nova), **inProgress** (em andamento) ou **closed** (concluída). Tarefas complexas que demandam maior esforço para serem realizadas são chamadas *doses de Iocane* (**isIocaine** = true). **Users** são alocados à equipe do projeto (**Team**) através de **Membership** (Filiação), que indica para cada **User** o **Role** (Papel) a ser desempenhado. **Project Stats Detail** (Estatísticas Detalhadas do Projeto) armazena dados a respeito de um Projeto. Analogamente, **Sprint Stats Detail** (Estatísticas Detalhadas da *Sprint*) armazena dados a respeito de uma *Sprint*. As propriedades de **Project Stats Detail** e **Sprint Stats Detail** referem-se a medidas associadas respectivamente a Projeto e *Sprint*. Por exemplo, a propriedade *totalTasks* em **Sprint Stats Detail** refere-se à quantidade de tarefas planejadas para a *Sprint*.

A Figura 5.8 apresenta o fragmento do modelo estrutural de SonarQube relevante à integração. Em seguida é apresentada uma breve descrição do modelo.

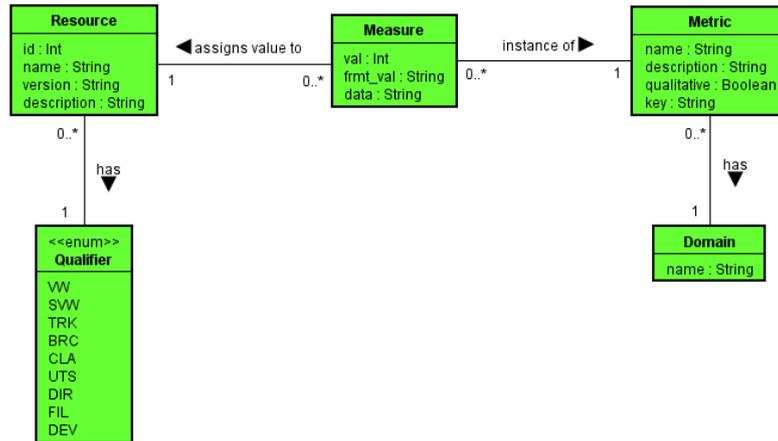


Figura 5.8 - Fragmento de SonarQube relevante à integração.

Resource (Recurso) representa as entidades que podem ser submetidas à análise de código. Um **Resource** tem um **Qualifier** (Qualificador) que indica seu tipo, podendo ser: **VW** (visão de banco de dados), **SWW** (subvisão de um banco de dados), **TRK** (projeto de código fonte de software), **BRC** (código de módulo de um software), **UTS** (código de teste unitário), **DIR** (diretório), **FIL** (arquivo) e **DEV** (desenvolvedor). Uma **Metric** (Métrica) é uma medida e possui um **Domain** (Domínio), que designa o atributo medido por **Metric** (por exemplo, complexidade e tamanho). Uma **Measure** (Medida) representa a medição de um **Resource** aplicando-se uma **Metric**. As propriedades **val** e **data** em **Measure** armazenam os valores medidos, sendo **val** para valores numéricos e **data** para valores alfanuméricos. A propriedade **fmt_val** fornece o valor medido formatado para visualização de relatórios na ferramenta.

A Figura 5.9 apresenta o fragmento do modelo estrutural de SoMeSPC relevante à integração. Conforme mencionado anteriormente, SoMeSPC foi desenvolvida a partir de uma Arquitetura de Referência para Medição de Software definida com base na primeira versão de ORMS. Embora a versão atual de ORMS, usada na integração, tenha algumas diferenças em relação à versão original, há consistência entre a maioria dos conceitos do modelo estrutural de SoMeSPC e ORSM. No fragmento de SoMeSPC relevante para a integração, há apenas um conceito que não possui correspondente direto em ORMS: **Numeric Value** (Valor Numérico), especializado a partir de **Measured Value** (Valor Medido) para permitir o registro de valores de uma escala numérica. Os conceitos **Human Resource Role** (Papel Recurso Humano) e **Team Allocation** (Alocação de Equipe) equivalem a **Organizational Role** (Papel Organizacional) e **Team Membership** (Filiação à Equipe) em ORSM.

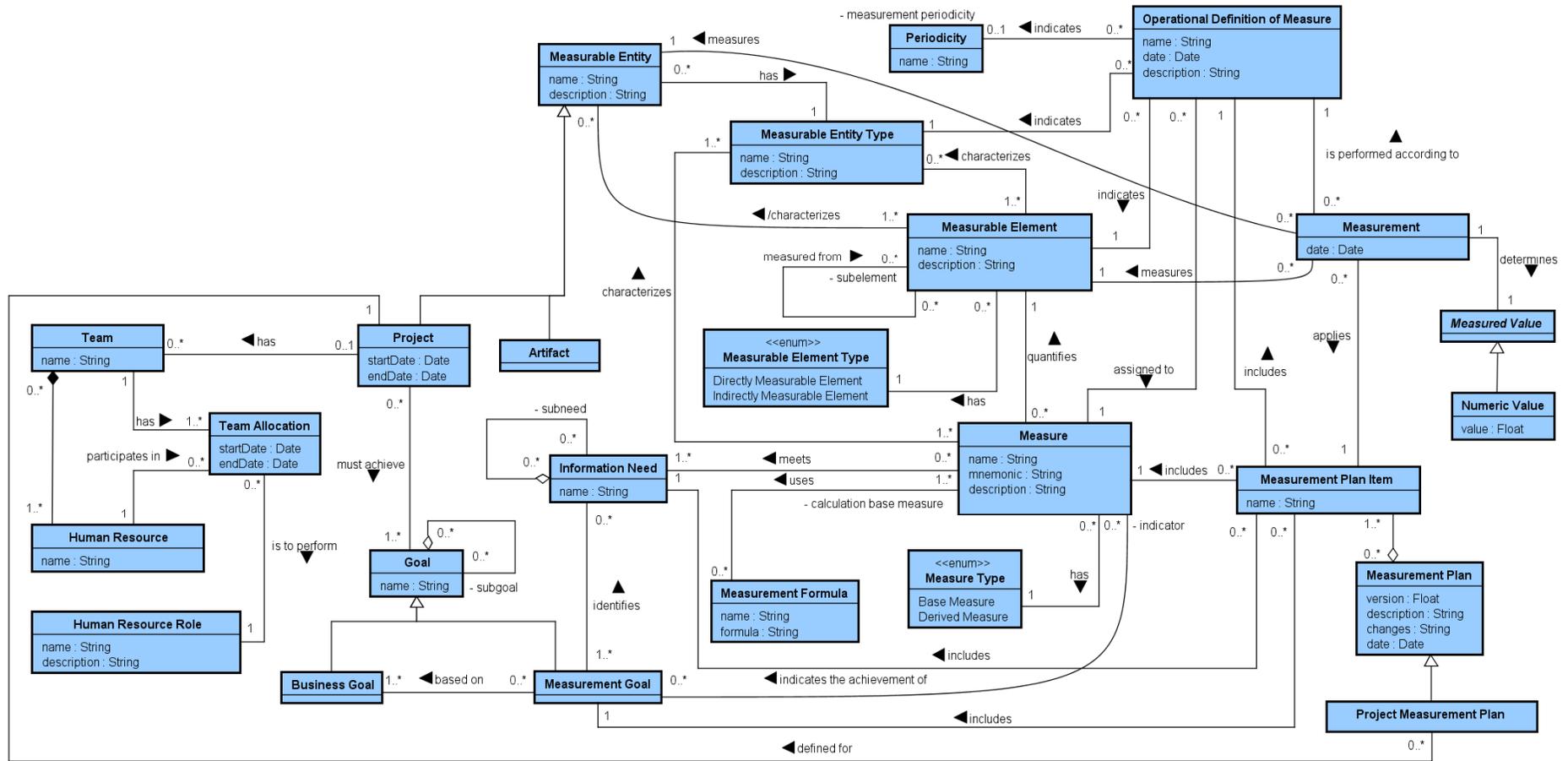


Figura 5.9 - Fragmento de SoMeSPC relevante à integração.

5.1.5.3 Mapeamentos Verticais

Os mapeamentos verticais visam atribuir semântica aos modelos de sistemas por meio de relacionamentos entre elementos dos modelos conceituais dos sistemas e elementos do modelo da ontologia utilizada (CALHAU, 2011). A Tabela 5.6 apresenta os mapeamentos verticais dos conceitos. Não foram representados na tabela os mapeamentos envolvendo conceitos idênticos que existem apenas em ORMS e SoMeSPC (por exemplo, *Goal* em SoMeSPC equivale a *Goal* em ORMS e nenhuma outra ferramenta trata esse conceito).

Tabela 5.6 - Mapeamentos Verticais de Conceitos.

ORMS	SoMeSPC	Taiga	SonarQube
Project	Project	Project	-
Sprint	-	Sprint	-
User Story	-	User Story	-
Sprint Activity	-	Task, quando status = new	-
Sprint Activity Occurrence	-	Task, quando status = inProgress ou closed	-
Human Resource	Human Resource	User	-
Organizational Team	Team, quando não associado a um Project	-	-
Project Team	Team, quando associado a um Project	Team	-
Team Member	Human Resource, quando associado a uma Team Allocation	User, quando associado a um Membership	-
Team Membership	Team Allocation	Membership	-
Organizational Role	Human Resource Role	Role	-
Human Resource Allocation	-	Associação de um User a uma Task, quando a Task tem status = new	-
Human Resource Participation	-	Associação de um User a uma Task, quando a Task tem status = inProgress ou closed	-
Measurable Entity	Measurable Entity	-	Resource
Measurable Entity Type	Measurable Entity Type	-	Qualifier
Measurable Element	Measurable Element	-	Domain
Directly Measurable Element	Measurable Element, quando Measurable Element.type = Directly Measurable Element	-	-
Indirectly Measurable Element	Measurable Element, when Measurable Element.type = Indirectly Measurable Element	-	-
Measure	Measure	-	Metric
Base Measure	Measure, quando Measure.type = Base Measure	-	-
Derived Measure	Measure, quando Measure.type = Derived Measure	-	-
Measurement	Measurement	-	Measure
Measured Value	Numeric Value	-	Measure.val

Conforme discutido no Capítulo 4, existem situações onde os mapeamentos não ocorrem diretamente de conceitos para classes, mas de conceitos para atributos, instâncias, e outros. Algumas dessas situações apareceram durante a integração das ferramentas SoMeSPC, Taiga e SonarQube, sendo descritas e tratadas a seguir.

Os conceitos *Measure* e *Measured Value* de ORMS não foram mapeados para nenhum conceito particular de Taiga. Em relação ao conceito *Measure*, é possível perceber que algumas das propriedades presentes em algumas classes do modelo conceitual do Taiga referem-se a medidas, ou seja, representam instâncias do conceito *Measure*. São elas: *totalPoints* na classe *User Story*, *totalUserStories*, *completedUserStories*, *totalTasks*, *completedTasks*, *iocaneDoses*, *totalPoints* e *completedPoints*, na classe *Sprint Stats Detail*, *totalPoints*, *closedPoints*, *totalMilestones* e *speed*, na classe *Project Stats Detail*. Na integração das ferramentas, essas propriedades dão origem a instâncias da classe *Measure*.

Em relação ao conceito *Measured Value*, pode-se notar que os valores atribuídos às propriedades citadas são os valores medidos para as medidas que essas propriedades representam. Assim, na integração, os valores atribuídos às propriedades nas instâncias da classe devem ser registrados como valores medidos para as respectivas medidas.

A Tabela 5.7 apresenta as medidas definidas a partir das propriedades citadas, para atender a necessidades de informação da organização.

Tabela 5.7 - Medidas identificadas a partir de propriedades de classes do Taiga.

#	Medida	Propriedade
1	Pontos de Estória Planejados para o Projeto	Project Stats Detail.totalPoints
2	Pontos de Estória Concluídos no Projeto	Project Stats Detail.closedPoints
3	Taxa de Conclusão de Pontos de Estória no Projeto	Medida 1 / Medida 2
4	Número de Sprints Planejadas para o Projeto	Project Stats Detail.totalMilestones
5	Número de Sprints Realizadas no Projeto	Quantidade de Sprint Stats Detail com Sprint Stats Detail.isClosed = true
6	Taxa de Conclusão de Sprints no Projeto	Medida 4 / Medida 5
7	Número de Estórias Planejadas para a Sprint	Sprint Stats Detail.totalUserStories
8	Número de Estórias Concluídas na Sprint	Sprint Stats Detail.completedUserStories
9	Taxa de Conclusão de Estórias na Sprint	Medida 7 / Medida 8
10	Pontos de Estória Planejados para a Sprint	Sprint Stats Detail.totalPoints
11	Pontos de Estória Concluídos na Sprint	Sprint Stats Detail.completedPoints
12	Taxa de Conclusão de Pontos de Estórias na Sprint	Medida 10 / Medida 11
13	Número de Tarefas Planejadas para a Sprint	Sprint Stats Detail.totalTasks
14	Número de Tarefas Concluídas na Sprint	Sprint Stats Detail.completedTasks
15	Taxa de Conclusão de Tarefas na Sprint	Medida 13 / Medida 14
16	Número de Estórias Concluídas para o Projeto	Soma de Sprint Stats Detail.completedUserStories
17	Média de Estórias Concluídas por Sprint do Projeto	Medida 16 / Medida 5

Tabela 5.7 - Medidas identificadas a partir de propriedades de classes do Taiga (cont.).

