Apoio à Área de Processo Verificação de Software em Ambientes de Desenvolvimento de Software Orientados a Organização

Andrea Oliveira Soares Barreto, Ana Regina Cavalcanti da Rocha

COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação
Caixa Postal 68511 – CEP: 21945-970
Rio de Janeiro – RJ – Brasil

ansoares@cos.ufrj.br, darocha@cos.ufrj.br

Resumo. *A necessidade de verificar efetivamente a qualidade do software, isto é, determinar se atende aos requisitos funcionais e não-funcionais especificados é da maior importância. Este trabalho apresenta a definição de um processo de verificação de software, cujo objetivo é apoiar essa verificação durante todo o processo de desenvolvimento e/ou manutenção, possibilitando a melhoria da qualidade dos produtos. Com o objetivo de fornecer apoio à execução do processo proposto, uma ferramenta foi desenvolvida. Essa ferramenta está inserida no contexto dos Ambientes de Desenvolvimento de Software Orientados a Organização.*

Abstract. *The need to effectively verify software quality, i.e., to determine if the specified functional and non-functional requirements are met, is now more compelling then ever. This paper presents the definition of a software verification process. The goal of this process is to support this verification through all development or maintenance processes providing the means to deliver better products. To support the execution of the proposed process, a tool has been developed. This tool is part of the Enterprise-Oriented Software Development Environments.*

1. Introdução

O papel dos computadores na sociedade, o poder de processamento que eles oferecem e a sua variedade de utilizações em domínios diferentes têm crescido drasticamente. Além disso, a capacidade de se colocar produtos no mercado em um intervalo de tempo cada vez menor tem muitas vezes sido fundamental para o sucesso de uma empresa. Desse modo, existe uma constante pressão para que o software seja produzido de forma mais rápida, com alta qualidade e mantendo um custo adequado.

No contexto atual do desenvolvimento de software, é cada vez maior a necessidade de verificar efetivamente a qualidade do software [Golubic 2003]. Uma das formas de se realizar essa verificação buscando obter produtos de melhor qualidade é introduzir atividades de verificação ao longo do processo de desenvolvimento. Sem dúvida, uma verificação efetiva aumenta a visibilidade do processo de desenvolvimento e reduz os riscos do projeto [Singh e Bawa 1995].

Há diversas definições para o termo verificação. A norma internacional ISO/IEC 12207 (1995) define verificação como sendo a confirmação, por exame e fornecimento de evidência objetiva, do atendimento aos requisitos especificados. O objetivo da verificação é determinar se os produtos de software de uma atividade atendem completamente aos requisitos ou condições impostas a eles nas atividades anteriores [ISO/IEC 12207 1995]. De acordo com Pfleeger (2004), a verificação assegura que cada função opera corretamente. Para o CMMI (Capability Maturity Model Integration) [CMMI 2002], a área de processo verificação deve garantir que os artefatos verificados atendem aos requisitos especificados para eles.

Este artigo descreve uma abordagem para apoiar a execução da Verificação de Software em Ambientes de Desenvolvimento de Software Orientados à Organização. Com essa finalidade foi definido um processo de verificação com base nos estudos das técnicas de engenharia de software para a verificação de software da ISO 12207 (1995) e do CMMI (2002). Além disso, foi implementada uma ferramenta denominada *VerificationPlan* que apóia tal processo e está inserida no contexto dos Ambientes de Desenvolvimento de Software Orientados à Organização.

A seção a seguir discute a importância da verificação em um projeto de software e suas relações com a qualidade do produto final. A seção 3 apresenta a Estação TABA na qual este trabalho está inserido. Na seção 4 são abordados os processos de verificação descritos na literatura que serviram como base para o processo definido neste trabalho. A seção 5 apresenta o processo de verificação definido. A seção 6 discute a abordagem proposta para verificação de software, apresentando a ferramenta *VerificationPlan* que apóia algumas atividades do processo de verificação definido. Finalmente, a seção 7 apresenta as considerações finais.

2. Qualidade de Software X Verificação de Software

A demanda e a preocupação com a produção de software de alta qualidade a baixo custo passaram a ser um dos motivos que culminaram na introdução de atividades agregadas sob o nome de garantia da qualidade de software ao longo de todo o processo de desenvolvimento de software. A qualidade é um aspecto que deve ser tratado simultaneamente com o processo de desenvolvimento, pois ela não pode ser imposta depois que o produto está finalizado [Rocha *et al.* 2001].

