

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/242730479>

# Uso de uma Ontologia de Avaliação de Software para o Desenvolvimento e Integração de Ferramentas

## Article

CITATIONS

0

READS

36

**3 authors**, including:



**Ricardo de Almeida Falbo**

Universidade Federal do Espírito Santo

**172** PUBLICATIONS **1,661** CITATIONS

[SEE PROFILE](#)



**Rodrigo Dal Moro**

Universidade Federal do Espírito Santo

**7** PUBLICATIONS **68** CITATIONS

[SEE PROFILE](#)

**Some of the authors of this publication are also working on these related projects:**



Standards Harmonization [View project](#)



Knowledge Management in Software Testing [View project](#)

# Uso de uma Ontologia de Avaliação de Software para o Desenvolvimento e Integração de Ferramentas

Ricardo de Almeida Falbo, Rodrigo Dal Moro, Ana Christina de Oliveira  
Bringunte, Murilo de Oliveira Palácio

Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO) – Universidade  
Federal do Espírito Santo (UFES)  
Av. Fernando Ferrari s/n, Campus de Goiabeiras – 29.060-900 – Vitória – ES – Brasil  
falbo@inf.ufes.br, rdalmoro@yahoo.com.br, {ana.christina.b,  
muriloop}@gmail.com

**Resumo.** *Avaliação é parte essencial de diversos processos de software, tais como Garantia da Qualidade e Avaliação e Melhoria de Processos. A avaliação de entidades nesse contexto é bastante corriqueira, sendo aplicada, dentre outros, a processos, produtos e recursos. Ainda que a avaliação de cada um desses tipos de entidades tenha suas particularidades, há aspectos comuns que, se propriamente capturados, podem servir de uma especificação base para o desenvolvimento de diferentes ferramentas. Este artigo apresenta uma ontologia de avaliação que visa capturar a conceituação básica envolvida nesse universo de discurso. Essa ontologia foi utilizada como base para o desenvolvimento de uma ferramenta de avaliação de processos no contexto de Avaliação e Melhoria de Processos e para a sua integração com uma ferramenta de apoio ao processo de Garantia da Qualidade.*

**Abstract.** *Evaluation is an essential part of several software processes, such as Quality Assurance, and Process Evaluation and Improvement. The evaluation of entities in these contexts is commonplace, and it is applied to processes, products and resources, among others. Although the evaluation of each one of these kinds of entities has its particularities, there is a set of commonalities that, if properly captured, can be used as a basic specification for the development of different tools. This paper presents an ontology of evaluation that aims to capture the common conceptualization involved in this domain of discourse. This ontology was used as a basic specification for the development of a tool supporting software process evaluation in the context of a Software Process Evaluation and Improvement process, and its integration to a tool supporting the Quality Assurance process.*

## 1. Introdução

Avaliação é parte essencial de diversos processos de software, tais como Garantia da Qualidade, Avaliação e Melhoria de Processos e Gerência de Recursos Humanos. Em cada um desses processos, um ou mais tipos de entidades são alvo de avaliação. Por exemplo, no processo de Gerência de Recursos Humanos, o foco está na avaliação do desempenho de pessoas e equipes; já no processo de Garantia da Qualidade, deve-se avaliar a aderência de artefatos e processos a padrões [Softex 2007a].

Ainda que a avaliação de cada um desses tipos de entidades tenha suas particularidades, há diversos aspectos comuns. De maneira geral, é necessário identificar características que sejam capazes de indicar a qualidade da entidade sendo avaliada, medir essas características, analisar os resultados das medições, concluir sobre a necessidade de ajustes, estabelecer ações para tratar os problemas e acompanhá-las até as suas efetivas conclusões.

Capturar a conceituação comum envolvida no domínio de avaliações de software é importante para dar clareza de que atividades de diferentes processos têm em comum conceitos importantes. Com essa percepção, é possível pensar em um arcabouço básico para o desenvolvimento de ferramentas de apoio a esses processos no que se refere à avaliação, bem como para a integração das mesmas.

Ontologias de domínio têm sido utilizadas para explicitar, pelo menos parcialmente, as conceituações de domínios relevantes para certas comunidades. Assim, no contexto deste trabalho, foi desenvolvida uma ontologia de avaliação de software, visando formalizar parcialmente o conhecimento envolvido nesse domínio, tendo como foco principal os aspectos relacionados mais diretamente com a avaliação de produtos e processos de software. Ela foi desenvolvida com o propósito de estabelecer uma conceituação comum sobre esse domínio, de modo a apoiar o desenvolvimento e a integração de ferramentas de apoio à avaliação no contexto de diferentes processos. A ontologia proposta foi utilizada como uma especificação base para apoiar o desenvolvimento e a integração de ferramentas de apoio aos processos de Avaliação e Melhoria de Processos e Garantia da Qualidade no ambiente ODE (*Ontology-based software Development Environment*) [Falbo et al. 2003], um ambiente de desenvolvimento de software baseado em ontologias.

Este artigo está estruturado da seguinte forma: a seção 2 trata brevemente do universo de discurso da ontologia proposta, avaliação de software, e os potenciais usos dessa ontologia; a seção 3 apresenta a ontologia de avaliação de software desenvolvida; a seção 4 discute o uso dessa ontologia em dois cenários diferentes: no desenvolvimento de uma ferramenta de apoio à avaliação em um processo de Avaliação e Melhoria de Processos e na integração dessa ferramenta com uma ferramenta de apoio ao processo de Garantia da Qualidade; a seção 5 discute alguns trabalhos relacionados; e finalmente, a seção 6 apresenta as conclusões e perspectivas futuras deste trabalho.

## **2. Avaliação de Software e Ontologias**

Diversos processos de software envolvem atividades relativas à avaliação de entidades. Tomando por base o modelo MPS.BR [Softex 2007a], pode-se perceber que processos como Garantia da Qualidade, Avaliação e Melhoria de Processos e Gerência de Recursos Humanos fazem referência explícita à avaliação em seus resultados esperados.

O processo de Garantia da Qualidade visa assegurar que produtos de trabalho e a execução de processos estejam em conformidade com os planos e recursos predefinidos, o que envolve, dentre outros [Softex 2007a]: (i) avaliar a aderência de produtos e processos a padrões e procedimentos; (ii) identificar, registrar e comunicar problemas e não-conformidades; (iv) estabelecer ações para tratar os problemas e acompanhá-las até as suas efetivas conclusões.

O processo de Avaliação e Melhoria de Processos, por sua vez, tem como propósito determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e para apoiar a organização a planejar, realizar e implantar melhorias contínuas nesses processos com base no entendimento de seus pontos fortes e fracos [Softex 2007a]. Dentre seus resultados esperados, os seguintes estão diretamente relacionados à avaliação: (i) avaliações dos processos padrão da organização são realizadas para identificar pontos fortes e fracos e oportunidades de melhoria; e (ii) registros das avaliações realizadas são mantidos acessíveis. Contudo, outros resultados também estão relacionados, ainda que indiretamente, com avaliação.