#	Medida	Propriedade
18	Velocidade da Equipe no Projeto	Project Stats Detail.speed
19	Número de Doses de Iocaine na Sprint	Sprint Stats Detail.iocaineDoses
20	Taxa de Doses de Iocaine na Sprint	Medida 19 / Medida 14
21	Número de Tarefas Atribuídas a Membro do Projeto	Quantidade de Tasks com Task.isClosed = false atribuídas a um dado User associado ao Membership que se refere ao Team do Projeto.
22	Número de Tarefas Concluídas pelo Membro do Projeto	Quantidade de Tasks com Task.isClosed = true atribuídas a um dado User associado ao Membership que se refere ao Team do Projeto
23	Taxa de Conclusão de Tarefas de Membro do Projeto	Medida 21 / Medida 22
24	Número de Pontos de Estória Atribuídos a Membro do Projeto	Quantidade de UserStory.totalPoints com UserStory.isClosed = false atribuídas a um dado User associado ao Membership que se refere ao Team do Projeto
25	Número de Pontos de Estória Concluídos pelo Membro do Projeto	Quantidade de UserStory.totalPoints com UserStory.isClosed = true atribuídas a um dado User associado ao Membership que se refere ao Team do Projeto
26	Taxa de Conclusão de Pontos de Estória de Membro do Projeto	Medida 24 / Medida 25
27	Taxa de Doses de Iocaine de Membro do Projeto	Quantidade de Tasks com Task.isIocaine = true atribuídas a um dado User associado ao Membership que se refere ao Team do Projeto
28	Taxa de Doses de Iocaine por Membro	Medida 27 / Medida 22

A Tabela 5.8 apresenta os mapeamentos verticais dos relacionamentos. Similar aos mapeamentos verticais de conceitos, não foram representados na tabela os mapeamentos envolvendo relações idênticas que existem apenas em ORMS e SoMeSPC (por exemplo, *Measurement Goal is based on Business Goal* em SoMeSPC equivale a *Measurement Goal is based on Business Goal* em ORMS e nenhuma outra ferramenta trata esse relacionamento).

Tabela 5.8 - Mapeamentos Verticais dos Relacionamentos entre Ontologia, SoMeSPC e Taiga.

ORMS			SoMeSPC			Taiga			Sonar Cube		
Conceito	Relação	Conceito	Conceito	Relação	Conceito	Conceito	Relação	Conceito	Conceito	Relação	Conceito
Sprint	implements →	User Story	-	-	-	Sprint	implements →	User Story	-	-	-
Sprint Activity	defined for →	Sprint	-	-	-	Task	/defined for →	Sprint	-	-	-
Sprint Activity	produces →	User Story				Task	defined for →	User Story			
Project	composed by →	Sprint	-	-	-	Project	composed by →	Sprint	-	-	-
Project	has →	Project Team	Project	has →	Team	Project	has →	Team	-	-	-
Team	has →	Team Membership	Team	has →	Team Allocation	Team	← refers to	Membership	-	-	-
Team Membership	allocates →	Team Member	Human Resource	← participates in	Team Allocation	Membership	refers to →	User	-	-	-
Team Membership	is to play →	Organizational Role	Team Allocation	is to perform →	Human Resource Role	Membership	refers to →	Role	-	-	-
Team Member	/member of →	Team	Human Resource	← composed by	Team	-	-	-	-	-	-
Human Resource Allocation	is to perform →	Project Activity				Task, quando status = new	assigned to →	User	-	-	-
	allocates →	Human Resource							-	-	-
Human Resource Participation	participation of →	Human Resource				Task, quando status = inProgress ou closed	assigned to →	User	-	-	-
	part of	Project Activity							-	-	-
Measurable Entity	instance of →	Measurable Entity Type	Measurable Entity	has →	Measurable Entity Type	-	-	-	Resource	has →	Qualifier
Measure	used to quantify →	Measurable Element	Measure	quantifies →	Measurable Element	-	-	-	Metric	has →	Domain
Base Measure	used to quantify →	Directly Measurable Element	Measure, quando Measure Type = Base Measure	quantifies →	Measurable Element, quando Measurable Element Type = Directly Measurable Element	-	-	-	-	-	-
Derived Measure	used to quantify →	Indirectly Measurable Element	Measure, quando Measure Type = Derived Measure	quantifies →	Measurable Element, quando Measurable Element Type = Indirectly Measurable Element	-	-	-	-	-	-
Measurement	measures →	Measurable Entity	Measurement	measures →	Measurable Entity	-	-	-	Measure	assign values to →	Resource
Measurement	applies →	Measure	Measurement	applies →	Measurement Plan Item	-	-	-	Measure	instance of →	Metric

5.1.5.4 Modelo Estrutural de Integração

Para a elaboração do modelo de integração, inicialmente tomou-se como base o modelo conceitual da ontologia selecionada. Em seguida, foram inseridos no modelo conceitos e relações presentes nos modelos estruturais das ferramentas integradas e não contemplados pela ontologia. Então, as cardinalidades de algumas relações estabelecidas na ontologia foram alteradas para se adequarem às relações correspondentes nas ferramentas integradas, que eram mais restritivas. Por fim, conceitos e relações necessários à integração e inexistentes tanto na ontologia quanto nas ferramentas foram inseridos.

A Figura 5.10 apresenta o modelo estrutural de integração. No modelo, estão em branco conceitos oriundos na ORMS e, seguindo-se as mesmas cores usadas nos modelos conceituais das ferramentas apresentados anteriormente, em marrom estão os conceitos originados no Taiga (***Sprint Stats Detail*** e ***Project Stats Detail***) e em azul o conceito com origem em SoMeSPC (***Numeric Value***). Conforme discutido no Capítulo 4, a integração pode necessitar de conceitos e relações inexistentes na ontologia e nas ferramentas, devendo ser incluídos no modelo integrador. É o caso do conceito ***CASE Tool***, que não tem origem na ontologia nem nas ferramentas integradas, mas é necessário para a identificação das ferramentas. Uma vez que a integração trata de medições realizadas automaticamente pelas ferramentas integradas, esse conceito e suas relações foram incluídos para possibilitar o registro da ferramenta responsável pela medição de uma medida (***CASE tool responsible for measurement***) e da ferramenta que executou uma medição (***CASE tool measurement executor***). Assim, torna-se possível identificar, por exemplo, que os dados para a medida *Número de Doses de Iocaine na Sprint* devem ser coletados pelo Taiga.

As relações *has* (entre ***Project*** e ***User Story***), *must achieve* (entre ***Project*** e ***Goal***) e a relação de especialização entre ***Numeric value*** e ***Measured Value*** (em azul no modelo) foram incluídas no modelo devido a relações presentes no Taiga e em SoMeSPC. As relações *indicates* (entre ***Operational Definition of Measure*** e ***CASE Tool***) e *performs* (entre ***CASE Tool*** e ***Measurement***) (também em azul no modelo) foram inseridas para atender necessidades da integração. Por fim, as relações *produces* (entre ***Sprint Activity*** e ***User Story***) e *implements* (entre ***Sprint*** e ***User Story***) (em vermelho no modelo) tiveram suas cardinalidades alteradas para ficarem consistentes com as relações estabelecidas no Taiga, uma vez que elas são mais restritivas no Taiga do que na ontologia.

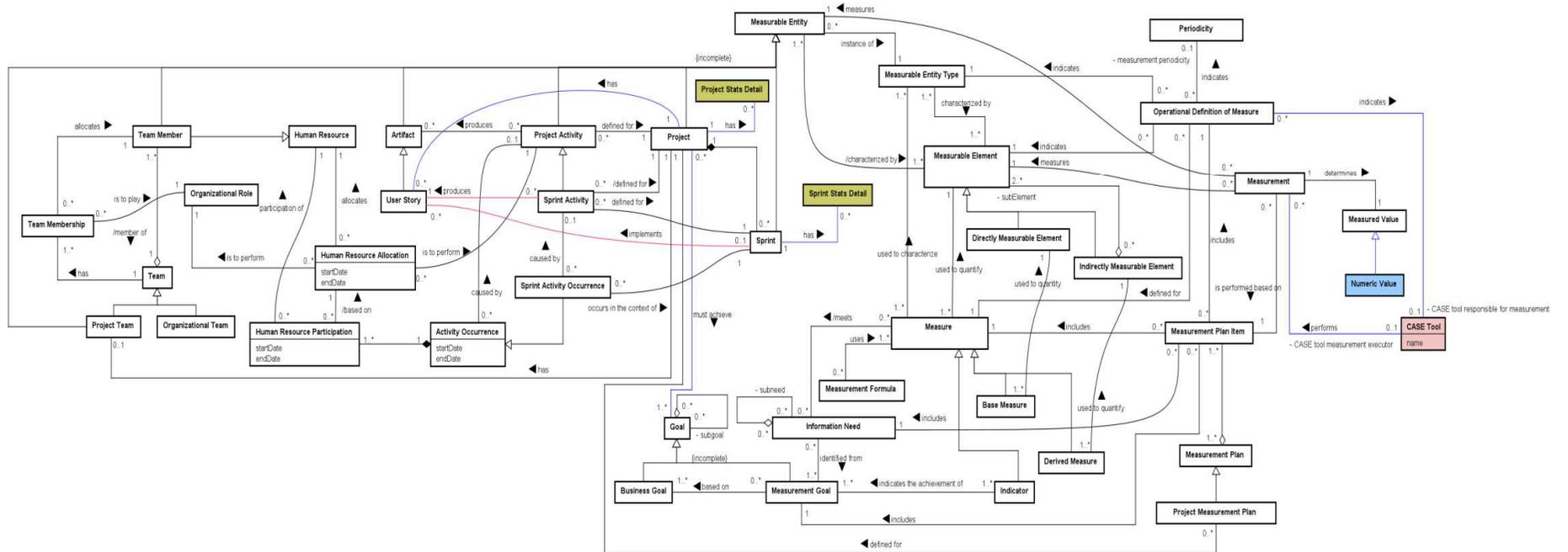


Figura 5.10 - Modelo Integrador Estrutural.

5.1.5.5 Mapeamentos Horizontais

Os mapeamentos horizontais visam tratar conceitos e relacionamentos ausentes na ontologia e introduzidos no modelo de integração a partir dos modelos estruturais das ferramentas. Assim, são tratados os mapeamentos existentes e que não foram considerados nos mapeamentos verticais. A Tabela 5.9 apresenta os mapeamentos horizontais dos conceitos e a Tabela 5.10 apresenta os mapeamentos horizontais dos relacionamentos.

Tabela 5.9 - Mapeamentos Horizontais de Conceitos.

Modelo de Integração	SoMeSPC	Taiga	SonarQube
Numeric Value	Numeric Value	-	-
Sprint Stats Detail	-	Sprint Stats Detail	-
Project Stats Detail	-	Project Stats Detail	-

Tabela 5.10 - Mapeamentos Horizontais de Relacionamentos.

Modelo de Integração			SoMeSPC			Taiga			SonarQube		
C	R	C	C	R	C	C	R	C	C	R	C
Numeric Value	subtype of →	Measured Value	-	-	-	Numeric Value	subtype of →	Measured Value	-	-	-
Project	must achieve →	Goal	Project	must achieve →	Goal	-	-	-	-	-	-
Project	has	Project Stats Detail	-	-	-	Project	has	Project Stats Detail	-	-	-
Sprint	has	Sprint Stats Detail	-	-	-	Sprint	has	Sprint Stats Detail	-	-	-

Legenda: C - Conceito; R - Relação

As atividades realizadas até o momento no contexto da análise da integração visam à obtenção do modelo estrutural da integração, cujo foco principal é a integração na camada de dados. Porém, em uma solução de integração mais completa, as camadas de serviço e processo também devem ser contempladas.

As ferramentas envolvidas na iniciativa de integração para o LEDS não têm modelos comportamentais sequenciais. Além disso, elas não tratam um mesmo processo. Embora Taiga e SonarQube forneçam dados relacionados à medição, eles não são sistemas concebidos para apoiar o processo de medição de software, então não possuem funcionalidades equivalentes a atividades do processo de medição. Assim, utilizou-se a abordagem discutida no Capítulo 4 para conduzir a integração semântica dessas ferramentas na camada de serviço e processo.

Inicialmente foram identificados os serviços disponíveis nas ferramentas e necessários para realizar a comunicação entre elas e foi feito o mapeamento destes com as atividades da OTMS, indicando-se os serviços que apoiam cada uma das atividades. O

resultado do mapeamento é apresentado na Tabela 5.11. Para a atividade *Analisar Medições*, as subatividades *Selecionar Objetivo de Medição para Análise* e *Reportar Resultados da Análise* não foram incluídas, pois não são apoiadas por serviços das ferramentas integradas.

Tabela 5.11 - Mapeamentos entre Serviços das Ferramentas e Atividades do Processo de Medição.