Dentre as atividades de garantia de qualidade de software está a Verificação. Um dos principais objetivos da verificação é a remoção de defeitos. Defeitos podem ser inseridos no produto de software durante todas as fases do processo de desenvolvimento. A Tabela 1 apresenta a relação entre as atividades do processo de desenvolvimento com a inserção e a remoção de defeitos [Kan 2003].

Idealmente, os defeitos de um produto de software devem ser detectados o mais cedo possível, para evitar retrabalho. Além disso, o custo para detectar e corrigir defeitos cresce bastante à medida que eles são propagados para fases posteriores do processo de desenvolvimento. Estudos mostram que o custo de corrigir um defeito de projeto ou de codificação na própria fase é entre 10 a 100 vezes menor do que o custo de corrigi-lo na fase de testes [Andersson 2003]. Essa é uma das principais motivações para o uso da verificação de software, que possibilita a detecção dos defeitos mais cedo.

Assim sendo, para serem mais efetivas, as atividades de verificação devem estar completamente integradas ao processo de desenvolvimento [Andersson 2003].

Tabela 1. Relação entre as atividades do processo de desenvolvimento com a inserção e a remoção de defeitos [Kan 2003]

Fase do Desenvolvimento	Inserção de Defeito	Remoção de Defeito
Requisitos	Processo de levantamento de requisitos e desenvolvimento das especificações	Verificação dos requisitos
Projeto	Atividades de projeto	Verificação do projeto
Projeto Detalhado	Atividades de projeto	Verificação do projeto
Implementação	Codificação	Verificação do código
Integração	Processo de integração	Testes de integração
Teste de Unidade	Correções inadequadas	Testes
Teste de Componente	Correções inadequadas	Testes
Teste de Sistema	Correções inadequadas	Testes

Devido à importância da verificação de software, muitos estudos têm sido realizados com o objetivo de entender, avaliar e modelar técnicas e métodos para verificação de software [Andersson 2003] [Laitenberger *et al.* 2002] [Thelin 2002] [Berling e Runeson 2003]. Duas atividades principais de verificação podem ser identificadas: Revisão por Pares e Testes [Thelin 2002] [CMMI 2002]. Revisão por pares é um método estático de verificação no qual um artefato é examinado por qualquer integrante da equipe do projeto (inclusive a equipe de garantia da qualidade), com propósito de detectar defeitos [Laitenberger *et al.* 2002]. É conduzida de acordo com processos bem definidos que se constituem de vários papéis e responsabilidades de revisores. Logo, cada revisão deve ser planejada cuidadosamente. Revisões por pares são implementadas através de inspeções, *walkthroughs* estruturados ou outros métodos de revisão [Laitenberger *et al.* 2002] [Silva e Travassos 2003].

Teste é um método para examinar o comportamento do software através de sua execução [Juristo *et al.* 2003] e são usados depois que partes do software foram implementadas. O ideal é que as revisões por pares e os testes sejam usados em conjunto, sempre que possível.

Alguns estudos demonstram que para que a verificação seja efetiva, critérios de verificação devem ser definidos para garantir que os artefatos verificados atendem aos requisitos para eles especificados [ISO/IEC 12207 1995] [CMMI 2002]. Os critérios de verificação são dependentes do artefato a ser verificado. Uma pesquisa na literatura foi realizada para este trabalho e resultou em uma lista de critérios, para cada artefato, baseada em [Myers 1979] [Porter *et. al.* 1995] [Humphrey 1995] [ISO/IEC 12207 1995]. Por razão de escopo essa lista não será apresentada neste artigo. Como exemplo, para uma especificação de requisitos do software, alguns dos critérios a considerar são: clareza, completeza, consistência externa e testabilidade. Para a verificação do código de unidades, alguns dos critérios são: rastreabilidade para os requisitos do software, cobertura do teste de unidade e viabilidade de manutenção.