Por fim, o processo de Gerência de Recursos Humanos, cujo propósito é prover à organização e aos projetos os recursos humanos necessários e manter suas competências consistentes com as necessidades do negócio, envolve, dentre outros, a avaliação de desempenho de indivíduos e grupos [Softex 2007a].

Em cada um desses contextos, as avaliações assumem particularidades que começam pelo tipo de entidade que se está avaliando e vão até diferentes características a serem avaliadas e métodos de avaliação. No entanto, é possível perceber também aspectos comuns, dentre eles a forte relação com a medição. Inicialmente, é necessário definir as entidades que serão avaliadas e quais de suas características serão usadas na avaliação. Segundo, é preciso definir quais medidas serão usadas para quantificar essas características. Uma vez planejada a medição, pode-se passar efetivamente à sua execução, o que envolve a aplicação de procedimentos de medição para se obter um valor para uma medida da entidade em questão. Uma vez realizada a medição, devem-se analisar seus resultados para avaliar as entidades, gerando observações, tais como problemas e não conformidades detectados. Essas observações devem dar origem a ações para tratar os problemas detectados.

Como pode ser observado pelo exposto, avaliações permeiam diversos processos de software, guardando, ainda assim, muitas similaridades a despeito do contexto em que são realizadas. Em outras palavras, há uma conceituação comum, isto é, um conjunto de conceitos usados para interligar abstrações de entidades nesse domínio.

Uma ontologia é uma descrição formal e explícita de uma conceituação. Ao explicitar e formalmente definir os conceitos, relações, propriedades e restrições em um domínio particular, tem-se uma ontologia de domínio, a qual pode ter diversos usos, tais como [Jasper e Uschold 1999]: (i) apoiar a comunicação entre pessoas trabalhando no domínio; (ii) integração de dados, modelos e sistemas desenvolvidos de forma independente para o domínio e (iii) como uma especificação reutilizável para a construção de sistemas no domínio.

Pensando nesses usos, desenvolveu-se uma ontologia de avaliação de software, apresentada na próxima seção. Como qualquer artefato de engenharia, uma ontologia deve ser construída seguindo métodos apropriados. Para desenvolver a ontologia de avaliação de software, foi adotado o método SABiO (*Systematic Approach for Building Ontologies*) [Falbo 2004], o qual preconiza a realização das seguintes atividades: (i) identificação do propósito e especificação de requisitos, que visa identificar questões que a ontologia deve ser capaz de responder (questões de competência), (ii) captura da ontologia, que tem por objetivo capturar os conceitos, relações, propriedades e restrições

relevantes sobre o domínio em questão; e (iii) formalização, que busca escrever os axiomas da ontologia em uma linguagem formal (neste trabalho, optou-se pela lógica de primeira ordem). Paralelamente a essas atividades, ocorrem as atividades de integração com ontologias existentes, avaliação da ontologia e documentação da ontologia. SABiO advoga, ainda, o uso de uma linguagem de modelagem para facilitar a comunicação, sugerindo um perfil UML como linguagem gráfica para representação de ontologias [Mian e Falbo 2003], adotado neste trabalho.

### 3. Uma Ontologia de Avaliação de Software

Segundo o método SABiO, primeiro, deve-se identificar o propósito da ontologia. Deseja-se falar de avaliação de software de forma ampla, mas com foco maior para o uso da ontologia como uma especificação conceitual para o desenvolvimento e integração de ferramentas de apoio a atividades de avaliação.

No que se refere ao domínio em si, dois aspectos em especial foram levados em conta: (i) avaliação está intimamente ligada à medição e, portanto, aspectos relacionados à medição devem ser considerados; e (ii) ainda que avaliações ocorram no contexto de diferentes processos de software, foram especialmente consideradas avaliações de produtos e processos de software.

Uma vez que medição de software é parte do universo de discurso da ontologia de avaliação de software, esta foi desenvolvida de forma integrada com a ontologia de medição de software proposta em [Dal Moro e Falbo 2008]. Assim, a conceituação básica relacionada a entidades a serem avaliadas, seus elementos mensuráveis e medição é reutilizada da mesma. A Figura 2 apresenta um fragmento do modelo conceitual dessa ontologia, contendo os conceitos reutilizados neste trabalho.

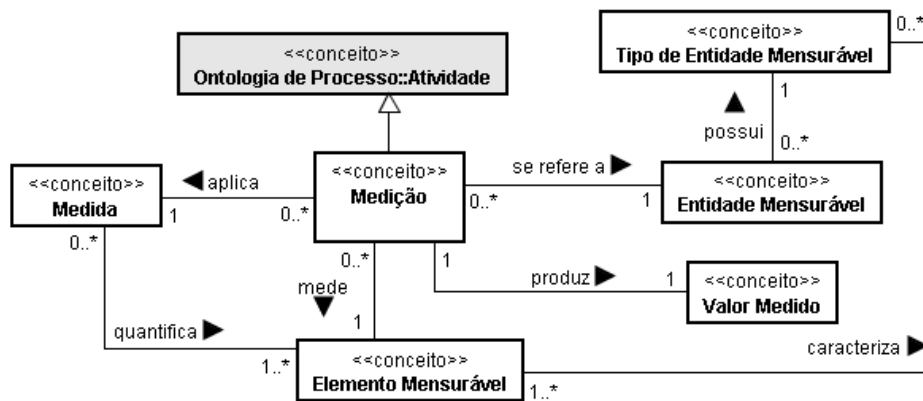


Figura 2 – Fragmento da Ontologia de Medição proposta em [Dal Moro e Falbo 2008].

Um *tipo de entidade mensurável* representa uma categoria de entidades indicando quais características (*elementos mensuráveis*) podem ser usadas para medir e avaliar *entidades mensuráveis*, instâncias concretas desse tipo. São essas entidades que efetivamente são medidas e avaliadas. No contexto de engenharia de software, são entidades mensuráveis, dentre outros, processos, artefatos, projetos e recursos. Uma *medição* é uma atividade que visa medir uma característica (*elemento mensurável*) de uma *entidade mensurável*, aplicando uma *medida* para obter um *valor*.

A ontologia de medição de software foi desenvolvida de forma integrada com uma ontologia de processo de software [Guizzardi et al. 2008] e reutiliza conceitos

importantes dessa, dentre eles atividade, destacado na figura em cinza. Da mesma forma, a ontologia de avaliação aqui proposta também reutiliza conceitos da ontologia de processo de software e conceitos oriundos desta ou da ontologia de medição são apresentados sempre destacados em cinza.