Atividade OTMS	Serviços	Descrição	Ferramenta
<i>Planejar Medição</i>			
Iniciar Planejamento da Medição	Cadastro de Plano de Medição de Projeto	Serviço que permite a criação de um plano de medição para um projeto.No contexto desta atividade trata do início da criação de um plano de medição, permitindo o registro de dados gerais como data de criação do plano.	SoMeSPC
Estabelecer Objetivos de Medição	Cadastro de Objetivo Estratégico	Serviço que permite cadastrar objetivos estratégicos. Objetivos estratégicos precisam ser cadastrados para serem inclusos em planos de medição.	SoMeSPC
	Cadastro de Objetivos de Medição	Serviço que permite cadastrar objetivos de medição, associando-os a objetivos estratégicos. Objetivos de medição precisam ser cadastrados para serem inclusos em planos de medição.	
	Criação de Plano de Medição de Projeto	Serviço que permite a criação de um Plano de Medição. No contexto desta atividade, trata da inclusão de objetivos estratégicos e de medição em um plano de medição.	
Identificar Necessidades de Informação	Cadastro de Necessidades de Informação	Serviço que permite cadastrar e associar necessidades de informação objetivos de medição e a medidas. Necessidades de informação precisam ser cadastradas para serem inclusas em planos de medição.	SoMeSPC
	Criação de Plano de Medição de Projeto	Serviço que permite a criação de um Plano de Medição. No contexto desta atividade, trata da inclusão de necessidades de informação em um plano de medição.	
Identificar Medidas	Cadastro de Medidas	Serviço que permite cadastrar medidas. As medidas precisam ser cadastradas para serem inclusas em planos de medição.	SoMeSPC
	Criação de Plano de Medição de Projeto	Serviço que permite a criação de um Plano de Medição. No contexto desta atividade, trata da inclusão de medidas em um plano de medição.	
Estabelecer Definição Operacional de Medidas	Cadastro de Definição Operacional de Medidas	Serviço que permite cadastrar definição operacional de medida. As definições operacionais de medidas precisam ser cadastradas para serem inclusas em planos de medição.	SoMeSPC
	Criação de Plano de Medição de Projeto	Serviço que permite a criação de um Plano de Medição. No contexto desta atividade, trata da inclusão de definição operacional de medida em um plano de medição.	
<i>Executar Medições</i>			
Selecionar Item do Plano de Medição para a Medição	Cadastro de Plano de Medição de Projeto	No contexto desta atividade, este serviço permite disponibilizar os itens de plano de medição para a seleção.	SoMeSPC

Tabela 5.11 - Mapeamentos entre Serviços das Ferramentas e Atividades do Processo de Medição (cont.).

Atividade OTMS	Serviços	Descrição	Ferramenta
Executar Medições			
Selecionar Entidade para a Medição	Cadastro de Plano de Medição de Projeto	No contexto desta atividade, este serviço permite disponibilizar o item de plano de medição selecionado, que indica o Tipo de Entidade Mensurável que pode ser selecionado nesta atividade.	SoMeSPC
	Cadastro de Projeto Cadastro de <i>Sprint</i> Cadastro de Estória Cadastro de Tarefa	No contexto desta atividade, estes serviços permitem disponibilizar as entidades mensuráveis do Taiga para seleção.	Taiga
	Cadastro de Código Fonte	No contexto desta atividade, este serviço permite disponibilizar as entidades mensuráveis do SonarQube para seleção.	SonarQube
Coletar Dados	Cadastro de Medição	Serviço usado para registrar medições.	SoMeSPC
	Cadastro de Projeto Cadastro de <i>Sprint</i> Cadastro de Estória Cadastro de Tarefa	No contexto desta atividade, estes serviços permitem obter valores medidos para a medida indicada no item de plano de medição selecionado, caso a ferramenta responsável pela medição da medida seja Taiga.	Taiga
	Cadastro de Código Fonte	No contexto desta atividade, este serviço permite obter valores medidos para a medida indicada no item de plano de medição selecionado, caso a ferramenta responsável pela medição da medida seja SonarQube.	SonarQube
Analisar Medições			
Selecionar Medidas para a Análise	Análise Tradicional de Medições	Serviço usado para realizar a análise de medições. No contexto desta atividade o serviço permite a seleção de medidas para análise.	SoMeSPC
Selecionar Dados para a Análise	Análise Tradicional de Medições	Serviço usado para realizar a análise de medições. No contexto desta atividade o serviço permite a seleção de medições para análise.	SoMeSPC
Analisar Dados	Análise Tradicional de Medições	Serviço usado para realizar a análise de medições. No contexto desta atividade o serviço permite registrar os resultados análises efetuadas.	SoMeSPC

Em seguida, para atribuir semântica ao processo de medição de software definido para o LEDS, foi feito o mapeamento de suas atividades com as atividades estabelecidas na OTMS. Uma vez que o processo de medição do LEDS foi definido com base na OTMS, o alinhamento entre os processos já era esperado. A Tabela 5.12 mostra o mapeamento entre as atividades dos processos de medição do LEDS e da OTMS.

Tabela 5.12 - Mapeamentos entre Atividades do Processo de Medição .

Atividade LEDS	Atividade OTMS
Planejar Medição	Planejar Medição
Estabelecer Objetivos de Medição	Estabelecer Objetivos de Medição
Identificar Necessidades de Informação	Identificar Necessidades de Informação
Identificar Medidas	Identificar Medidas
Estabelecer Definição Operacional de Medidas	Estabelecer Definição Operacional de Medidas
Realizar Estimativas do Projeto	Executar Medições
Realizar Estimativas de <i>Sprints</i> do Projeto	Executar Medições

Tabela 5.12 - Mapeamentos entre Atividades do Processo de Medição (cont.).

Atividade LEDES	Atividade OTMS
Registrar Conclusão de Tarefa	Executar Medições
Medir Código Fonte	Executar Medições
Registrar Homologação de Estória	Executar Medições
Registrar Encerramento de <i>Sprint</i>	Executar Medições
Atualizar Estimativas	Executar Medições
Realizar Encerramento do Projeto	Executar Medições
Analisar Medições da <i>Sprint</i>	Analisar Medições
Analisar Medições do Projeto	Analisar Medições

Analisando-se os mapeamentos realizados, percebe-se que os serviços mapeados para as atividades da OTMS podem, por transitividade, ser mapeados para as atividades do processo de medição do LEDES, indicando que eles apoiam esse processo. Assim, uma vez realizada a integração das ferramentas considerando-se os serviços identificados, obtém-se apoio integrado ao processo de medição.

5.1.6 Projeto, Implementação e Implantação da Integração

Para viabilizar a integração na camada de serviços, foi necessário garantir que as ferramentas fossem capazes de expor os serviços necessários para apoiar o processo de medição. Taiga e SonarQube já possuíam API de serviços com os serviços necessários, enquanto SoMeSPC não. Dessa forma, após a seleção de SoMeSPC como uma das ferramentas a serem integradas, ela foi evoluída para permitir a disponibilização dos serviços e também para realização de melhorias na ferramenta. As principais alterações realizadas dizem respeito à plataforma e ao desenvolvimento de uma API (*Application Programming Interface*) para disponibilizar os serviços necessários para a integração. As alterações foram tratadas no contexto deste trabalho e em (CASTRO, 2015).

Em relação à plataforma, SoMeSPC deixou de estar apoiada no portal corporativo *Liferay*⁸ e ser baseada em tecnologia de *portlets* e passou a utilizar o servidor de aplicação web *Apache Tomcat*⁹ e tecnologias de aplicações web em Java. Também foi necessário fazer a portabilidade do sistema gerenciador de banco de dados originalmente usado em SoMeSPC (*PostgreSQL*¹⁰) para *MySQL*¹¹, que é utilizado no LEDES.

A API foi desenvolvida seguindo os princípios do padrão arquitetural REST (*Representational State Transfer*) (FIELDING, 2000) e inclui serviços relacionados à criação e

⁸ <http://www.liferay.com>

⁹ <http://tomcat.apache.org>

¹⁰ <http://www.postgresql.org>

¹¹ <http://www.mysql.com>

obtenção de entidades mensuráveis, projetos, recursos humanos, equipes, medidas, medições, plano de medição, entre outros. Foi utilizado o *framework Jersey*¹², desenvolvido para permitir a criação de serviços web baseados na arquitetura REST. Para acesso aos serviços, o cliente REST deve usar um dos verbos do protocolo HTTP (*HyperText Transfer Protocol*) para a URI (*Uniform Resource Identification*) do serviço, sendo mais comuns os verbos GET, para obter dados, e POST, para enviar dados. Assim como as ferramentas Taiga e SonarQube, a notação utilizada para a comunicação com os serviços de SoMeSPC é a JSON (*JavaScript Object Notation*).

Com as três ferramentas disponibilizando APIs de serviços, a integração na camada de serviços tornou-se viável. Decidiu-se, então, pela criação de um mediador para orquestrar os serviços e fornecer funcionalidades para apoiar as atividades do processo de medição do LEDS. A Figura 5.11 ilustra a interação do mediador com as APIs de serviços das ferramentas integradas.

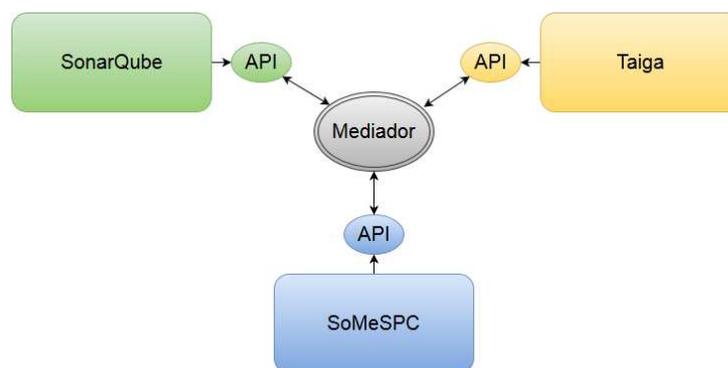


Figura 5.11 - Interação do Mediador com as APIs de serviços.

O desenvolvimento do mediador se deu através de uma extensão de SoMeSPC. Foram adicionados pacotes de classes em SoMeSPC para permitir que o mediador seja executado sempre que SoMeSPC esteja ativa.

Para a construção da interface gráfica do mediador, foram utilizados os *frameworks Google AngularJS*¹³ e *Twitter Bootstrap*¹⁴. A combinação desses *frameworks* possibilita a criação de interfaces web ricas e responsivas, facilitando o acesso por dispositivos móveis como *tablets* e *smartphones*. *Google AngularJS* é um *framework* baseado no padrão MVC (*Model-View-Controller*) para aplicações web. Esse *framework* é executado pelo navegador web e é usado pelo mediador para a comunicação entre o cliente e as APIs de serviços das ferramentas.

¹² <http://jersey.java.net>

¹³ <http://angularjs.org>

¹⁴ <http://getbootstrap.com>

Twitter Bootstrap, por sua vez, é um *framework* para estilização de páginas web. Ele possui uma série de folhas de estilo (CSS - *Cascading Style Sheets*) incluídas que permitem a utilização de temas e customização de paletas de cores, além de garantir a responsividade das páginas web para acesso por dispositivos móveis. No mediador, esse *framework* foi utilizado para estilização das interfaces gráficas.

As funcionalidades providas pelo mediador são: (i) criação de Plano de Medição para projetos; (ii) agendamento de medições automáticas; (iii) apresentação de dados para análise das medições. As funcionalidades (i) e (iii) já existiam em SoMeSPC, porém foram evoluídas e encapsuladas no mediador para serem guiadas por objetivos e, também, para terem seu uso facilitado. A funcionalidade (ii) não existia em nenhuma das ferramentas e foi desenvolvida no mediador. As funcionalidades providas pelo mediador são disponibilizadas a partir do menu de SoMeSPC, por meio das opções *Novo Plano de Medição Integrado*, que apoia a criação de planos de medição, e *Painel de Controle*, que permite controlar o agendamento de medições e realizar a análise de medições. A Figura 5.12 apresenta o menu de SoMeSPC, incluindo as opções para acessar as funcionalidades do mediador.

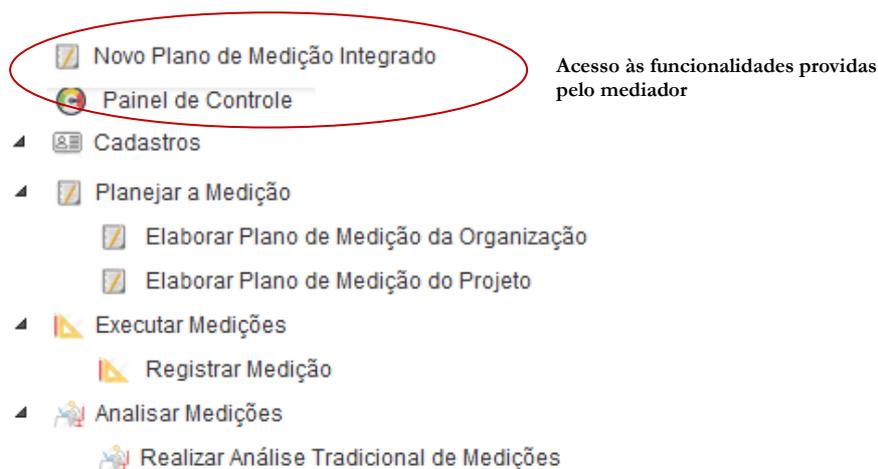


Figura 5.12 - Menu de SoMeSPC incluindo itens para acesso às funcionalidades do mediador.