3. Estação TABA

Ambiente de Desenvolvimento de Software (ADS) é um sistema computacional que provê suporte para o desenvolvimento e manutenção de software e para o gerenciamento dessas atividades, contendo uma base de dados central e um conjunto de ferramentas de apoio [Travassos 1994]. Ambientes de Desenvolvimento de Software Orientados à Organização (ADSOrgs) foram definidos como sendo Ambientes de Desenvolvimento de Software que apóiam a gerência do conhecimento ao longo dos processos de desenvolvimento e manutenção de software [Villela 2004].

A Estação TABA, é um meta-ambiente capaz de gerar ADSOrgs, através de configuração e instanciação. O meta-ambiente é um ambiente que abriga um conjunto de programas que interagem com os usuários para definir interfaces, selecionar ferramentas e estabelecer os tipos de objetos que irão compor o ambiente de desenvolvimento específico. O objetivo da Estação TABA é auxiliar na definição, implementação e execução de ADS adequados a contextos específicos [Villela 2004] [Rocha *et al*. 1990].

Neste momento a Estação TABA permite configurar um ambiente específico para uma organização através de seu processo padrão e processos especializados para as diferentes abordagens de desenvolvimento utilizadas na organização (por exemplo, desenvolvimento OO ou desenvolvimento de sistemas baseados em conhecimento). Estes processos são finalmente instanciados para projetos específicos considerando-se suas particularidades (por exemplo, tamanho, requisitos de confiabilidade, etc) [Villela 2004]. Desta forma, os ambientes contemplados na Estação TABA podem ser definidos da seguinte maneira [Villela 2004]:

- **Meta-Ambiente**: ambiente que apóia a configuração de ambientes para organizações específicas;
- **Ambiente Configurado**: ambiente configurado a partir do meta-ambiente que apóia a instanciação de ADSOrg para projetos específicos;
- **ADSOrg**: ambiente de desenvolvimento de software instanciado a partir do Ambiente Configurado.

4. Processos de Verificação de Software

Para que a verificação de software seja realizada de forma organizada, disciplinada e seguindo um conjunto de atividades bem definidas, alguns processos de verificação de software foram definidos na literatura.

A norma internacional ISO/IEC 12207 (1995) define o processo de verificação como um dos processos de apoio. O processo tem como objetivo determinar se os produtos de software desenvolvidos em uma determinada atividade atendem completamente os requisitos ou condições impostas a eles nas atividades anteriores. Além disso, a norma define que, para a eficácia de custo e desempenho, a verificação deve ser integrada, o quanto antes, ao processo que a utiliza. Segundo esta norma, o processo de verificação inclui análise, revisão e teste. Esse processo é composto de duas atividades: (i) implementação do processo e (ii) verificação. A atividade de

implementação do processo consiste das seguintes tarefas: determinar se o projeto justifica um esforço de verificação, estabelecer um processo de verificação, determinar as atividades do ciclo de vida e os produtos de software que requerem verificação e desenvolver e documentar um plano de verificação. A segunda atividade do processo de verificação da ISO/IEC 12207 (1995) é composta pelas seguintes tarefas: verificação do contrato, verificação do processo, verificação dos requisitos, verificação de projeto, verificação do código, verificação da integração e verificação da documentação.

No CMMI (2002), a área de processo Verificação possui os seguintes objetivos: (i) preparação para verificação; (ii) realização de revisões por pares; (iii) verificação dos artefatos selecionados. O primeiro objetivo consiste em selecionar os artefatos para verificação, estabelecer um ambiente para verificação e estabelecer procedimentos e critérios para verificação. O segundo objetivo é composto pelas seguintes práticas: preparar as revisões por pares, conduzir as revisões por pares e analisar os dados das revisões por pares. O terceiro objetivo consiste em realizar a verificação e analisar os resultados da verificação e identificar ações corretivas.

Como se pode observar nas duas abordagens, deve-ser realizar, basicamente, duas atividades: (i) Planejamento da Verificação e (ii) Execução da Verificação.

Para realização deste trabalho o primeiro passo foi definir um processo baseado nas abordagens descritas acima. Além disso, como o Modelo Brasileiro de Melhoria de Processo de Software – mpsBr [mpsBr 2004] é fortemente baseado na ISO 12207 (1995) e no CMMI (2002), o processo definido também é aderente a esse modelo. Dessa forma, as empresas que utilizam a Estação TABA têm à sua disposição recursos (processos definidos e ferramentas que apóiam a execução desses processos) aumentando a possibilidade de uma avaliação bem sucedida em qualquer uma dessas abordagens.