### Competência da Ontologia de Avaliação de Software

A ontologia de avaliação de software visa retratar o universo de avaliações de entidades, sejam essas avaliações objetivas, ou seja, envolvendo a análise de medições, sejam subjetivas (que não envolvem análise de medições), incluindo a indicação de ações que podem ser estabelecidas a partir de observações oriundas de uma avaliação. As questões de competência que devem ser respondidas por essa ontologia são as seguintes:

- QC1. Que entidade está sendo avaliada?
- QC2. Em relação ao uso de medidas, como uma avaliação pode ser classificada?
- QC3. Quando uma avaliação é feita com base em dados de medições, como esses dados são analisados, buscando se chegar a uma conclusão sobre a entidade que está sendo avaliada?
- QC4. Que tipos de observação podem ser produzidos em avaliações?
- QC5. Que observações de avaliação foram produzidas em uma avaliação?
- QC6. Quais são os tipos de ação que podem ser indicadas em avaliações?
- QC7. Quais foram as ações estabelecidas a partir das observações feitas em avaliações?

Em função da complexidade do domínio de avaliações de software, a ontologia foi desenvolvida de forma modular em três partes, considerando a relação entre algumas das questões de competência anteriormente relacionadas:

- Tipos de avaliações e itens a serem avaliados (questões 1 e 2);
- Avaliações baseadas em dados de medições (questão 3);
- Resultados de uma avaliação (questões 4 a 7).

### Tipos de avaliações e itens a serem avaliados

A Figura 3 mostra o modelo conceitual da parte da ontologia que trata de tipos de avaliações e itens avaliados em uma avaliação.

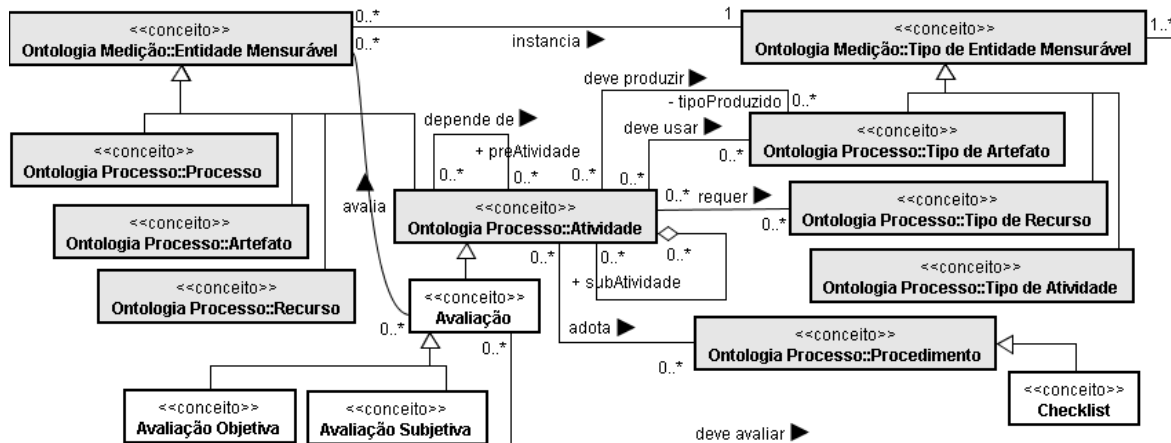


Figura 3 – Tipos de Avaliações e Itens a serem avaliados

Como mostra a figura, uma *avaliação* é uma *atividade* que avalia *entidades mensuráveis* (dentre outras, *processos*, *artefatos*, *recursos* e *atividades*), as quais são instâncias de um tipo (tal como um artefato “Documento de Requisitos do Projeto X” que é uma instância do tipo de artefato “Documento de Requisitos”). Uma *avaliação objetiva* é aquela que é feita tomando por base valores medidos em medições. Uma *avaliação subjetiva* é aquela que é feita sem considerar dados de medições.

Assim como quaisquer atividades, durante o planejamento de avaliações, devem-se definir pré-atividades, sub-atividades, tipos de artefatos a serem usados como insumo, tipos de artefatos a serem produzidos, tipos de recursos requeridos e procedimentos a serem adotados, dentre eles, *checklists*, muito utilizados em avaliações. Além disso, durante o planejamento de uma atividade de avaliação, tipicamente informa-se o tipo da entidade mensurável que deverá ser avaliada e posteriormente seleciona-se qual instância desse tipo será efetivamente avaliada. Assim, o seguinte axioma deve ser observado:

- Se uma avaliação  $av_1$  avalia uma entidade mensurável  $e_1$ , então deve haver um tipo de entidade  $t_1$  que  $av_1$  deve avaliar, tal que  $e_1$  seja uma instância de  $t_1$ .

$$(\forall av_1 \in \text{Avaliação}, e_1 \in \text{EntidadeMensuravel}) \text{avalia}(av_1, e_1) \rightarrow (\exists t_1 \in \text{TipoEntidadeMensuravel}) \\ (\text{instancia}(e_1, t_1) \wedge \text{deveAvaliar}(av_1, t_1))$$

### **Avaliações baseadas em dados de medições**

Conforme anteriormente citado, *avaliações objetivas* são feitas com base em valores medidos em medições. Para tal, é necessário analisar os valores medidos e a questão de competência QC3 pode ser desmembrada em questões mais detalhadas, a saber:

- QC3.1. Quais são os valores medidos analisado durante uma análise de medição?
- QC3.2. A que medidas se referem os valores sendo analisados?
- QC3.3. Que modelo de análise de medição é adotado em uma análise de medição?
- QC3.4. Quais são as medidas tratadas em um modelo de análise de medição?
- QC3.5. Quais são os critérios de decisão de um modelo de análise de medição baseado em critérios?
- QC3.6. Quais são as premissas e a conclusão de um critério de decisão?
- QC3.7. Qual foi a conclusão obtida em uma análise de medição?

A Figura 4 mostra o modelo conceitual da parte da ontologia que trata de avaliações objetivas. Uma *avaliação objetiva* é uma atividade de avaliação que obrigatoriamente é composta de uma ou mais atividades de *análise de medição*. Uma *análise de medição* é uma atividade que visa analisar os valores medidos de medidas, adotando um *modelo de análise de medição* como *procedimento* para atribuir uma conclusão que indique, de algum modo, uma característica da entidade sendo mensurada. Modelos de análise de medição são desenvolvidos para permitir a análise de um conjunto de medidas. No contexto de software, um tipo de modelo de análise de medição bastante utilizado são os modelos de análise de medição baseados em critérios. Um *modelo de análise de medição baseado em critérios* é descrito por um conjunto de *critérios de decisão*. Cada critério de decisão é uma sentença que estabelece uma *conclusão* a partir de um conjunto de *premissas*, descrevendo o nível de confiança de um resultado [ISO/IEC 2002].