Para a funcionalidade **Criação do Plano de Medição**, o mediador disponibiliza um *wizard* para guiar o usuário. Os objetivos, necessidades de informação e medidas que foram identificados durante a atividade *Identificar Objetivos, Necessidades de Informação e Medidas Relevantes para a Organização* de OBA-MSI são disponibilizados nesta funcionalidade para que o usuário selecione os itens que devem constar no Plano de Medição do Projeto a ser gerado, devendo indicar a periodicidade da medição para as medidas selecionadas, ou seja, o intervalo de tempo em que dados devem ser coletados automaticamente a partir do Taiga ou do SonarQube. De acordo com a ferramenta fornecedora de dados para as medidas

selecionadas para o Plano de Medição, o mediador aciona os serviços das APIs do Taiga ou do SonarQube.

No desenvolvimento do *wizard* foi utilizada uma extensão do *framework AngularJS*, o *Angular Wizard*¹⁵, para permitir a adaptação dinâmica dos próximos passos a serem realizados de acordo com os dados selecionados pelo usuário. As figuras 5.13 e 5.14 apresentam algumas das telas do *wizard* para criação de Plano de Medição. Na Figura 5.13 é apresentada a tela onde são selecionados os objetivos, necessidades de informação e medidas para o Plano de Medição e é informada a periodicidade de coleta das medidas. Na Figura 5.14 é apresentada a tela com o resumo do Plano de Medição criado, mostrando as medidas incluídas no plano e os projetos para os quais foram definidos planos de medição.

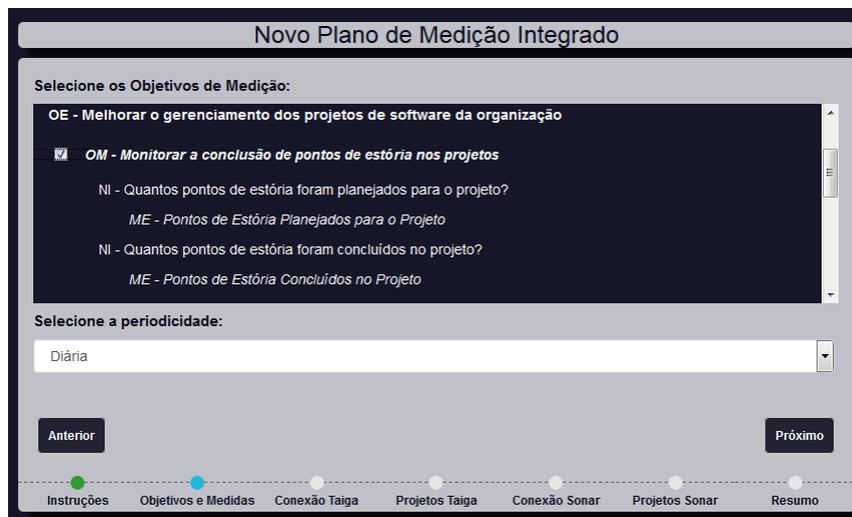


Figura 5.13 - Tela para seleção de objetivos e medidas para o Plano de Medição.

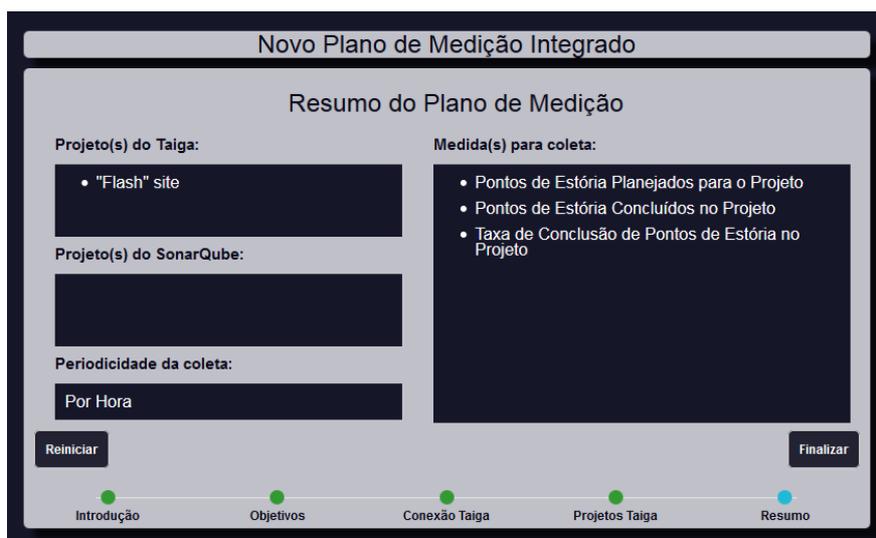


Figura 5.14 - Tela com o resumo do Plano de Medição criado.

¹⁵ <http://mgonzo.github.io/angular-wizard>

Uma vez definido o Plano de Medição para um projeto, são feitos agendamentos para a realização de medições automáticas das medidas presentes no Plano de Medição, considerando a periodicidade para elas determinada. Os agendamentos são realizados como *jobs* de medição e foram implementados com auxílio do *framework Quartz Scheduler*¹⁶. Foram criadas duas classes representando os *jobs* para cada ferramenta: *TaigaMedicaoJob* e *SonarQubeMedicaoJob*. Cada uma das classes implementa o método *execute()* do *framework*. Esse método é acionado sempre que a periodicidade de medição ocorre. Por exemplo, se a periodicidade da medição é diária, a cada 24 horas o método *execute* do respectivo *job* é acionado. Ao final da criação de um Plano de Medição, para cada medida incluída no plano, uma instância do *job* da ferramenta que provê os dados para a medida é criado. Dessa forma, o método *execute* dos *jobs* de medição é o responsável pelas medições e possui o conhecimento de qual serviço chamar, como obter os valores da medição e como armazená-los.

Para a funcionalidade **Agendamento de Medições**, o mediador disponibiliza para o usuário um painel com informações sobre os *jobs* de medição agendados, sendo possível ao usuário acompanhar a execução dos *jobs*, acioná-los, iniciá-los, pausá-los ou excluí-los. Na Figura 5.15 é apresentada a tela com o painel de agendamento de medições.



Figura 5.15 - Tela com o painel de agendamento de medições.

Para a funcionalidade **Análise de Medições**, o mediador permite que o usuário identifique o objetivo de medição a ser monitorado, a medida a ser analisada e selecione os dados a serem analisados informando a entidade mensurável, o período ao qual os dados se

¹⁶ <http://quartz-scheduler.org>

referem ou a quantidade de dados a ser considerada. A partir dessas informações são apresentados gráficos com os valores medidos selecionados para que o usuário realize a análise das medições. A Figura 5.16 apresentada a tela com a representação de dados para análise de medições.



Figura 5.16 - Análise de Medições do Mediador.

Após a implementação e testes da integração, o conjunto integrado foi implantado no LEDS. As ferramentas integradas estão acessíveis a partir de três pontos distintos: UFES, IFES e Internet. SoMeSPC e o mediador são hospedados em um servidor do NEMO, na UFES, SonarQube é hospedada em um servidor do LEDS e Taiga é acessada a partir de um serviço online gratuito disponível pela Internet. A Figura 5.17 ilustra a distribuição das ferramentas envolvidas na iniciativa de integração implantada no LEDS.

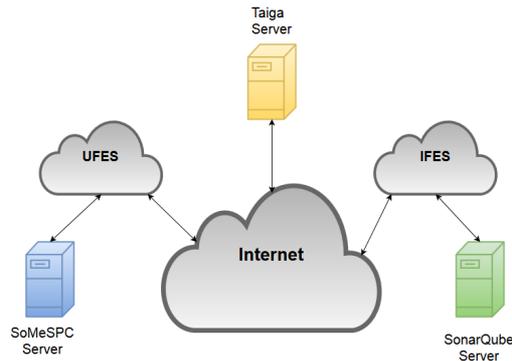


Figura 5.17 - Infraestrutura de implantação do conjunto integrado.

5.2 Avaliação da Integração de Ferramentas pelos Usuários do LEDS

Tratados no âmbito da Engenharia de Software Experimental, os estudos experimentais têm sido usados para encontrar indícios e aprimorar a utilização de técnicas no contexto de projetos de software (TRAVASSOS; GUROV; AMARAL, 2002). Neste trabalho foi realizado um *survey* buscando-se encontrar indícios que permitam avaliar e aprimorar a abordagem proposta (OBA-MSI) e o conjunto integrado implantado no LEDS.

Um *survey* é uma pesquisa de opinião que, frequentemente, é realizada em retrospecto. Ela pode ser realizada, por exemplo, após a utilização de uma ferramenta ou técnica (PFLEEGER, 1994). Dessa forma, um *survey* permite capturar um ‘retrato instantâneo’ de uma situação. Os principais meios de aplicação de um *survey* são questionários e entrevistas, que são realizados com uma amostra representativa da população. Os questionários são respondidos pela amostra selecionada e os resultados são coletados e analisados para derivar uma conclusão. O resultado é, então, generalizado para a população representada (MAFRA; TRAVASSOS, 2006).

5.2.1 Planejamento do Estudo

O **objetivo** do estudo realizado foi avaliar se o conjunto integrado de ferramentas implantado no LEDS é capaz de apoiar o processo de medição e se o uso do conjunto integrado agregou valor à organização. Utilizando-se a abordagem GQM (BASILI; CALDIERA; ROMBACH, 1994) este objetivo é assim formalizado:

Analisar o conjunto integrado de ferramentas implantado no LEDS, produzido utilizando-se OBA-MSI

Com o propósito de avaliar as funcionalidades providas por meio do mediador para apoiar o processo de medição de software

Com respeito à capacidade de apoiar o processo de medição de software e agregar valor à organização

Sob o ponto de vista de membros da equipe dos projetos

No contexto de projetos de software

Para analisar os resultados, foram considerados os seguintes indicadores:

- a) Adequação das Funcionalidades
- b) Utilidade das Funcionalidades
- c) Benefícios providos em comparação ao uso das ferramentas isoladamente

A **instrumentação** utilizada na condução do estudo consiste de três formulários: (i) um termo de consentimento para a realização do estudo, que visa resguardar os direitos dos participantes quanto ao estudo e seus resultados; (ii) um formulário para caracterizar o perfil dos participantes, que visa obter informações sobre o conhecimento e experiência dos participantes em medição de software; e (iii) um questionário que permite que os participantes registrem sua percepção após o uso do conjunto integrado das ferramentas para apoiar o processo de medição de software. Esses formulários são apresentados no Apêndice A desta dissertação.

O **procedimento** de condução do estudo consistiu de três etapas. Na primeira etapa, o pesquisador apresentou uma palestra sobre medição de software para a equipe do LEDES para alinhar o conhecimento e a terminologia sobre o tema. Em seguida, o pesquisador apresentou o conjunto integrado de ferramentas com destaque para as funcionalidades providas pelo mediador. O conjunto integrado foi, então, disponibilizado para uso. Após duas semanas de uso do conjunto integrado sem a intervenção do pesquisador, foi realizada a segunda etapa, que consistiu na aplicação de um questionário eletrônico, elaborado com auxílio do *Google Forms*¹⁷, para registrar a percepção dos participantes sobre o uso do conjunto integrado. O questionário foi enviado por e-mail para os participantes e as respostas foram coletadas. Na terceira etapa foi realizada uma entrevista com o coordenador do LEDES, a fim de se obter informações mais detalhadas sobre sua percepção quanto ao uso do conjunto integrado. Decidiu-se por realizar a entrevista com o coordenador pois ele é o maior interessado na utilização dos dados para apoiar a tomada de decisão e é o membro mais experiente do LEDES.

O **questionário** inclui questões relacionadas ao processo de medição definido, à adequação e utilidade das funcionalidades providas pelo mediador e aos benefícios

¹⁷ <http://www.google.com/forms/about/>

providos pelo uso do conjunto integrado quando comparado ao uso isolado das ferramentas. O questionário contém questões objetivas e discursivas. Para as objetivas é solicitado ao participante que inclua uma justificativa. O questionário também inclui questões discursivas que visam à avaliação geral e melhoria do conjunto integrado de ferramentas. Algumas questões do questionário são apresentadas na Figura 5.18. O questionário pode ser visto na íntegra no Apêndice A.

	<p>NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias) UFES (Universidade Federal do Espírito Santo)</p> <p>Estudo Experimental: <i>Uso do conjunto integrado de ferramentas SoMeSPC, Taiga e SonarQube para apoiar atividades do Processo de Medição de Software</i></p>	
---	---	---

Explicitar os objetivos, necessidades de informação e medidas melhorou a percepção das razões para se realizar a medição de software? *

Sim

Parcialmente

Não

Justifique sua resposta para a questão [Explicitar os objetivos, necessidades de informação e medidas melhorou a percepção das razões para se realizar a medição de software?].

A funcionalidade "Novo Plano de Medição Integrado" é adequada para apoiar o planejamento da medição? *

Sim

Parcialmente

Não

Justifique sua resposta para a questão [A funcionalidade "Novo Plano de Medição Integrado" é adequada para apoiar o planejamento da medição?].

Comparando-se a utilização das ferramentas SoMeSPC, Taiga e SonarQube individualmente para apoiar o processo de medição de software com o conjunto integrado, em sua opinião o conjunto integrado: *

Muito mais benefícios do que o uso das ferramentas individualmente

Mais benefícios do que o uso das ferramentas individualmente

Os mesmos benefícios do que o uso das ferramentas individualmente

Menos benefícios do que o uso das ferramentas individualmente

Muito menos benefícios do que o uso das ferramentas individualmente

Destaque os principais benefícios obtidos a partir da utilização do conjunto integrado.