5. Um Processo de Verificação de Software para Ambientes de Desenvolvimento de Software Orientados à Organização

A abordagem proposta para o processo de Verificação de Software envolve a execução de duas macro-atividades: (i) Planejamento da Verificação e (ii) Execução da Verificação. A Figura 1 ilustra as macro-atividades do processo de verificação.

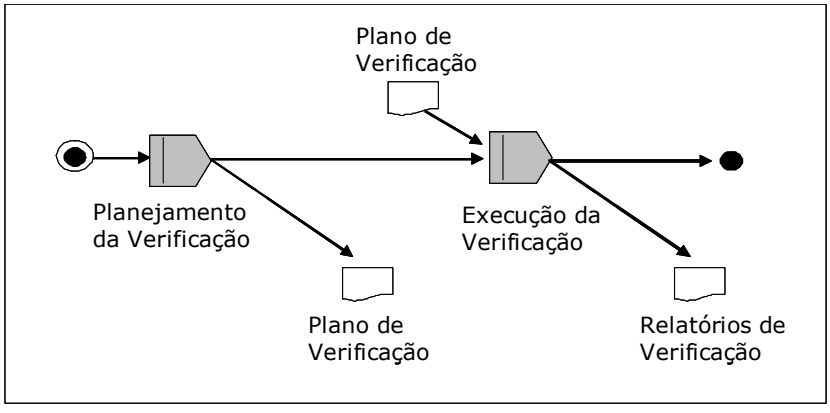

Figura 1. Macro-Atividades do Processo de Verificação

A primeira macro-atividade do processo é o **planejamento da verificação**. Essa macro-atividade pode ser dividida em duas atividades: Selecionar Artefatos para Verificação e Planejar a Verificação do Artefato. A Figura 2 ilustra as atividades que compõem o planejamento da verificação.

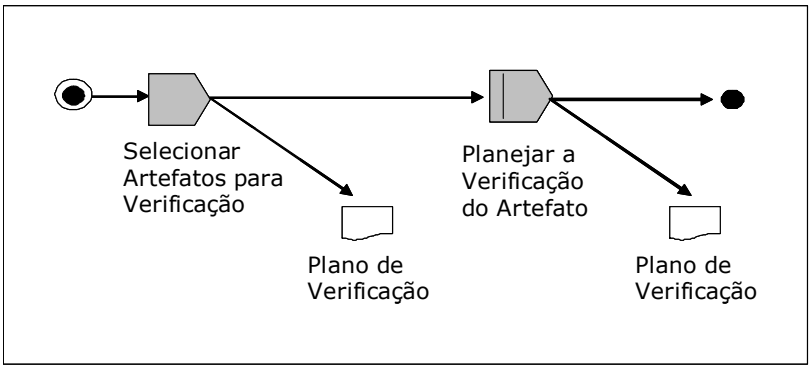

Figura 2. Macro-Atividade Planejamento da Verificação

A atividade **Selecionar Artefatos para Verificação** tem como objetivo identificar os artefatos que serão verificados ao longo do desenvolvimento. Para isso, os artefatos que serão produzidos devem ser analisados quanto às suas contribuições para o alcance dos objetivos e requisitos do projeto, considerando também os riscos do projeto. Para cada artefato selecionado, a atividade Planejar a Verificação do Artefato deve ser executada. A Figura 3 ilustra as sub-atividades que compõem a atividade Planejar a Verificação do Artefato.

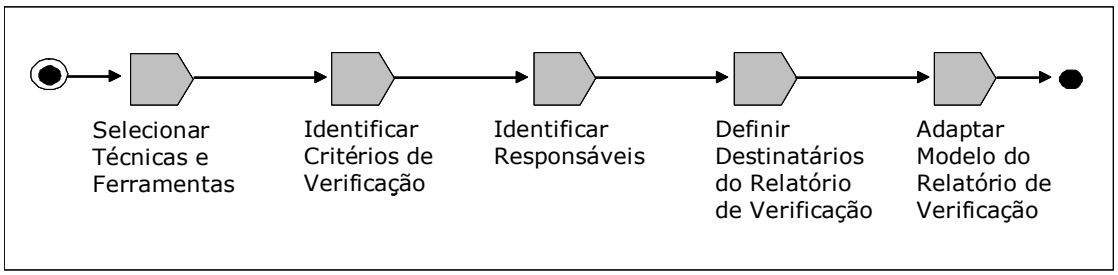

Figura 3. Atividade Planejar a Verificação do Artefato

A atividade **Planejar a Verificação do Artefato** é composta das seguintes sub-atividades:

- *Selecionar Técnicas e Ferramentas*: Nessa atividade as técnicas de verificação que estão disponíveis para uso e que serão aplicadas ao artefato devem ser selecionadas, bem como as ferramentas que serão usadas para apoiar a verificação. De acordo com as técnicas selecionadas, a verificação do artefato será realizada através de revisão por pares e/ou testes.
- *Identificar Critérios de Verificação*: Ao executar essa atividade, os critérios de verificação para o artefato em questão serão selecionados. Os critérios de verificação podem variar de acordo com o artefato a ser verificado.

- *Identificar Responsáveis*: O objetivo dessa atividade é definir as responsabilidades pela verificação do artefato, ou seja, as pessoas que devem acompanhar e participar da verificação do artefato.
- *Definir Destinatários do Relatório de Verificação*: Nessa atividade serão identificados os papéis e as pessoas que receberão o relatório de verificação do artefato.
- *Adaptar o Modelo do Relatório de Verificação*: O objetivo dessa atividade é possibilitar a adaptação do modelo do laudo de verificação e/ou do relatório de teste a serem gerados na verificação do artefato, de forma a torná-los mais adequados ao projeto.

Ao longo do planejamento da verificação, um plano de verificação contendo as informações do planejamento é produzido. Esse plano é a principal guia para a realização da próxima macro-atividade, a execução da verificação.

A macro-atividade **execução da verificação** pode ser dividida em duas atividades: Realizar Revisão por Pares e Realizar Testes. A verificação de um artefato deve ser executada através de pelo menos uma dessas duas atividades de acordo com o que foi planejado. A Figura 4 ilustra as atividades que compõem a execução da verificação.

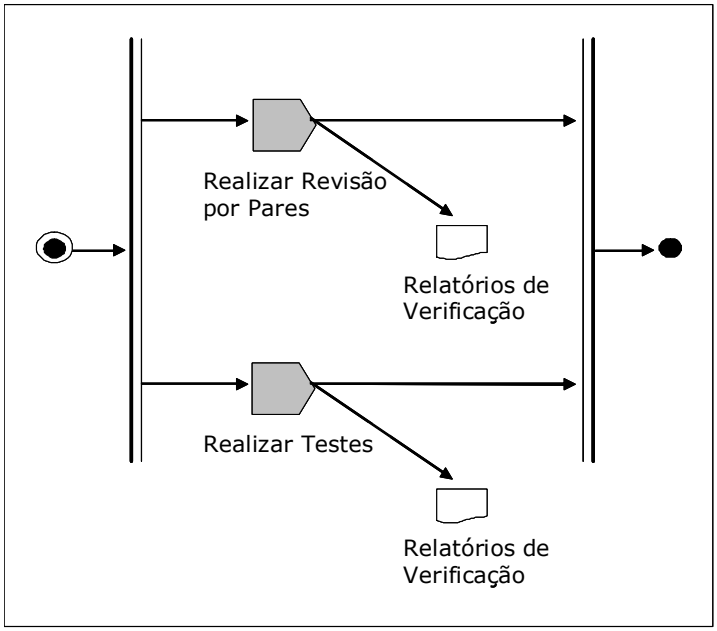

Figura 4. Macro-Atividade Execução da Verificação

Na atividade **Realizar Revisão por Pares** deve ser realizada a revisão por pares para verificação do artefato, conforme descrito no plano de verificação. Durante a execução dessa atividade deve ser produzido o relatório de verificação do artefato, neste caso, o laudo final da verificação do artefato. Esse laudo será encaminhado aos destinatários definidos no planejamento da verificação do artefato.