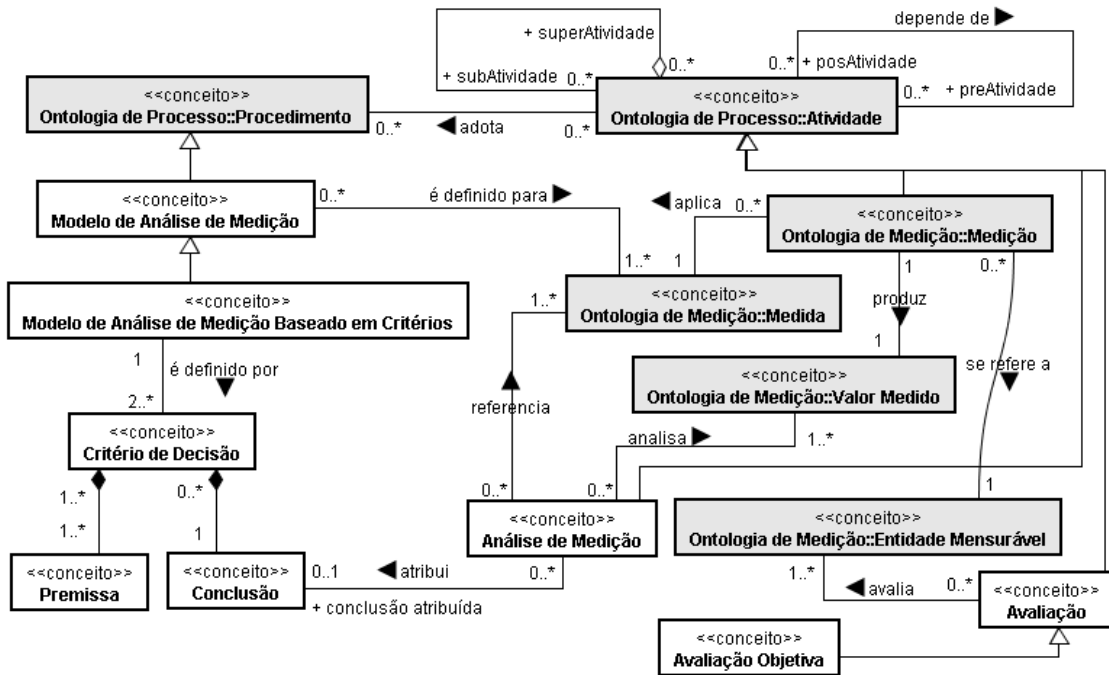


Figura 4 –Avaliações baseadas em dados de medições.

Vale destacar que alguns dos conceitos da ontologia de avaliação são especializações de conceitos da ontologia de processo, que impõem restrições específicas para o domínio de avaliações de software. Tais restrições não são capturadas por um modelo estrutural tipicamente epistemológico como o diagrama apresentando na Figura 4, bem como outras restrições ontológicas. Assim, é necessário descrever tais restrições na forma de axiomas, dentre elas:

- Se uma avaliação  $av_1$  é uma avaliação objetiva, então ela é composta de pelo menos uma atividade de análise de medição  $am_1$ .

$$(\forall av_1 \in Avaliacao) avaliacaoObjetiva(av_1) \rightarrow (\exists am_1 \in AnaliseMedicao) (subAtividade(am_1, av_1))$$

- Se uma atividade de análise de medição  $a_1$  analisa um valor medido  $v_1$  produzido por atividade de medição  $med_1$ , então  $a_1$  depende de  $med_1$ .

$$(\forall a_1, med_1, v_1) (analisa(a_1, v_1) \wedge produz(med_1, v_1)) \rightarrow dependeDe(a_1, med_1)$$

- Se uma atividade de análise de medição  $a_1$  referencia uma medida  $m_1$ , então um procedimento  $p_1$  dentre os procedimentos adotados por  $a_1$  deve ser um modelo de análise de medição definido para analisar a medida  $m_1$ .

$$(\forall a_1 \in AnaliseMedicao, m_1 \in Medida) referencia(a_1, m_1) \rightarrow (\exists p_1 \in ModeloAnaliseMedicao) (adota(a_1, p_1) \wedge ehDefinidoPara(p_1, m_1))$$

- Se uma análise de medição  $am_1$  é sub-atividade de uma avaliação objetiva  $ao_1$  e analisa um valor medido  $v_1$ , então a medição  $med_1$  que produziu o valor medido  $v_1$  deve se referir a uma entidade mensurável  $em_1$  que é avaliada por  $ao_1$ .

$$(\forall am_1 \in AnaliseMedicao, ao_1 \in AvaliacaoObjetiva, v_1 \in ValorMedido) analisa(am_1, v_1) \wedge (subAtividade(am_1, ao_1)) \rightarrow (\exists em_1 \in EntidadeMensuravel, med_1 \in Medicao) (produz(med_1, v_1) \wedge seRefereA(med_1, em_1) \wedge avalia(ao_1, em_1))$$



- Se uma análise de medição  $am_1$  analisa um valor medido  $v_1$ , então  $am_1$  referencia alguma medida  $m_1$  tal que o valor medido  $v_1$  é produzido por uma medição  $med_1$  que aplica a medida  $m_1$ .

$$(\forall am_1 \in AnaliseMedicao, v_1 \in ValorMedido) analisa(am_1, v_1) \rightarrow (\exists m_1 \in Medida, med_1 \in Medicao) (referencia(am_1, m_1) \wedge produz(med_1, v_1) \wedge aplica(med_1, m_1))$$

## Resultados de uma avaliação

A Figura 5 mostra o modelo conceitual da parte da ontologia que trata dos resultados de uma avaliação. Uma *avaliação* que gera como produto um *documento de avaliação* contendo *observações de avaliação*, i.e., evidências objetivas documentadas [ABNT 2000], que podem ser *pontos fortes* ou *problemas*. Um *ponto forte* é uma característica excepcionalmente boa de uma entidade, observada em uma avaliação. *Problemas* podem ser *defeitos*, *pontos fracos*, *não conformidades* e *oportunidades de melhoria*. Um *ponto fraco* é uma característica considerada inadequada ou que não atende aos requisitos da entidade. Uma *oportunidade de melhoria* é uma característica que pode ser melhorada, mas que atende aos requisitos mínimos colocados para a entidade. Uma não conformidade é um não atendimento de um requisito de qualidade [ABNT 2000]. Por fim, um *defeito* é uma imperfeição em um produto [Koscianski e Soares 2006].