Figura 5.18 - Fragmento do questionário de avaliação.

Os **participantes** do estudo foram os membros do LEDES, totalizando 7 participantes, dos quais um é o coordenador do LEDES e 6 são membros de equipes de desenvolvimento. Dentre os participantes, apenas um declarou ter conhecimento teórico *médio* sobre medição. Os demais declararam ter *baixo* conhecimento. Em relação a experiência prática, dois participantes declararam ter *baixa* experiência prática e cinco declararam não ter experiência, sendo que o uso do conjunto integrado foi seu primeiro contato com a execução de práticas de medição de software.

Embora a maioria dos participantes do estudo não tenha experiência prática em medição de software e tenha conhecimento teórico limitado, eles não foram excluídos da pesquisa, uma vez que todos são membros do LEDES e utilizaram o conjunto integrado até o momento desta pesquisa. De fato, era de se esperar que a maioria dos participantes não apresentasse conhecimento teórico e prático em medição de software, uma vez que a integração de ferramentas foi realizada para apoiar a introdução do processo de medição no LEDES. Dessa forma, a avaliação realizada deve ser considerada uma avaliação sob o ponto de vista de iniciantes em medição de software.

5.2.2 Resultados

Nesta seção são apresentados os resultados obtidos a partir dos questionários respondidos pelos participantes do estudo. Algumas análises quantitativas dos resultados são apresentadas.

5.2.2.1 Percepção acerca do Processo de Medição

As duas primeiras perguntas feitas para os participantes do estudo, dizem respeito ao processo de medição de software definido. O objetivo das perguntas foi capturar a percepção dos participantes quanto ao entendimento do processo de medição e às razões para realizar medição de software.

Em relação ao processo de medição de software definido, a maioria dos participantes (seis) disse ter compreendido o processo de medição definido para a organização e apenas um participante respondeu que não compreendeu o processo. Esse participante justificou sua resposta informando não ter muito conhecimento em medição de software, o que pode ter dificultado seu entendimento. As respostas são ilustradas pela Figura 5.19.

Você compreendeu o Processo de Medição definido para a organização?

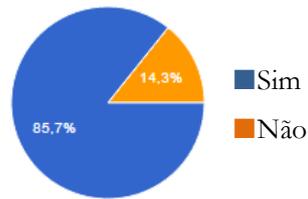


Figura 5.19 - Compreensão do processo de medição.

Com relação às razões para realizar a medição de software através da explicitação dos objetivos, necessidades de informação e medidas, cinco participantes responderam que houve melhoria na percepção dos motivos de se realizar medição de software e dois participantes disseram que houve melhoria parcial. Esses dois participantes declararam ter baixo conhecimento teórico e nenhuma experiência prática em medição de software. A Figura 5.20 ilustra as respostas.

Explicitar os objetivos, necessidades de informação e medidas melhorou a percepção das razões para se realizar a medição de software?

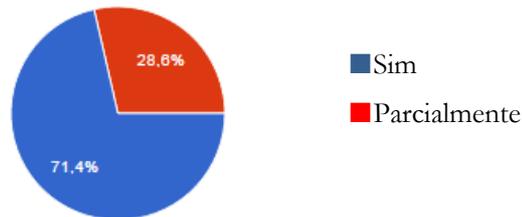
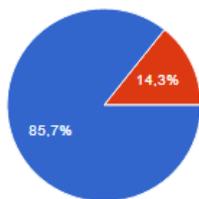


Figura 5.20 - Melhoria da percepção das razões para realizar a medição de software.

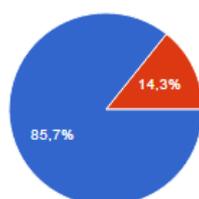
5.2.2.2 Adequação e Utilidade das Funcionalidades

A respeito da adequação e utilidade das funcionalidades, notou-se homogeneidade nas respostas. Para todas as funcionalidades, seis participantes responderam que são adequadas e úteis e apenas um participante respondeu que as funcionalidades são parcialmente adequadas e úteis. Esse participante não apresentou justificativa para suas respostas. A Figura 5.21 apresenta as questões e seus percentuais de resposta.

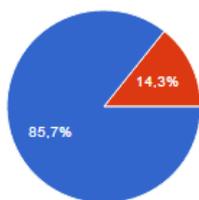
A funcionalidade "Novo Plano de Medição Integrado" é adequada para apoiar o planejamento da medição?



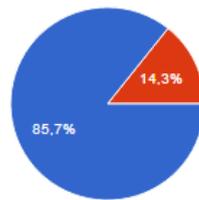
A funcionalidade "Novo Plano de Medição Integrado" é útil no apoio ao processo de Medição de Software?



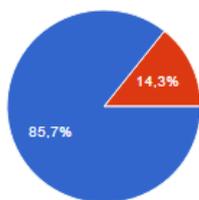
O uso da funcionalidade "Agendamento de Medições" é adequada para apoiar a execução de medições?



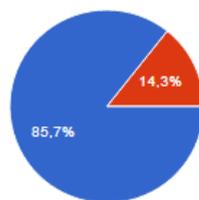
O uso da funcionalidade "Agendamento de Medições" é útil no apoio ao processo de Medição de Software?



O uso da funcionalidade "Análise de Medições" é adequada para apoiar a análise de medições?



O uso da funcionalidade "Análise de Medições" é útil no apoio ao processo de Medição de Software?



■ Sim ■ Parcialmente

Figura 5.21 - Adequação e utilidade das funcionalidades.

5.2.2.3 Benefícios Obtidos

Com relação aos benefícios obtidos pelo uso do conjunto integrado quando comparado ao uso individual das ferramentas integradas, 4 participantes responderam haver muito mais benefícios e 3 participantes responderam haver mais benefícios, conforme ilustra a Figura 5.22.

Comparando-se a utilização das ferramentas SoMeSPC, Taiga e SonarQube individualmente para apoiar o processo de medição de software com o conjunto integrado, em sua opinião o conjunto integrado:



Figura 5.22 - Benefícios obtidos com o uso do conjunto integrado comparado ao uso individual das ferramentas.

As justificativas apresentadas estão relacionadas à precisão dos dados, economia no tempo de trabalho e simplificação do acompanhamento dos projetos através das informações disponibilizadas pela integração.

5.2.2.4 Dificuldades de Uso e Sugestões de Melhoria

Com relação às dificuldades para utilizar o conjunto integrado, nenhum participante relatou ter dificuldade e um dos participantes ressaltou que isso se deve ao fato de a ferramenta mediadora ser autoexplicativa.

Quanto às sugestões de melhoria, sugeriu-se a criação de um *dashboard* com gráficos definidos especificamente para o coordenador/gestor. Essa sugestão poderá ser tratada em trabalhos futuros. Também foi sugerido ter predefinidos os dados de autenticação para a criação de planos de medição, evitando a digitação dessas informações para cada plano de medição a ser criado ou modificado.

5.2.3 Entrevista com o Coordenador do LEDS

Na terceira etapa do estudo, foi feita uma entrevista com o coordenador do LEDS com a finalidade de capturar a percepção sobre o uso do conjunto integrado por parte da pessoa que ocupa o papel principal na tomada de decisão e, dessa forma, é a pessoa mais diretamente interessada nos dados fornecidos. Além disso, considerando-se que a maioria dos participantes do estudo é iniciante em medição de software, a entrevista visou obter algumas informações adicionais com o membro mais experiente do LEDS. Após o coordenador do LEDS responder o questionário de avaliação, foram feitas as seguintes perguntas: (a) Quais as suas percepções com relação ao processo de medição de software definido para o LEDS? (b) O que você achou da abordagem adotada para iniciar a medição de software no LEDS? (c) De que maneira a implantação do conjunto integrado de ferramentas auxilia (ou pode auxiliar) nas tomadas de decisões no LEDS? (d) Você já conseguiu perceber benefícios no uso do conjunto integrado? (e) Alguma crítica ou sugestão para melhorar o trabalho realizado no LEDS? Um resumo das respostas fornecidas pelo coordenador é apresentado a seguir.

Segundo o coordenador, o processo de medição definido e a abordagem adotada para implementar o processo de medição no LEDS são adequados. Ele relatou que o conjunto integrado favorece a análise dos dados, que é feita diariamente, uma vez que concentra informações advindas das ferramentas integradas em um único ponto e evita que

ele precise acessar as ferramentas individualmente. Além disso, novas medidas que não estão diretamente presentes nas demais ferramentas são fornecidas pelo conjunto integrado. O coordenador destacou que apesar de a implantação do conjunto integrado ser recente, já é possível analisar dados dos projetos e realizar comparações no desempenho das equipes de projetos, permitindo a identificação de problemas e o auxílio na melhoria do desempenho da equipe com desempenho inferior. Por fim, o coordenador sugeriu que o pesquisador tenha uma participação ativa e presencial no LEDES para auxiliar na capacitação em medição de software dos membros do LEDES e potencializar a obtenção dos benefícios da utilização do conjunto integrado. Segundo o coordenador, essa participação constante do pesquisador no LEDES auxiliaria na institucionalização da medição de software no LEDES e na melhoria contínua do processo.

5.2.4 Discussões

Conforme estabelecido no planejamento do estudo, seu objetivo é avaliar se o conjunto integrado de ferramentas é capaz de apoiar o processo de medição de software e agregar valor à organização. Para isso, são considerados três indicadores: (a) Adequação das Funcionalidades; (b) Utilidade das Funcionalidades; e (c) Benefícios providos em comparação ao uso das ferramentas isoladamente.

Com relação às funcionalidades, considerou-se que elas são adequadas e úteis para apoiar ao processo de medição de software, uma vez que a maioria dos participantes considerou as funcionalidades adequadas e úteis para apoiar as atividades de medição e nenhum participante teve percepção contrária a essa.

Quanto aos benefícios providos pelo conjunto integrado em comparação ao uso das ferramentas isoladamente, todos os participantes afirmaram haver mais benefícios em se utilizar o conjunto integrado do que as ferramentas isoladamente. Em uma entrevista, o coordenador do LEDES reforçou essa percepção afirmando que, apesar de a implantação do conjunto integrado ser recente, já é possível ter uma visão mais abrangente dos projetos em andamento, identificar problemas e propor soluções.

Considerando-se a os resultados, pode-se concluir que, no contexto da avaliação realizada no LEDES, as funcionalidades são adequadas e úteis no apoio ao processo de medição de software e o uso do conjunto integrado é mais vantajoso que o uso das ferramentas isoladamente e, dessa forma, agrega valor à organização.

5.2.5 Ameaças à Validade

Todo estudo apresenta ameaças à validade de seus resultados. As ameaças devem ser tratadas na medida do possível e devem ser consideradas juntamente com os resultados obtidos no estudo. As ameaças relacionadas a este estudo foram divididas em categorias e são apresentadas a seguir.

Validade Interna: é definida como a capacidade de um novo estudo repetir o comportamento do estudo atual com os mesmos participantes e objetos com que ele foi realizado (BARROS; WERNER; TRAVASSOS, 2002). A principal ameaça à validade interna é a comunicação e o compartilhamento de informações entre participantes do estudo. Para tratar essa ameaça, o questionário eletrônico foi enviado para o e-mail pessoal do participante, de forma que ele pudesse responder individualmente as questões no momento que considerasse mais adequado. Isso minimiza a ameaça da comunicação, uma vez que não é obrigatório que os participantes estejam fisicamente próximos durante a realização do estudo, entretanto, não elimina a possibilidade de que haja comunicação entre eles.

Validade Externa: essa ameaça está relacionada à capacidade de repetir o mesmo comportamento com grupos diferentes daquele que participou do estudo (BARROS; WERNER; TRAVASSOS, 2002). Nesse contexto, uma ameaça detectada foi o fato de os participantes serem, com exceção do coordenador, alunos de graduação de cursos relacionados a computação e iniciantes em medição de software (com baixo conhecimento teórico e pouca ou nenhuma experiência prática). Considerando-se que os participantes do estudo deveriam ser membros do LEDES que usaram o conjunto integrado, não foi possível contornar essa ameaça.

Validade de Constructo: refere-se à relação entre os instrumentos e os participantes do estudo e a teoria que está sendo provada (BARROS; WERNER; TRAVASSOS, 2002). Foi identificada a ameaça de o participante dar respostas que não refletem a realidade devido a expectativas pessoais e por imaginar que ele próprio está sendo submetido à avaliação. Para minimizar a possibilidade de ocorrer essa ameaça, os participantes foram informados de que o experimento não representa qualquer tipo de avaliação pessoal, mas sim avaliação do conjunto integrado de ferramentas. Também foi assegurado o anonimato das respostas.

Validade de Conclusão: mede a relação entre os tratamentos e os resultados e afere a capacidade do estudo em gerar conclusões (BARROS; WERNER; TRAVASSOS, 2002). Foram identificadas as seguintes ameaças: (i) a pequena quantidade de participantes;

(ii) o fato de a maioria dos participantes ser iniciante em medição de software; e (iii) o curto período de utilização do conjunto integrado antes da condução da avaliação. Essas ameaças limitam a possibilidade de generalização dos resultados obtidos e do comportamento observado. Por isso, os resultados do estudo não podem ser generalizados e são considerados apenas resultados preliminares e indícios, não sendo conclusivos.