Na execução da atividade **Realizar Testes,** deve ser realizado o planejamento dos testes identificando aqueles que são mais adequados para verificação do artefato em

questão e, após esse planejamento, os testes devem ser realizados. Durante a realização dos testes os relatórios de teste devem ser produzidos. Esses relatórios serão encaminhados aos destinatários definidos no planejamento da verificação do artefato.

O processo descrito não é executado de forma seqüencial. A primeira atividade (*Selecionar Artefatos para Verificação*) deve ser executada durante o planejamento do projeto. As demais atividades devem ser executadas várias vezes ao longo do desenvolvimento do projeto, uma vez para cada artefato selecionado.

O processo de verificação é dependente do processo de desenvolvimento e por isso sua execução é integrada à execução do processo de desenvolvimento. Ao longo do processo de desenvolvimento, os artefatos que foram selecionados para verificação serão produzidos. Para cada um desses artefatos, a atividade Planejar a Verificação do Artefato deve ser executada antes da produção do artefato. Após a produção do mesmo, as atividades Realizar Revisão por Pares e/ou Realizar Testes devem ser executadas conforme o planejamento da verificação.

Para guiar o usuário na execução dessas atividades, elas serão incluídas no processo de desenvolvimento. Desta forma o processo de desenvolvimento será alterado dependendo da seleção dos artefatos para verificação e do planejamento da verificação de cada artefato. A Figura 5 apresenta um exemplo de execução do processo de verificação integrado ao processo de desenvolvimento.