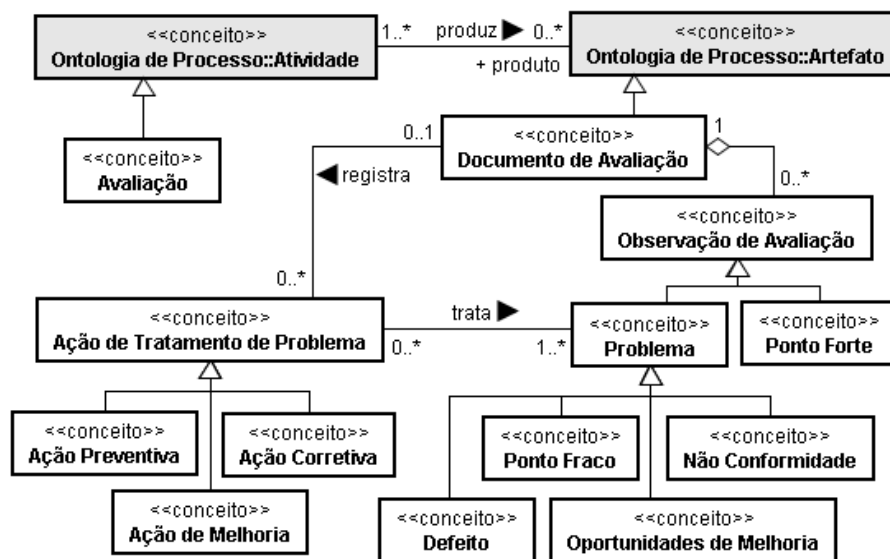


Figura 5 –Resultados de uma avaliação

A partir de observações realizadas em uma avaliação, *ações de tratamento de problema* podem ser apontadas e registradas no documento de avaliação. Uma ação pode ser: uma *ação corretiva*, que visa eliminar a causa de uma não conformidade identificada ou de outra situação indesejável de uma entidade; uma *ação preventiva*, que visa eliminar a causa de uma não conformidade potencial ou outra situação potencialmente indesejável; ou uma *ação de melhoria*, que visa aumentar a capacidade de satisfazer os requisitos de qualidade já existentes de uma entidade [ABNT 2000].

De maneira análoga à parte da ontologia que trata de avaliações objetivas, o modelo da Figura 5 não é capaz de capturar toda a conceituação do domínio, sendo necessário descrever restrições na forma de axiomas. Entretanto, devido a limitações de espaço, os mesmos não são apresentados neste artigo.

## Avaliação da Ontologia

Uma maneira de se avaliar uma ontologia consiste em instanciar a mesma para avaliar se a ontologia é capaz de responder às questões de competência colocadas e, por conseguinte, representar as situações que pretende descrever. Assim, a ontologia foi instanciada, considerando tanto avaliações de artefatos quanto de processos de software. Devido às limitações de espaço, a seguir, apresentamos apenas parte de uma instanciamento feita para uma avaliação de processo de software. No texto, os conceitos instanciados estão sublinhados.

Seja uma unidade organizacional (UO) que está sendo avaliada visando uma certificação no nível G do MPS.BR [Softex 2007a]. Dois de seus projetos (P1 e P2) foram selecionados para a avaliação. Uma avaliação *av* foi realizada para caracterizar a aderência da UO ao resultado esperado GPR 3<sup>1</sup> do MPS.BR. A atividade *av* possui uma sub-atividade de análise de medição *am* que tem por objetivo atribuir, com base em valores medidos na medição de cada um dos projetos, o nível de implementação de GPR3 na UO. Os valores medidos para P1 e P2 foram ambos L (largamente implementado). Usando como modelo de análise de medição baseado em critérios as regras para agregar a caracterização dos resultados esperados dos processos (Tabela 8 do Guia de Avaliação do MPS.BR [Softex 2007b]), que referenciam a medida “Grau de Implementação de Resultado Esperado do Processo em Projeto”, chegou-se à conclusão de que a caracterização agregada para UO é L, tendo em vista que o primeiro critério desse modelo indica como premissa que, se todas as caracterizações nos projetos forem iguais, então a caracterização agregada para a UO é a mesma atribuída nos projetos. Por fim, durante essa avaliação, percebeu-se o seguinte ponto fraco: o ciclo de vida praticado no projeto P1 não estava totalmente aderente ao modelo de ciclo de vida selecionado. E se propôs a seguinte ação preventiva para tratar o problema detectado: criar um novo item no *checklist* de verificação da execução do projeto para avaliar se o modelo de ciclo de vida definido para o projeto está sendo seguido.

## 4. Uso da Ontologia de Avaliação de Software como uma Especificação

A ontologia de avaliação anteriormente apresentada foi usada como uma especificação base para o desenvolvimento de uma ferramenta de apoio à avaliação de processos de software, denominada AvaliaODE, e para a reengenharia de uma ferramenta de apoio ao processo de Garantia da Qualidade, denominada GQA-ODE, visando à integração com a primeira. Ambas foram desenvolvidas no ambiente ODE, um ambiente de desenvolvimento de software baseado em ontologias [Falbo et al. 2003] e tiveram seus requisitos inspirados em resultados esperados estabelecidos no MPS.BR. Vale ressaltar, ainda, que ambas foram desenvolvidas desconsiderando o uso de dados de medições nas avaliações e, portanto, a parte da ontologia que trata desse aspecto não foi utilizada.

AvaliaODE tem por objetivo apoiar a avaliações no contexto de um processo de Avaliação e Melhoria de Processos de Software. Em seu estágio atual, ela provê funcionalidades de apoio: (i) à elaboração de uma proposta de escopo para a avaliação de um processo; (ii) à avaliação de um processo segundo *checklists* previamente