5.3 Considerações Finais do Capítulo

Este capítulo apresentou a aplicação de OBA-MSI para realizar a integração de ferramentas no LEDS, tendo sido apresentados os principais resultados da execução de cada atividade de OBA-MSI. O uso de OBA-MSI em uma iniciativa de integração real mostrou que a abordagem é viável. No entanto, é preciso considerar que a abordagem foi utilizada por seu proponente, o que implica em viés na avaliação de sua utilização. Assim, para uma avaliação menos influenciada, OBA-MSI deve ser utilizada por outras pessoas para conduzir iniciativas de integração de ferramentas para apoiar medição de software.

Conforme discutido anteriormente, devido ao fato de as ferramentas envolvidas na integração não seguirem um modelo comportamental sequencial bem definido e duas delas não terem sido concebidas para apoiar o processo de medição, adotou-se a abordagem sugerida em OBA-MSI, que difere um pouco da sugerida em OBA-SI, para realizar a integração semântica nas camadas de serviços e processo. Ao invés de o mapeamento semântico entre funcionalidades das ferramentas integradas e atividades da ontologia indicar equivalência semântica, ele indica que as funcionalidades apoiam as atividades.

Embora as ferramentas integradas não tenham um modelo comportamental sequencial, o mediador desenvolvido para prover funcionalidades da integração apresenta esse tipo de comportamento. Dessa forma, é possível fazer um mapeamento entre as funcionalidades providas pelo mediador e as atividades da OTMS. A Tabela 5.13 apresenta este mapeamento, mostrando que a integração realizada é capaz de apoiar todas as atividades do processo de medição. Embora as subatividades do processo de medição definido em OTMS não estejam representadas na Tabela 5.13, vale ressaltar que a subatividade *Selecionar Objetivo de Medição para Análise*, que não era apoiada por nenhuma das ferramentas integradas, passou a ser apoiada pela funcionalidade *Analisar Medições* do mediador.

Tabela 5.13 – Mapeamento entre funcionalidades do mediador e atividades da OTMS.

Atividade de OTMS	Funcionalidades do Mediador
Planejar Medição	Criação do Plano de Medição
Executar Medições	Agendamento de Medições
Analisar Medições	Análise de Medições

Por fim, cabe destacar que a integração realizada não só permitiu a integração das ferramentas, mas também potencializou a possibilidade de uso de SoMeSPC. Conforme comentado anteriormente neste capítulo, devido à necessidade de entrada de dados manualmente em SoMeSPC quando usada isoladamente, seu uso no dia a dia de uma organização pode se tornar inviável. Na integração, essa limitação foi tratada, uma vez que a estrutura provida pela SoMeSPC é utilizada pelo mediador e os dados são obtidos automaticamente a partir das demais ferramentas.

Após um período inicial de utilização do conjunto integrado no LEDS, foi realizada uma avaliação dos resultados segundo a perspectiva dos usuários. Com essa avaliação observou-se que as funcionalidades providas pelo mediador foram consideradas adequadas e úteis no apoio ao processo de medição de software definido para o LEDS. Os participantes consideraram o uso do conjunto integrado mais vantajoso do que o uso das ferramentas de maneira isolada, agregando maior valor para a organização. No entanto, é importante destacar que, devido às limitações apontadas na Seção 5.2.5, os resultados dessa avaliação não podem ser generalizados.

Capítulo 6

Conclusão

Neste capítulo são feitas as considerações finais deste trabalho (Seção 6.1), sendo apresentadas suas principais contribuições (Seção 6.2) e perspectivas de trabalhos futuros para continuidade e aprimoramento da pesquisa (Seção 6.3).

6.1 Considerações Finais

Medição de software é uma prática importante que provê informações necessárias para a tomada de decisão nos âmbitos organizacional e dos projetos. Para que a medição seja efetuada adequadamente e possa prover os benefícios esperados, ela deve ser orientada a objetivos (FLORAC; CARLETON, 1997 e PARK, 1997). Idealmente, o processo de medição de software de uma organização deve estar definido e ser institucionalizado e a etapa de planejamento da medição deve ser realizada com base nos objetivos da organização e dos projetos (SEI, 2010).

A eficiência da medição de software está fortemente relacionada ao apoio de ferramentas associadas a todo o processo de medição (DUMKE; EBERT, 2007). Mesmo que as ferramentas utilizadas em uma organização não sejam específicas para apoiar o processo de medição de software, muitas vezes elas fornecem dados relacionados à medição, que são úteis para a tomada de decisão. Entretanto, essas ferramentas precisam ser integradas para que possam apoiar o processo de medição como um todo.

A integração de aplicações não é uma tarefa simples (THEMISTOCLEOUS; IRANI; LOVE, 2004). A heterogeneidade dos sistemas é o maior problema enfrentado durante uma iniciativa de integração, pois cada aplicação possui modelos de dados e de processo (comportamento) implementados de forma independente, definindo à sua própria maneira os conceitos relevantes (IZZA, 2009). Isso leva a diversos conflitos, sendo que os principais estão relacionados à semântica dos dados e serviços compartilhados entre as ferramentas (POKRAEV, 2009).

Ontologias são consideradas um meio importante para alcançar a integração semântica, pois fornecem especificações formais de conceituações compartilhadas (NARDI; FALBO; ALMEIDA, 2013). Elas podem ser usadas para estabelecer um entendimento comum de um universo de discurso, servindo como uma interlíngua para a comunicação entre sistemas e evitando problemas semânticos (CALHAU, 2011).

No contexto desse trabalho, o estado da arte de iniciativas de integração entre ferramentas de medição foi investigado através de um mapeamento sistemático (FONSECA; BARCELLOS; FALBO, 2015) e uma revisão sistemática da literatura. Com isso, algumas lacunas foram identificadas, entre elas: (i) a falta de uma visão holística do processo de software; (ii) o apoio limitado ao planejamento da medição, (iii) a não preocupação com aspectos semânticos; e (iv) a não cobertura da camada de processo.

Considerando essas lacunas, este trabalho teve como objetivo definir uma abordagem para a integração de ferramentas visando apoiar o processo de medição de software. Para isso, foram utilizadas a Ontologia de Tarefa de Medição de Software (OTMS) (BARCELLOS; FALBO, 2013) e a Ontologia de Referência de Medição de Software (ORMS) (BARCELLOS, 2009) em uma especialização de OBA-SI visando apoiar a integração de ferramentas de apoio à medição de software. OBA-MSI (*Ontology-Based Approach for Measurement Systems Integration*) é uma especialização de OBA-SI centrada na fase de levantamento de requisitos. OBA-MSI define um conjunto de atividades visando: (i) definir um processo de medição de software alinhado ao processo estabelecido em OTMS; (ii) garantir que o processo de medição de software seja orientado a objetivos e apoiado por ferramentas integradas; (iii) tratar a semântica das ferramentas integradas; e (iv) permitir a integração nas camadas de dados, serviços e processo.

O objetivo geral deste trabalho foi detalhado em três objetivos específicos, sendo que todos foram alcançados neste trabalho. A Tabela 6.1 apresenta os objetivos específicos do trabalho e o principal produto que serve como evidência do alcance de cada objetivo.

Tabela 6.1 - Objetivos específicos do trabalho.

Objetivos	Produto
Analisar o estado da arte da integração de ferramentas no contexto de medição de software	Revisão Terciária, Mapeamento Sistemático da Literatura e Revisão Sistemática da Literatura (vide Capítulo 3)
Especializar OBA-SI para tratar aspectos relacionados à integração de ferramentas de apoio à medição de software	OBA-MSI (vide Capítulo 4)
Aplicar especialização de OBA-SI para realizar a integração de ferramentas visando apoiar o processo de medição de software em uma organização	Aplicação da abordagem no LEDES (vide Capítulo 5)

Ao se comparar a abordagem proposta neste trabalho e a iniciativa de integração conduzida no LEDES com as iniciativas descritas nos estudos analisados durante a revisão sistemática da literatura, percebe-se que existem algumas diferenças importantes. Primeiramente, OBA-MSI é uma abordagem para apoiar a condução de iniciativas de integração de ferramentas para apoiar o processo de medição de software. Nos estudos investigados não se identificou nenhuma abordagem para apoiar a integração de

ferramentas, tendo sido identificadas apenas iniciativas de integração propriamente ditas. Entretanto, é preciso destacar que a revisão sistemática da literatura investigou iniciativas e, dessa forma, o foco da investigação não foram abordagens. Então, embora as iniciativas encontradas não tenham seguido uma abordagem sistemática, não se pode afirmar que elas não existam.

Outra diferença, conforme já destacado neste trabalho, está relacionada à cobertura da integração na camada de processo. Nenhuma das iniciativas analisadas considera a integração na camada de processo. Ao se aplicar OBA-MSI, uma vez que o ponto de partida é o processo de medição de software a ser apoiado, a integração na camada de processo é estimulada, sendo tratada no âmbito da análise da integração. Dessa forma, a iniciativa conduzida no LEDS lida com a camada de processo, enquanto as demais iniciativas não.

Outra diferença também já destacada neste trabalho, está relacionada à preocupação com a semântica durante a integração. Apenas um dos trabalhos analisados, *SOFAS* (GHEZZI; GALL, 2011), se preocupa com aspectos relativos à semântica. Apesar de tanto *SOFAS* quanto OBA-MSI aplicarem ontologias durante a integração, é possível identificar algumas diferenças entre as ontologias usadas nas duas abordagens: (i) *SOFAS* aplica ontologias relacionadas aos domínios das ferramentas integradas (gerência de problemas, gerência de configuração e codificação), enquanto OBA-MSI aplica ontologias relacionadas ao domínio de medição de software; (ii) as ontologias utilizadas em *SOFAS* são consideradas ontologias leves (*lightweight ontologies*), cuja preocupação não é dispor de uma representação conceitual adequada, mas sim garantir propriedades computacionais desejáveis (GUIZZARDI, 2007), enquanto as ontologias utilizadas por OBA-MSI são ontologias de referência, cujo objetivo é fazer a melhor descrição possível de um domínio na realidade, considerando um certo nível de granularidade e certo ponto de vista (GUIZZARDI, 2005); (iii) as ontologias de medição utilizadas por OBA-MSI são fundamentadas na ontologia de fundamentação UFO, garantindo um melhor tratamento da ambiguidade de conceitos fundamentais através da explicitação dos compromissos ontológicos adjacentes às ontologias (GUIZZARDI, 2006), enquanto as ontologias utilizadas em *SOFAS* não são baseadas em ontologias de fundamentação.

Também destaca-se a diferença na cobertura do apoio ao processo de medição. OBA-MSI prevê apoio a todo o processo de medição, enquanto outras iniciativas (*Tool Support for SM, DSS, SM in a CI Environment, SOFAS, Dione, QualitySpy, 3C, DePress*) apoiam apenas as atividades relacionadas a coleta e análise de dados.

Com relação a aspectos operacionais da integração, o projeto e implementação da iniciativa de integração realizada no LEDES aplicando-se OBA-MSI (vide Seção 5.1) assemelha-se à proposta *MetriFlame* (KOMI-SIRVIO; PARVIAINEN; RONKAINEN, 2001), uma vez que ambas fazem uso de agendadores para executar medições. Porém, como *MetriFlame* não especifica detalhes sobre o projeto nem sobre a implementação dos agendadores, não é possível fazer uma comparação mais criteriosa entre as duas abordagens.

Analisando-se a iniciativa de integração realizada no LEDES considerando-se o *framework* de Izza (2009), nota-se que a iniciativa possui um escopo intraorganizacional, uma vez que apenas o LEDES foi envolvido. Ela trata o ponto de vista do programador, pois é uma iniciativa funcional e implementada; trata o ponto de vista do projetista através dos modelos e diagramas elaborados; e trata o ponto de vista do usuário com as funcionalidades providas pelo mediador. Com relação à dimensão de camadas, a iniciativa aborda a camada de dados através dos mapeamentos dos conceitos e relações da ontologia com os das ferramentas no modelo integrador; a camada de serviços é tratada pela identificação e utilização dos serviços providos pelas ferramentas; e a camada de processo é tratada pelo mapeamento dos serviços fornecidos pelas ferramentas para as atividades do processo de medição da OTMS. Sobre o nível da integração, a iniciativa cuida do nível semântico por meio das ontologias de medição de software e da abordagem OBA-MSI.

A Tabela 6.2 apresenta um resumo comparativo entre as principais características da iniciativa de integração conduzida no LEDES e as iniciativas encontradas na revisão sistemática da literatura apresentada no Capítulo 3.

Tabela 6.2 – Resumo comparativo entre a iniciativas de integração.

Iniciativa	Atividades de Medição Apoiadas	Processos Medidos	Camadas de Integração	Nível de Integração	Como a Integração ocorre
TAME	Planejamento da Medição Execução da Medição Análise da Medição	Codificação	Serviço	Sintático	Componentes
Tool Support for SM	Execução da Medição Análise da Medição	Codificação	Dados e Serviço	Sintático	Programas utilitários convertem dados para a interoperabilidade.
GQM Tool	Planejamento da Medição Execução da Medição Análise da Medição	Codificação	Dados	Sintático	Acesso direto a banco de dados
MetriFlame	Planejamento da Medição Execução da Medição Análise da Medição	Depende dos dados disponíveis	Dados	Sintático	Agendador e ferramenta central
DSS	Execução da Medição Análise da Medição	Gerência de Clientes	Dados	Sintático	Data warehouse e aplicação web

Tabela 6.2 – Resumo comparativo entre a iniciativas de integração (cont.).