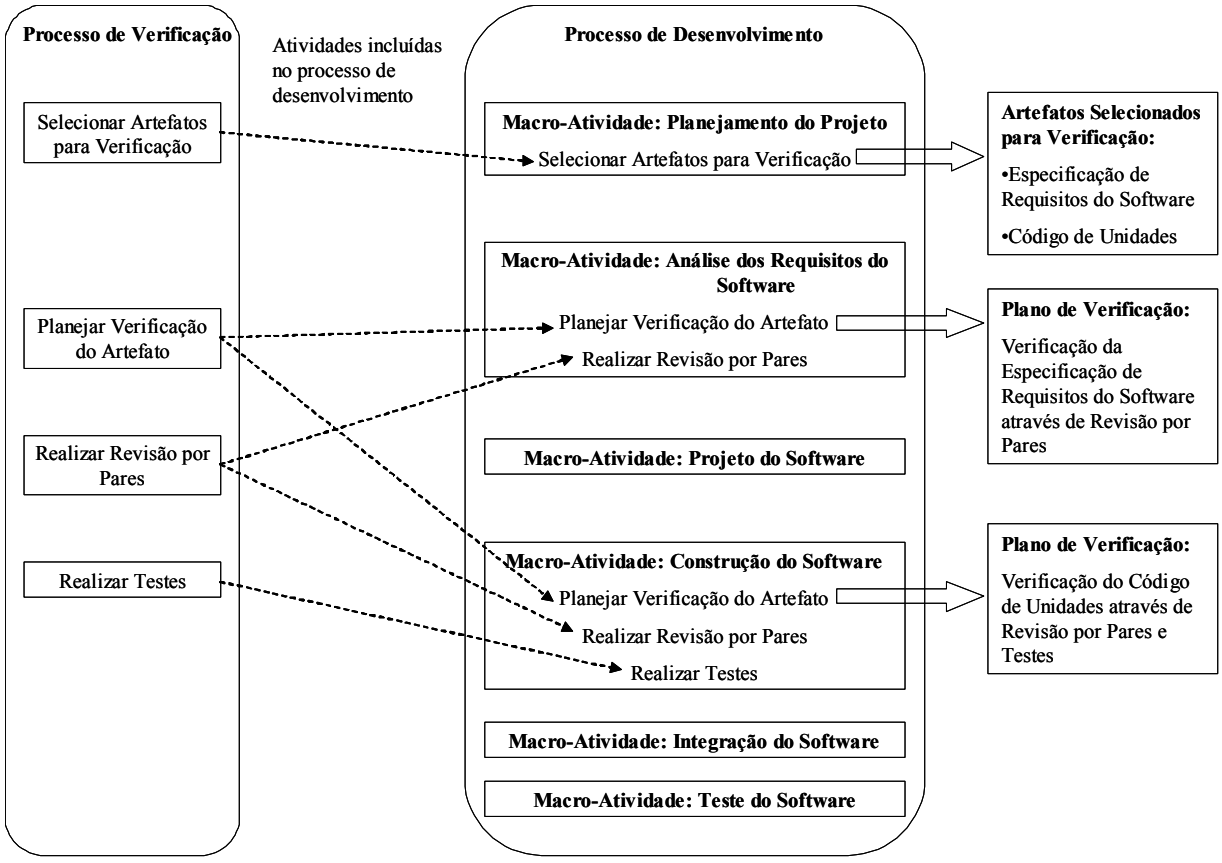

Figura 5. Processo de Verificação integrado ao Processo de Desenvolvimento

Na Tabela 2 são apresentados os responsáveis por cada atividade do processo. A Tabela 3 relaciona as atividades do processo definido com as práticas específicas da área de processo Verificação do CMMI (2002).

A próxima seção apresenta a abordagem de apoio definida para o processo de verificação, bem como a ferramenta *VerificationPlan* que apóia o planejamento do processo definido.

Tabela 2. Responsáveis pelas atividades do processo de Verificação

Atividade do Processo de Verificação	Gerente do Projeto	Equipe do Projeto
Selecionar Artefatos para Verificação	✓	
Planejar Verificação do Artefato	✓	
Realizar Revisão por Pares		✓
Realizar Testes		✓

Tabela 3. Relacionamento entre o Processo Definido e as práticas específicas da área de processo Verificação do CMMI

Prática Especifica \ Atividade do Processo	Selecionar Artefatos para Verificação	Planejar Verificação do Artefato	Realizar Revisão por Pares	Realizar Testes
SP 1.1 - Selecionar Produtos de Trabalho para Verificação	✓			
SP 1.2 - Estabelecer o Ambiente para Verificação		✓		
SP 1.3 - Estabelecer Procedimentos e Critérios para Verificação		✓	✓	
SP 2.1 - Preparar para Revisão por Pares			✓	
SP 2.2 - Conduzir Revisão por Pares			✓	
SP 2.3 – Analisar os Dados da Revisão por Pares			✓	
SP 3.1 – Realizar a Verificação			✓	✓
SP 3.2 - Analisar os Resultados da Verificação e Identificar Ações Corretivas			✓	✓

6. Abordagem de Apoio ao Processo de Verificação

Buscando-se apoiar a abordagem de verificação de software descrita na seção anterior, a ferramenta *VerificationPlan* foi definida e implementada. Essa ferramenta apóia todas as atividades do planejamento da verificação: seleção de artefatos para verificação e planejamento da verificação do artefato. A ferramenta é disponibilizada em um Ambiente de Desenvolvimento de Software Orientado a Organização (ADSOrg).

VerificationPlan baseia-se fundamentalmente no processo definido e guia o usuário durante a realização do planejamento da verificação. As atividades serão discutidas a seguir. Essa ferramenta possui dois módulos de execução de acordo com o usuário que a está executando. Isso ocorre porque algumas atividades só podem ser executadas por determinados usuários (como pode ser visto na Tabela 1). Os módulos são apresentados abaixo:

- *Módulo do Gerente do Projeto*: Selecionar Artefatos para Verificação, Planejar Verificação do Artefato.
- *Módulo da Equipe do Projeto*: Realizar Revisão por Pares e Realizar Testes.

A Figura 6 apresenta interface básica da ferramenta de apoio. No lado esquerdo pode-se visualizar o processo de verificação que guia a ferramenta e no lado direito da tela identifica-se a atividade que está sendo realizada pelo usuário. Além disso, é possível realizar a busca e o registro de conhecimento no que diz respeito às atividades do processo através da interação com uma ferramenta de Aquisição de Conhecimento – *Acknowledge* [Montoni *et al.* 2004].