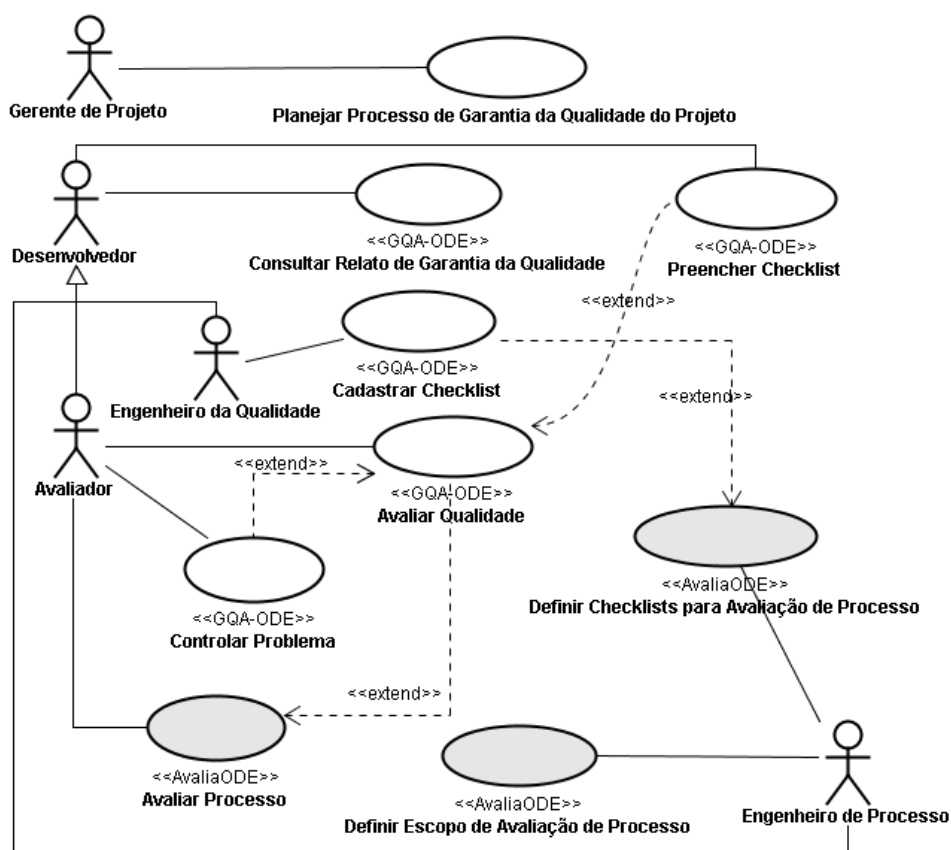
---

<sup>1</sup> GPR 3 é um resultado esperado do Processo de Gerência de Projetos do MPS.BR, que diz que o modelo e as fases do ciclo de vida do projeto devem ser definidos.

definidos; (iii) ao registro de observações feitas durante uma avaliação; e (iv) à avaliação de produtos no contexto de avaliações de processos.

GQA-ODE, por sua vez, visa apoiar um processo de Garantia da Qualidade e seus requisitos funcionais incluem, dentre outros, a definição de *checklists* para apoiar a avaliação da qualidade e o registro e tratamento de não conformidades detectadas.

GQA-ODE foi desenvolvida antes da elaboração da ontologia de avaliação, enquanto AvaliaODE foi desenvolvida tomando por base essa ontologia. Durante o desenvolvimento de AvaliaODE, percebeu-se que muitas de suas funcionalidades eram diretamente relacionadas a funcionalidades de GQA-ODE. De fato, ambas buscavam apoiar avaliações de software e muito poderia ser compartilhado. Assim, foi feito um trabalho de reengenharia de GQA-ODE visando à integração com AvaliaODE. A Figura 6 mostra um diagrama de casos de uso contendo as funcionalidades de ambas as ferramentas, já considerando a reengenharia de GQA-ODE. Como se pode perceber pelas associações de extensão entre casos de uso mostradas na figura, diversas funcionalidades de GQA-ODE foram reutilizadas por AvaliaODE, tendo como resultado uma integração em nível de controle, a qual se refere à habilidade de uma ferramenta notificar ou iniciar uma ação em outra ferramenta [Falbo et al. 2003].



**Figura 6 –Funcionalidades de GQA-ODE e AvaliaODE Integradas.**

Para utilizar GQA-ODE, é necessário, primeiro, ter um processo de Garantia da Qualidade (GQA) definido para o projeto. Essa atividade, ainda que mostrada na Figura 6, é feita utilizando a ferramenta de definição de processos de ODE [Bertollo et al. 2006], a qual foi estendida para contemplar particularidades do processo de GQA.

No planejamento do processo de GQA, os marcos e pontos de controle do projeto são apresentados e o gerente de projeto pode definir as atividades de GQA a serem realizadas nesses pontos. Para cada atividade definida, o gerente de projeto deve definir, ainda, se essa atividade vai avaliar produtos (artefatos) ou processos e definir o tipo do artefato ou o processo padrão que será avaliado. Além disso, para cada atividade do processo de GQA, assim como para os demais processos, podem ser definidos sub-atividades, artefatos (consumidos e produzidos), recursos (humanos, hardware e software) e procedimentos necessários para a sua execução.

Uma vez planejado o processo de GQA do projeto e atingido um marco desse projeto no qual está prevista a realização de uma atividade de GQA, o caso de uso “Avaliar Qualidade” pode ser realizado. Caso a atividade de GQA tenha definido como tipo do item a ser avaliado um processo padrão, o processo do projeto corrente que instancia o processo padrão a ser avaliado é automaticamente selecionado como o item a ser avaliado. Caso a atividade de GQA tenha definido como tipo do item a ser avaliado um tipo de artefato, o avaliador deve selecionar, dentre os artefatos desse tipo disponíveis, aquele que será avaliado. Definido o item a ser avaliado, o avaliador pode selecionar *checklists* previamente preenchidos para usar como base para a sua avaliação ou pode ele mesmo preencher *checklists*. Finalmente, o avaliador, consultando os *checklists* preenchidos, pode tecer suas considerações acerca da avaliação e registrar problemas. Para cada um dos problemas detectados, ações de tratamento devem ser estabelecidas e acompanhadas até a sua efetiva conclusão, o que é apoiado pelo caso de uso “Controlar Problema”.

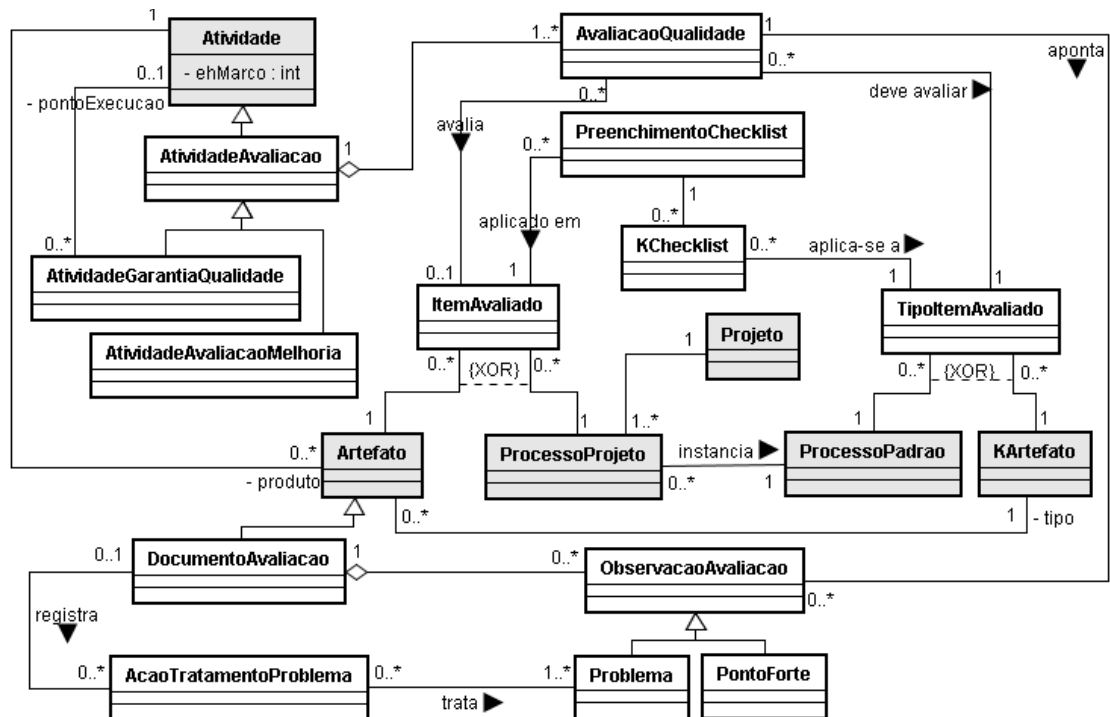
Além das funcionalidades anteriormente mencionadas, GQA-ODE permite que se cadastrem *checklists* a serem utilizados na avaliação de processos e produtos da organização. Esses *checklists* podem ser preenchidos durante a realização de uma atividade de avaliação ou por um desenvolvedor durante a realização de uma atividade qualquer do processo de software.