Iniciativa	Atividades de Medição Apoiadas	Processos Medidos	Camadas de Integração	Nível de Integração	Como a Integração ocorre
SM in CI Environment	Execução da Medição Análise da Medição	Codificação	Dados	Sintático	Motor de Integração Contínua
SOFAS	Execução da Medição Análise da Medição	Codificação	Serviço	Semântico	Orquestração de Serviços por <i>Workflow</i>
Dione	Execução da Medição Análise da Medição	Codificação	Serviço	Sintático	<i>Smart clients</i>
QualitySpy	Execução da Medição Análise da Medição	Codificação	Dados	Sintático	Conectores
3C	Execução da Medição Análise da Medição	Codificação e Testes	Serviço	Sintático	Motor de Integração Contínua
ASSIST	Planejamento da Medição Execução da Medição Análise da Medição	Gerência de Projetos e Codificação	Dados	Sintático	Acesso direto a banco de dados
DePress	Execução da Medição Análise da Medição	Gerência de Configuração e Codificação	Dados	Sintático	<i>Framework</i> de integração
Iniciativa de integração no LEDES, usando OBA-MSI	Planejamento da Medição Execução da Medição Análise da Medição	Gerência de Projetos e Codificação	Dados, Serviço e Processo	Semântico	Aplicação mediadora e serviços

Entre as limitações deste trabalho, pode ser destacada a avaliação da abordagem. Mesmo sendo realizada em uma organização no contexto de projetos reais, a integração foi conduzida pelo proponente de OBA-MSI, o que influencia os resultados. Além disso, embora funcione como uma empresa de desenvolvimento de software, o LEDES é uma organização cujos participantes são alunos em fase de preparação profissional e iniciantes em medição de software, o que também influencia nos resultados da avaliação da utilização do conjunto integrado. Além disso, o tempo de utilização do conjunto integrado antes da avaliação foi relativamente curto. Dessa forma, os resultados da avaliação não podem ser considerados conclusivos, mas apenas indícios de que a abordagem é viável e os resultados da integração agregam valor à organização.

6.2 Contribuições

As principais contribuições desta dissertação são:

- (i) A abordagem OBA-MSI, que define atividades necessárias para o apoio ao processo de medição de software das organizações por meio da integração de ferramentas de medição com base em ontologias de domínio e de tarefa de medição.

- (ii) Mapeamento Sistemático da Literatura, que descreve o panorama das iniciativas de integração de aplicações para apoiar medição de software. Os principais resultados do mapeamento foram registrados em (FONSECA; BARCELLOS; FALBO, 2015).
- (iii) Revisão Sistemática da Literatura, que fornece uma investigação mais profunda das propostas encontradas no mapeamento sistemático.
- (iv) Conjunto integrado de ferramentas implantado no LEDS, que apoia o processo de medição na organização e, além disso, pode ser reutilizado em outras organizações.

6.3 Trabalhos Futuros

Considerando a pesquisa aqui apresentada, há algumas perspectivas de trabalhos futuros. No âmbito da *pesquisa* podem-se destacar:

- (i) Atualizar a investigação da literatura realizada, a fim de verificar o surgimento de novas iniciativas de integração de ferramentas para apoiar medição de software e suas características.
- (ii) Realizar uma investigação formal da literatura buscando-se abordagens para integração de ferramentas para apoiar medição de software.
- (iii) Adaptar OBA-MSI para se tornar uma abordagem mais genérica que possa ser utilizada no contexto de domínios de aplicação de medição em geral, utilizando-se como base a Ontologia de Núcleo de Medição e a Ontologia de Tarefa de Medição definidas em (FRAUCHES, 2014).
- (iv) Explorar o uso do conjunto integrado definido para o LEDS para apoiar o ensino em Medição de Software.

Em relação à *avaliação de OBA-MSI e do conjunto integrado de ferramentas implantado no LEDS*, pode-se:

- (i) Realizar uma nova iniciativa de integração de ferramentas, a qual deve ser conduzida por outra pessoa, para permitir avaliar a viabilidade de uso de OBA-MSI por outras pessoas, que não seu propositor.
- (ii) Realizar uma nova avaliação sobre a utilização do conjunto integrado de ferramentas no LEDS, após um período maior de utilização.
- (iii) Realizar novas avaliações sobre a utilização do conjunto integrado de ferramentas, incluindo novos participantes e novas organizações,

preferencialmente com características diferentes das do LEDES e seus membros.

Em relação ao *conjunto integrado de ferramentas implantado no LEDES*, pode-se:

- (i) Evoluir o conjunto integrado de ferramentas para permitir a integração de dados de medições retroativas, uma vez que as organizações normalmente possuem dados históricos em seus sistemas de apoio mesmo antes da institucionalização da medição de software.
- (ii) Evoluir o conjunto integrado de ferramentas para a utilização de um barramento de serviços, visando facilitar a adição de novas ferramentas ao conjunto integrado.
- (iii) Incluir outras ferramentas no conjunto integrado.
- (iv) Evoluir o conjunto integrado de ferramentas para exibir as medições através de outros tipos de gráficos (atualmente, suporta apenas gráficos de linhas).

Referências Bibliográficas

- BARCELLOS, M. P. **Uma Estratégia para Medição de Software e Avaliação de Bases de Medidas para Controle Estatístico de Processos de Software em Organizações de Alta Maturidade**, 2009. Tese de Doutorado, COPPE/UFRJ - Universidade Federal do Rio de Janeiro.
- BARCELLOS, M. P.; FALBO, R. A. **Using a Foundational Ontology for Reengineering a Software Enterprise Ontology**. Lecture Notes in Computer Science. v. 5833 LNCS, p.179–188, 2009.
- BARCELLOS, M. P.; FALBO, R. A. **A software measurement task ontology**. Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13. **Anais...** . p.311, 2013. New York, New York, USA: ACM Press.
- BARCELLOS, M. P.; FALBO, R. A.; ROCHA, A. R. **Establishing a well-founded conceptualization about software measurement in high maturity levels**. Proceedings - 7th International Conference on the Quality of Information and Communications Technology, QUATIC 2010, p. 467–472, 2010.
- BARROS, M. DE O.; WERNER, C. M. L.; TRAVASSOS, G. H. **Um Estudo Experimental sobre a Utilização de Modelagem e Simulação no Apoio à Gerência de Projetos de Software**. XVI Simpósio Brasileiro de Engenharia de Software, p. 191–206, 2002.
- BASIL, V. R.; CALDIERA, G.; ROMBACH, H. D. **Goal Question Metric Approach**. Encyclopedia of Software Engineering, 1994. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- BASIL, V. R.; ROMBACH, H. D. **The TAME project: towards improvement-oriented software environments**. IEEE Transactions on Software Engineering, v. 14, n. 6, p. 758–773, 1988.
- BRINGUENTE, A. C. O. **Reengenharia de Uma Ontologia de Processo de Software e seu Uso para a Integração de Ferramentas de Apoio ao Planejamento de Projetos**, 2011. (Dissertação de Mestrado). Universidade Federal do Espírito Santo.
- BRINGUENTE, A. C. O.; FALBO, R. A.; GUIZZARDI, G. **Using a foundational ontology for reengineering a Software Process Ontology**. Proceedings of the XXVI Brazilian Symposium on Data Base, 2011.
- CAGLAYAN, B.; MISIRLI, A. T.; CALIKLI, G.; BENER, A.; AYTAC, T.; TURHAN, B. **Dione: An Integrated Measurement and Defect Prediction Solution**. Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software

- Engineering - FSE '12. **Anais...** . p.1–4, 2012. New York, New York, USA: ACM Press.
- CALHAU, R. F. **Uma abordagem baseada em ontologias para integração semântica de sistemas**, 2011. (Dissertação de Mestrado). Universidade Federal do Espírito Santo.
- CALHAU, R. F.; FALBO, R. A. **An Ontology-Based Approach for Semantic Integration**. 2010 14th IEEE International Enterprise Distributed Object Computing Conference. **Anais...** . p.111–120, 2010. IEEE.
- CARD, D. N.; DOMZALSKI, K.; DAVIES, G. **Making statistics part of decision making in an engineering organization**. IEEE Software, v. 25, n. 3, p. 37–47, 2008.
- CASTRO, H. **Evolução de uma Ferramenta de Medição para Apoiar a Medição de Software no Contexto de Métodos Ágeis**, 2015. (trabalho em andamento). Universidade Federal do Espírito Santo.
- CHULANI, S.; RAY, B.; SANTHANAM, P.; LESZKOWICZ, R. **Metrics for managing customer view of software quality**. Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717). **Anais...** . p.189–198, 2003. IEEE Comput. Soc.
- DUMKE, R.; EBERT, C. **Software Measurement: Establish - Extract - Evaluate - Execute**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**, 2000.
- FLORAC, W. A.; CARLETON, A. D. **Measuring the software process: statistical process control for software process improvement**. Boston, USA: Addison Wesley, 1997.
- FLORAC, W. A.; CARLETON, A. D.; BARNARD, J. R. **Statistical process control: analyzing a space shuttle onboard software process**. IEEE Software, v. 17, n. 4, p. 97–106, 2000.
- FONSECA, V. S.; BARCELLOS, M. P.; FALBO, R. A. **Integration of Software Measurement Supporting Tools: A Mapping Study**. Twenty-Seventh International Conference on Software Engineering and Knowledge Engineering (SEKE 2015). **Anais...** . p.516–521, 2015. Pittsburg: Knowledge Systems Institute.
- FRAUCHES, V. G. V. **Uma Abordagem Baseada em Ontologias para Obtenção de Indicadores a partir de Dados Abertos**, 2014. Universidade Federal do Espírito Santo.
- GHEZZI, G.; GALL, H. C. **SOFAS: A Lightweight Architecture for Software**

- Analysis as a Service.** 2011 Ninth Working IEEE/IFIP Conference on Software Architecture, p. 93–102, 2011.
- GUARINO, N. **Formal Ontology and Information Systems.** Proceedings of FOIS'98. *Anais...* . p.3–15, 1998.
- GUARINO, N.; STAAB, S.; STUDER, R. **Handbook on Ontologies.** Second ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models,** 2005.
- GUIZZARDI, G. **The Role of Foundational Ontologies for Conceptual Modeling and Domain Ontology Representation.** 2006 7th International Baltic Conference on Databases and Information Systems, v. 5, p. 17–25, 2006.
- GUIZZARDI, G. **On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models.** Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV. IOS Press, Amsterdam., v. 155, p. 18–39, 2007.
- IEEE. **IEEE Std 1061:1998 - IEEE Standard for a Software Quality Metrics Methodology.** 2005.
- ISO/IEC. **IEEE Standard Adoption of ISO/IEC 15939:2007—Systems and Software Engineering—Measurement Process.** 2007.
- ISO/IEC. **IEEE Standard Adoption of ISO/IEC 12207:2008 — Systems and software engineering — Software life cycle processes.** 2008.
- IZZA, S. **Integration of industrial information systems: from syntactic to semantic integration approaches.** Enterprise Information Systems, v. 3, n. 1, p. 1–57, 2009.
- JANUS, A.; DUMKE, R.; SCHMIETENDORF, A.; JAGER, J. **The 3C approach for Agile Quality Assurance.** 3rd International Workshop on Emerging Trends in Software Metrics (WETSoM). *Anais...* . p.9–13, 2012. IEEE.
- JURECZKO, M.; MAGOTT, J. **QualitySpy: a framework for monitoring software development processes.** Journal of Theoretical and Applied Computer Science, v. 6, n. 1, p. 35–45, 2012.
- KESER, B.; IYIDOGAN, T.; OZKAN, B. **ASSIST: An Integrated Measurement Tool.** Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement. *Anais...* . p.237–242, 2013. IEEE.
- KITCHENHAM, B.; BUDGEN, D.; BRERETON, O. P. **Using mapping studies as the basis for further research - A participant-observer case study.** Information and Software Technology, v. 53, n. 6, p. 638–651, 2011. Elsevier B.V.

- KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. *Engineering*, v. 2, p. 1051, 2007.
- KOMI-SIRVIO, S.; PARVIAINEN, P.; RONKAINEN, J. **Measurement automation: methodological background and practical solutions a multiple case study**. *Proceedings Seventh International Software Metrics Symposium. Anais...* . p.306–316, 2001. IEEE Comput. Soc.
- LAVAZZA, L. **Providing automated support for the GQM measurement process**. *IEEE Software*, v. 17, n. 3, p. 56–62, 2000.
- LEDS. **Laboratório de Extensão e Desenvolvimento de Sistemas**. Disponível em: <<http://leds.sr.ifes.edu.br/>>. Acesso em: 25/11/2015.
- DE LUCIA, A.; POMPELLA, E.; STEFANUCCI, S. **Assessing the maintenance processes of a software organization: an empirical analysis of a large industrial project**. *Journal of Systems and Software*, v. 65, n. 2, p. 87–103, 2003.
- MADEYSKI, L.; MAJCHRZAK, M. **Software Measurement and Defect Prediction with DePress Extensible Framework**. *Foundations of Computing and Decision Sciences*, v. 39, n. 4, p. 249–270, 2014.
- MAFRA, S. N.; TRAVASSOS, G. H. **Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software**. Rio de Janeiro, 2006.
- MARETTO, C. X. **Uma Arquitetura de Referência para Medição de Software**, 2013. (Dissertação de Mestrado). Universidade Federal do Espírito Santo.
- MARTINS, A. F.; FALBO, R. A. **Models for representing task ontologies**. *CEUR Workshop Proceedings*, v. 427, 2008.
- MCGARRY, J.; CARD, D.; JONES, C.; LAYMAN, B.; CLARK, E.; DEAN, J.; HALL, F. **Practical Software Measurement: Objective information for decision makers**. Boston, USA: Addison Wesley, 2002.
- MOREIRA, G. DE S. P.; MELLADO, R. P.; MONTINI, D. Á.; DIAS, L. A. V.; MARQUES DA CUNHA, A. **Software Product Measurement and Analysis in a Continuous Integration Environment**. 2010 Seventh International Conference on Information Technology: New Generations. *Anais...* . p.1177–1182, 2010. IEEE.
- NARDI, J. C.; FALBO, R. A.; ALMEIDA, J. P. A. **A Panorama of the Semantic EAI Initiatives and the Adoption of Ontologies by these Initiatives**. In: M. Sinderen; P. Oude Luttighuis; E. Folmer; S. Bosems (Orgs.); IWEI 2013, LNBIP 144, Lecture Notes in Business Information Processing. v. 144, p.198–211, 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.