Para apoiar uma avaliação no contexto de um processo de Avaliação e Melhoria de Processos de Software usando AvaliaODE, deve-se inicialmente definir o escopo da avaliação. O engenheiro de processo deve escolher o processo padrão que será avaliado e, opcionalmente, os tipos de artefatos por ele produzidos que também deverão ser avaliados. Neste momento ele deve indicar também, dentre os projetos que instanciaram o processo padrão a ser avaliado, quais os selecionados para a avaliação.

O próximo passo consiste em indicar os *checklists* a serem usados na avaliação. Neste momento, caso seja necessário, o engenheiro de processos pode criar um novo *checklist*, usando para tal o cadastro de *checklists* provido por GQA-ODE.

Concluído o planejamento da avaliação, a mesma pode ser realizada. Por meio do caso de uso “Avaliar Processo” de AvaliaODE, o avaliador seleciona os artefatos ou o processo a serem efetivamente avaliados e realiza a avaliação propriamente dita, realizando o caso de uso “Avaliar Qualidade” de GQA-ODE.

A Figura 7 mostra parcialmente o *framework* construído tomando por base a ontologia de avaliação e que foi usado para a construção de AvaliaODE e a reengenharia de GQA-ODE. As classes destacadas em cinza são classes já existentes no núcleo do ambiente ODE que trata de processos de software.



**Figura 7 – Framework de Avaliação.**

Comparando os diagramas da ontologia mostrados nas figuras 3 e 5 com o diagrama da Figura 7, pode-se notar que o último reflete basicamente a mesma conceituação dos dois primeiros. Contudo, critérios de design, tais como modularidade e acoplamento, nortearam algumas decisões, tal como a criação das classes *ItemAvaliado* e *TipoItemAvaliado*. Tais classes buscam isolar as classes do núcleo de processos de ODE do *framework* de avaliação, procurando deixá-los menos acoplados. Outra decisão se refere à decomposição de uma atividade de avaliação em várias avaliações, uma para cada item avaliado, e a associação de uma avaliação com as observações nela apontadas. Vale ressaltar, ainda, que a classe *KArtefato* se refere a tipos de artefatos. Por fim, a porção do modelo da Figura 7 que trata de observações de avaliação e ações de tratamento de problema é praticamente igual à ontologia, ainda que na Figura 7 tenham sido omitidas as especializações de *Problema* e *AcaoTratamentoProblema*.

## 5. Trabalhos Correlatos

Atualmente, muita atenção tem sido dada ao uso de ontologias na Engenharia de Software, com propósitos bastante variados. Em [Calero et al. 2006], por exemplo, há o relato de diversos trabalhos dessa natureza, dentre eles uma ontologia para medição de software. Entretanto, não encontramos trabalhos enfocando diretamente o domínio de avaliações de software. Assim, a seguir, discutimos alguns trabalhos relacionados com as ferramentas desenvolvidas e integradas usando a ontologia proposta neste trabalho.

De ante mão, vale a pena destacar que as funcionalidades providas por AvaliaODE e GQA-ODE são relativamente simples e não se distinguem muito de funcionalidades providas por outras ferramentas que, de maneira geral, provêm até mais funcionalidades de apoio. De fato, a característica marcante do trabalho aqui apresentado é a integração das ferramentas e o compartilhamento de dados e funcionalidades.

Um trabalho bastante correlato ao aqui apresentado foi realizado no âmbito da Estação TABA. Nesse ambiente há apoio tanto à Garantia da Qualidade [Montoni et al. 2006], quanto à avaliação de processo, este por meio da ferramenta AvalPro [Andrade 2005]. Montoni et al. (2006) adicionaram à ferramenta de apoio à adaptação de processos da Estação TABA uma funcionalidade que gera um *checklist* para avaliação da aderência de produtos e processos de projetos de software a partir do processo adaptado para o projeto. Durante uma avaliação da aderência de processos e produtos, os membros do Grupo de Garantia da Qualidade de Produtos e Processos (GQPP) podem identificar e registrar problemas e não-conformidades encontrados. Além disso, os membros do GQPP também podem estabelecer ações corretivas juntamente com os responsáveis por resolver as não-conformidades e acompanhá-las até suas efetivas conclusões, o que é feito com o apoio da ferramenta ActionPlanManager, que permite a definição de planos de ação. Uma característica marcante desse trabalho é a geração automática de *checklists* com base no processo de software adaptado. O *checklist* gerado, contudo, se limita a perguntar se cada um dos artefatos previstos no processo está sendo produzido. Em GQA-ODE, é possível planejar um processo de Garantia da Qualidade e definir os *checklists* apropriados a serem aplicados, o que possibilita uma avaliação de produto mais detalhada. No que se refere ao tratamento de problemas, o apoio oferecido pelas duas ferramentas é muito similar.

AvalPro [Andrade 2005] visa apoiar a avaliação de processos instanciados em projetos da organização. Dentre suas principais funcionalidades, destacam-se: (i) planejar a avaliação do processo; (ii) avaliar processos; (iii) realizar avaliação *post-mortem*; (iv) calcular medidas para monitoração dos processos; (v) analisar resultados preliminares; (vi) realizar reunião de consenso; (vii) elaborar o relatório de avaliação do processo instanciado; e (viii) comunicar resultados da avaliação.

Assim como AvalPro, outros trabalhos também tratam de ferramentas de apoio à avaliação de processos, dentre eles FAMP [Siquara 2007] e *Appraisal Assistant* [SQI 2007]. *Appraisal Assistant* visa apoiar avaliações de capacitação de processo ou de maturidade organizacional, buscando atender requisitos dos processos de avaliação da ISO/IEC 15504 [ISO/IEC 2003] como também do método SCAMPI [SEI 2006]. Dentre suas principais características podem-se citar: (i) apoio aos métodos de avaliação acima citados; (ii) conversão de resultados dos diferentes modelos; (iii) suporte a medidas para acompanhamento dos processos de avaliações; (iv) apoio à rastreabilidade entre os artefatos avaliados; e (v) apoio à geração de relatórios da avaliação de processo. Já FAMP provê apoio automatizado para a realização segundo o método SCAMPI. Ela proporciona suporte à identificação dos pontos fortes e fracos de um processo e apóia o acompanhamento dos planos de ação elaborados durante a avaliação de um processo.

Pode-se perceber que AvalPro e AvaliaODE têm basicamente o mesmo propósito, sendo mais gerais e não estando diretamente vinculados a um método de avaliação específico, como ocorre com *Appraisal Assistant* e FAMP. Em relação às funcionalidades providas pelas ferramentas, claramente há muita sobreposição, sendo que inegavelmente AvaliaODE provê um conjunto apenas limitado de funcionalidades quando comparada com outras ferramentas como AvalPro e *Appraisal Assistant*, que tratam, inclusive, o uso de medidas, ainda que parcialmente.

## 6. Conclusões

Avaliações permeiam diversos processos de software, tendo aspectos particulares e comuns em cada um deles. Visando capturar os aspectos comuns, foi desenvolvida uma ontologia para o domínio de avaliações de software. Essa ontologia foi utilizada como uma especificação base (modelo de domínio) para o desenvolvimento e reengenharia de ferramentas do ambiente de desenvolvimento de software ODE, visando à integração das mesmas. A partir do modelo de domínio estabelecido, um *framework* foi construído (projeto e implementação de domínio), em uma abordagem de Engenharia de Domínio. Enquanto o modelo de domínio foi a base para a análise de requisitos, esse *framework* foi a base para o projeto e a implementação de uma ferramenta de apoio à avaliação no contexto de um processo de Avaliação e Melhoria de Processos e para a reengenharia de uma ferramenta de apoio ao processo de Garantia da Qualidade.

As ferramentas em si, isoladamente, provêm funcionalidades básicas para apoiar os respectivos processos e, portanto, nesse aspecto, há diversas ferramentas relatadas na literatura que provêm funcionalidades análogas ou mais avançadas. Assim, no que se refere às ferramentas, muitas melhorias podem ser feitas ainda. Apenas para citar uma delas, as avaliações ainda não consideram o uso de dados de medições e, portanto, uma evolução seria prover apoio a avaliações objetivas. De fato, o próprio *framework* ainda não contempla essa questão e um trabalho a ser desenvolvido é a construção de um *framework* a partir da ontologia de medição e sua integração com o *framework* de avaliação. Esses *frameworks* integrados poderiam ser usados para aperfeiçoar as ferramentas apresentadas neste artigo.

## Agradecimentos

Este trabalho foi realizado com o apoio da VixTeam Consultoria e Sistemas e da Projeta Sistemas, empresas parceiras que têm financiado o projeto e dado feedback de sua aplicação a casos reais.

## Referências

- ABNT (2000) NBR ISO 9000 – *Sistemas de gestão da qualidade – Fundamentos e vocabulário*.
- Andrade, J.M.S., (2005) “Avaliação de Processos de Software em Ambientes de Desenvolvimento de Software Orientados à Organização”, Dissertação de Mestrado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- Bertollo, G., Segrini, B.M., Falbo, R.A. (2006) “Definição de Processos de Software em um Ambiente de Desenvolvimento de Software Baseado em Ontologias”. V Simpósio Brasileiro de Qualidade de Software. Vila Velha, Brasil, pp. 72-86.
- Calero, C., Ruiz, F., Piattini, M. (eds) (2006) *Ontologies for Software Engineering and Software Technology*, Springer-Verlag Berlin Heidelberg.
- Dal Moro, R., Falbo, R.A. (2008) “Uma Ontologia para o Domínio de Qualidade de Software com Foco em Produtos e Processos de Software”, III Workshop on Ontologies and Metamodeling Software and Data Engineering, XXII Simpósio Brasileiro de Engenharia de Software – SBES’2008, Campinas, Brasil.

- Falbo, R. A., Natali, A. C. C., Mian, P.G., Bertollo, G., Ruy, F.B. (2003) “ODE: Ontology-based software Development Environment”, In: Memórias de IX Congresso Argentino de Ciencias de la Computación, p. 1124-1135, La Plata, Argentina.
- Falbo, R. A. (2004) “Experiences in Using a Method for Building Domain Ontologies” Proc. of the 16th International Conference on Software Engineering and Knowledge Engineering, International Workshop on Ontology In Action, Banff, Canada.
- Guizzardi, G. Falbo, R.A. Guizzardi, R.S.S. (2008) “Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology”, Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments, Recife, Brazil.
- ISO/IEC (2002) ISO/IEC 15939:2002 – *Software Engineering – Software Measurement Process*.
- ISO/IEC (2003), ISO/IEC 15504 - *Information Technology – Process Assessment*.
- Jasper, R., Uschold, M. (1999) “A Framework for Understanding and Classifying Ontology Applications”, Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden.
- Koscianski, A., Soares, M.S. (2006) *Qualidade de Software*, Editora Novatec.
- Mian, P.G., Falbo, R.A. (2003) “Supporting Ontology Development with ODEd”, Journal of the Brazilian Computer Science, vol. 9, no. 2, pp 57-76.
- Montoni, M., et al. (2006) “Uma Abordagem de Garantia da Qualidade de Processos e Produtos de Software com Apoio de Gerência de Conhecimento na Estação TABA”, V Simpósio Brasileiro de Qualidade de Software, Vila Velha, Brasil.
- SEI – Software Engineering Institute (2006) Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document, CMU/SEI-2006-HB-002.
- Siquara, E. C., (2007) “Uma Proposta de Ferramenta para Automação de Avaliações CMMI Baseadas no Método SCAMPI”, Dissertação de Mestrado, Universidade de Salvador, Salvador, Brasil.
- Softex (2007a) MPS.BR – Melhoria de Processo do Software Brasileiro: Guia Geral, Versão 1.2.
- Softex (2007b) MPS.BR – Melhoria de Processo do Software Brasileiro: Guia de Avaliação, Versão 1.1.
- SQI – Software Quality Intitute (2007) Appraisal Assistant Beta, Queensland: Griffith University. <<http://www.sqi.gu.edu.au/AppraisalAssistant/>>.