- PARK, R. E. **Practical Software Measurement: Measuring for Process Management and Improvement** *Practical Software Measurement: Measuring for Process Management and Improvement*. , 1997.
- PARK, R. E.; GOETHERT, W. B.; FLORAC, W. A. **Goal-Driven Software Measurement—A Guidebook**. Software Engineering Institute Carnegie Mellon University, Pittsburgh, PA, v. 15213, n. August, 1996.
- PFLEEGER, S. L. **Design and Analysis in Software Engineering**. *Acm Sigsoft*, v. 19, n. 4, p. 16–20, 1994.
- PHAM, A. **Scrum em Ação: Gerenciamento e Desenvolvimento Ágil de Projetos de Software**. Paperback, 2011.
- POKRAEV, S. **Model-Driven Semantic Integration of Service-Oriented Applications**, 2009. University of Twente.
- SEI. **CMMI® for Development, Version 1.3**. Pittsburg, 2010.
- SOFTEX. **MPS.BR - Melhoria de Processo do Software Brasileiro - Guia Geral MPS de Software**. 2012.
- SOLINGEN, R.; BERGHOUT, E. **The Goal/Question/Metric Method: a practical guide for quality improvement of software development**. *A Practical Guide for Quality Improvement of Software Development*. New York, McCraw-Hill Publishers, p. 216, 1999.
- TARHAN, A.; DEMIRÖRS, O. **Investigating Suitability of Software Process and Metrics for Statistical Process Control**. . p.88–99, 2006.
- THEMISTOCLEOUS, M.; IRANI, Z.; LOVE, P. E. D. **Evaluating the integration of supply chain information systems: A case study**. *European Journal of Operational Research*, , n. 159, p. 393–405, 2004.
- TIAN, J.; TROSTER, J.; PALMA, J. **Tool support for software measurement, analysis and improvement**. *Journal of Systems and Software*, v. 39, n. 2, p. 165–178, 1997.
- TRAVASSOS, G.; GUROV, D.; AMARAL, E. **Introdução à Engenharia de Software Experimental**. Relatório Técnico ES-590/02 Programa de Engenharia de Sistemas e Computação COPPEUFRJ, p. 52, 2002.
- USCHOLD, M.; JASPER, R. **A Framework for Understanding and Classifying Ontology Applications**. *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*. *Anais...* . p.1–12, 1999.
- VERNADAT, F. B. **Interoperable enterprise systems: Principles, concepts, and methods**. *Annual Reviews in Control*, v. 31, n. 1, p. 137–145, 2007.

- WACHE, H.; VOGELE, T.; VISSER, U.; STUCKENSCHMIDT, H.; SCHUSTER, G.; NEUMANN, H.; HUBNER, S. **Ontology-Based Information Integration: A Survey of Existing Approaches**. International Joint Conference on Artificial Intelligence; Workshop: Ontologies and Information Sharing, p. 108–117, 2001.
- WEGNER, P. **Interoperability**. ACM Computing Surveys, v. 28, n. 1, p. 285–287, 1996.

Apêndice A

Formulários Utilizados no Estudo Experimental

Este apêndice apresenta os formulários utilizados durante o estudo para avaliação do conjunto integrado de ferramentas SoMeSPC, Taiga e SonarQube.

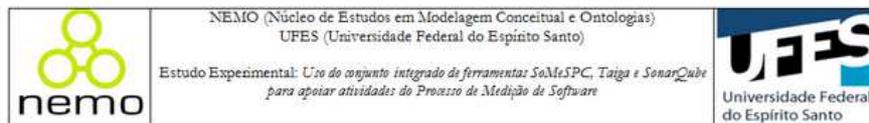
A1. Termo de Consentimento

Estudo Experimental

Estudo experimental do uso do conjunto integrado de ferramentas SoMeSPC, Taiga e SonarQube para apoiar atividades do Processo de Medição de Software

Termo de Consentimento

Termo de Consentimento do Estudo Experimental



Estudo Experimental: Uso do conjunto integrado de ferramentas SoMeSPC, Taiga e SonarQube para apoiar atividades do Processo de Medição de Software

Este estudo tem o objetivo de avaliar alguns resultados produzidos no contexto da dissertação de mestrado intitulada "Uma Abordagem Baseada em Ontologias para Integração Semântica de Ferramentas de Apoio à Medição de Software" realizada no Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo. Este estudo servirá como avaliação preliminar das funcionalidades providas pelo conjunto integrado de ferramentas implantado no LEDS.

Após o aceite a este termo de consentimento, serão apresentadas algumas questões. É garantida a confidencialidade dos dados individuais cedidos no estudo. Os dados são destinados à realização da pesquisa, não sendo usados como avaliação pessoal ou profissional. É assegurado o anonimato dos participantes na publicação dos resultados da pesquisa.

Apesar de convidado, a participação é voluntária, sendo de direito não querer participar ou abandonar a realização do estudo a qualquer momento.

Ao clicar em "Continuar", declaro ter mais de 18 anos de idade e participar voluntariamente da avaliação de resultados produzidos no contexto do trabalho "Uma Abordagem Baseada em Ontologias para Integração Semântica de Ferramentas de Apoio à Medição de Software", tendo lido as informações contidas neste termo antes de participar do estudo.

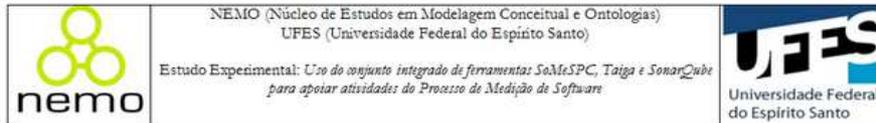
Continuar »

33% concluído

A2. Perfil do Usuário

Perfil de Participante

Perfil de Participante de Estudo Experimental



Nome *

Informe o seu nome

E-mail *

Informe o seu e-mail de contato

Formação Acadêmica *

Informe o curso ou a área da maior titulação indicada

Situação Atual da Formação Acadêmica *

Escolha a sua situação atual com relação à Formação Acadêmica

- Completo
- Incompleto

Grau de Formação Acadêmica *

Escolha o seu grau de formação acadêmica entre as opções abaixo

- Superior
- Especialização
- Mestrado
- Doutorado

Conhecimento Teórico em Medição de Software *

Selecione a opção que mais se adequa ao seu conhecimento da área de Medição de Software

- Nenhum
- Baixo (não fez nenhum curso ou disciplina e obteve algum conhecimento lendo livros ou outros materiais)
- Médio (fez alguma disciplina ou treinamento em medição de software com duração mínima de 4 horas ou trabalhou com o tema em Iniciação Científica ou Projeto de Graduação)
- Alto (é especialista no assunto, tem alguma certificação na área ou trabalhou com o tema em pesquisa de mestrado ou doutorado)

Experiência Prática em Medição de Software *

Para esta questão, considere seu tempo de experiência em atividades relacionadas à medição de software (planejamento, execução e análise)

- Nenhum
- Baixa (menos de 1 ano)
- Média (entre 1 e 3 anos)
- Alta (mais de 3 anos)

« Voltar

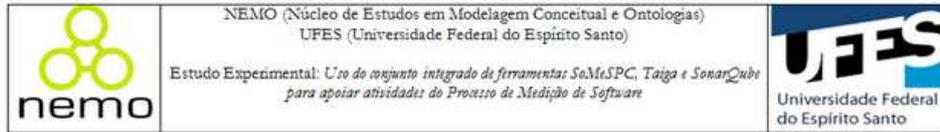
Continuar »

66% concluído

A3. Questionário de Avaliação

Avaliação do Conjunto Integrado

Questionário sobre o uso do conjunto integrado das ferramentas SoMeSPC, Taiga e SonarQube.



Você compreendeu o Processo de Medição definido para a organização? *

Informe o grau em que você compreendeu o Processo de Medição

- Sim
 Parcialmente
 Não

Justifique sua resposta para a questão [Você compreendeu o Processo de Medição definido para a organização?].

Informe uma justificativa para sua resposta na questão anterior.

Explicitar os objetivos, necessidades de informação e medidas melhorou a percepção das razões para se realizar a medição de software? *

Informe o grau em que a definição explícita dos objetivos, necessidades de informação e medidas melhorou a percepção das razões para se realizar a medição de software.

- Sim
 Parcialmente
 Não

Justifique sua resposta para a questão [Explicitar os objetivos, necessidades de informação e medidas melhorou a percepção das razões para se realizar a medição de software?].

Informe uma justificativa para sua resposta na questão anterior.

A funcionalidade "Novo Plano de Medição Integrado" é adequada para apoiar o planejamento da medição? *

Informe o grau em que a funcionalidade é adequada para apoiar o planejamento da medição.

- Sim
 Parcialmente
 Não

Justifique sua resposta para a questão [A funcionalidade "Novo Plano de Medição Integrado" é adequada para apoiar o planejamento da medição?].

Informe uma justificativa para sua resposta na questão anterior.

A funcionalidade "Novo Plano de Medição Integrado" é útil no apoio ao processo de Medição de Software? *

Informe o grau em que a funcionalidade é útil no apoio ao processo de Medição de Software

- Sim
- Parcialmente
- Não

Justifique sua resposta para a questão [A funcionalidade "Novo Plano de Medição Integrado" é útil no apoio ao processo de Medição de Software?].

Informe uma justificativa para sua resposta na questão anterior.

O uso da funcionalidade "Agendamento de Medições" é adequada para apoiar a execução de medições? *

Informe o grau em que a funcionalidade é adequada para apoiar a execução de medições.

- Sim
- Parcialmente
- Não

Justifique sua resposta para a questão [O uso da funcionalidade "Agendamento de Medições" é adequada para apoiar a execução de medições?].

Informe uma justificativa para sua resposta na questão anterior.

O uso da funcionalidade "Agendamento de Medições" é útil no apoio ao processo de Medição de Software? *

Informe o grau em que a funcionalidade é útil no apoio ao processo de Medição de Software

- Sim
- Parcialmente
- Não

Justifique sua resposta para a questão [O uso da funcionalidade "Agendamento de Medições" é útil no apoio ao processo de Medição de Software?].

Informe uma justificativa para sua resposta na questão anterior.

O uso da funcionalidade "Análise de Medições" é adequada para apoiar a análise de medições? *

Informe o grau em que a funcionalidade é adequada para apoiar a análise de medições.

- Sim
- Parcialmente
- Não

Justifique sua resposta para a questão [O uso da funcionalidade "Análise de Medições" é adequada para apoiar a análise de medições?].

Informe uma justificativa para sua resposta na questão anterior.

O uso da funcionalidade "Análise de Medições" é útil no apoio ao processo de Medição de Software? *

Informe o grau em que a funcionalidade é útil no apoio ao processo de Medição de Software

- Sim
- Parcialmente
- Não

Justifique sua resposta para a questão [O uso da funcionalidade "Análise de Medições" é útil no apoio ao processo de Medição de Software?].

Informe uma justificativa para sua resposta na questão anterior.

Comparando-se a utilização das ferramentas SoMeSPC, Taiga e SonarQube individualmente para apoiar o processo de medição de software com o conjunto integrado, em sua opinião o conjunto integrado: *

Indique o grau de benefício de se usar o conjunto integrado das ferramentas comparando-se com o uso das ferramentas individualmente.

- Muito mais benefícios do que o uso das ferramentas individualmente
- Mais benefícios do que o uso das ferramentas individualmente
- Os mesmos benefícios do que o uso das ferramentas individualmente
- Menos benefícios do que o uso das ferramentas individualmente
- Muito menos benefícios do que o uso das ferramentas individualmente

Destaque os principais benefícios obtidos a partir da utilização do conjunto integrado.

Destaque os benefícios identificados por você pela utilização das ferramentas integradas.

Você teve alguma dificuldade em utilizar o conjunto integrado de ferramentas?

Informe a(s) dificuldade(s) encontrada(s) no uso do conjunto integrado, caso exista alguma.

Você tem alguma sugestão para melhorar a o conjunto integrado de ferramentas?

Informe a(s) sua(s) sugestão(ões) com relação ao uso do conjunto integrado, caso tenha alguma.

« Voltar

Enviar



100% concluído.

Nunca envie senhas pelo Formulários Google.