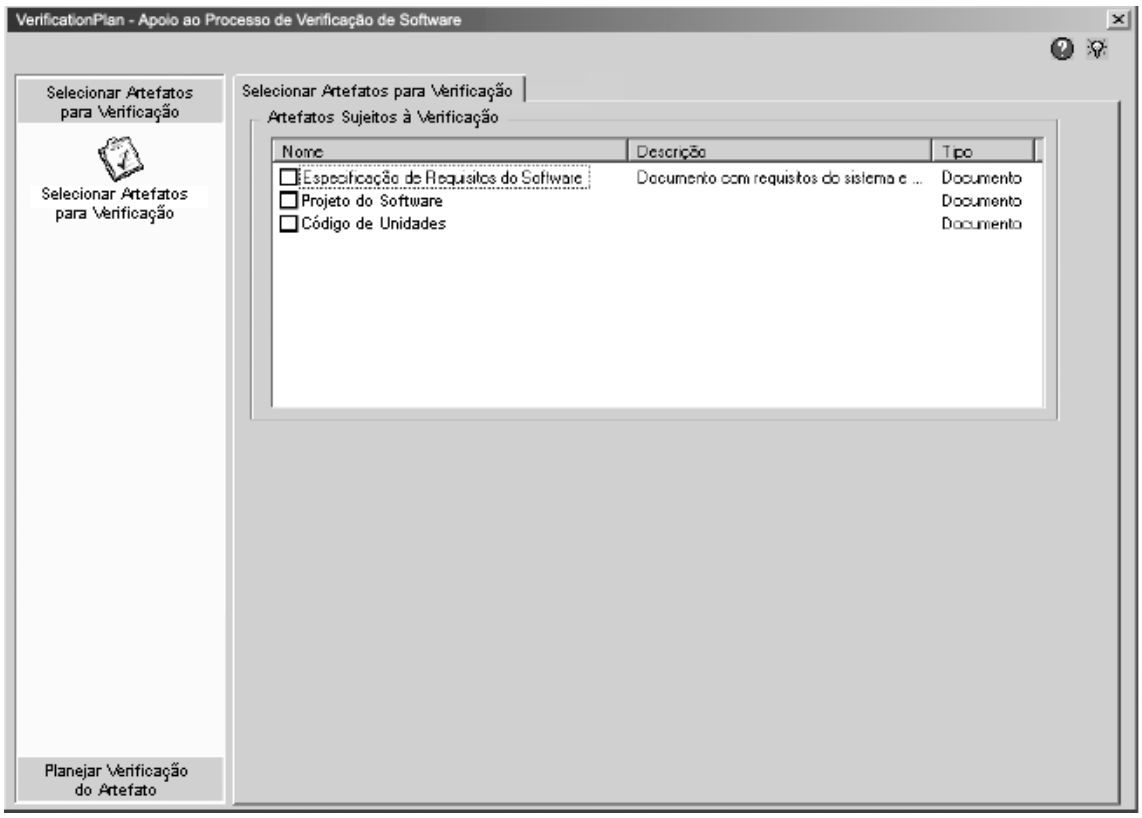

Figura 6. Interface da Ferramenta *VerificationPlan*

A Figura 6 ilustra a atividade *Selecionar Artefatos para Verificação*, na qual o gerente do projeto irá identificar os artefatos que serão verificados ao longo do projeto. Ao configurar um ADSOrg para uma organização, todos os artefatos sujeitos à verificação são identificados. Nos ADSOrgs, a ferramenta *VerificationPlan* é executada e então apresenta os artefatos identificados como sujeitos à verificação e que são

produzidos pelas atividades do processo de desenvolvimento do projeto em questão. O gerente pode, então, selecionar os artefatos que deseja verificar no projeto específico.

Na atividade *Planejar Verificação do Artefato*, que é composta de cinco sub-atividades, cada uma das sub-atividades é apoiada. Em todas as sub-atividades, o artefato cuja verificação está sendo planejada é identificado. A tela apresentada na Figura 7 corresponde à sub-atividade *Selecionar Técnicas e Ferramentas*. Nessa tela, uma lista de técnicas associadas ao artefato identificado é apresentada. Essa lista é cadastrada no Ambiente Configurado e é válida para todos os ADSOrgs instanciados a partir desse Ambiente Configurado. Além das técnicas, as ferramentas disponíveis para apoiar cada uma das técnicas também são cadastradas no Ambiente Configurado e, da mesma forma, são apresentadas na ferramenta VerificationPlan.

A sub-atividade *Identificar Critérios de Verificação* é apoiada pela ferramenta, que apresenta uma lista dos critérios de verificação para o artefato. Tais critérios também são cadastrados no Ambiente Configurado e usados nos ADSOrgs.

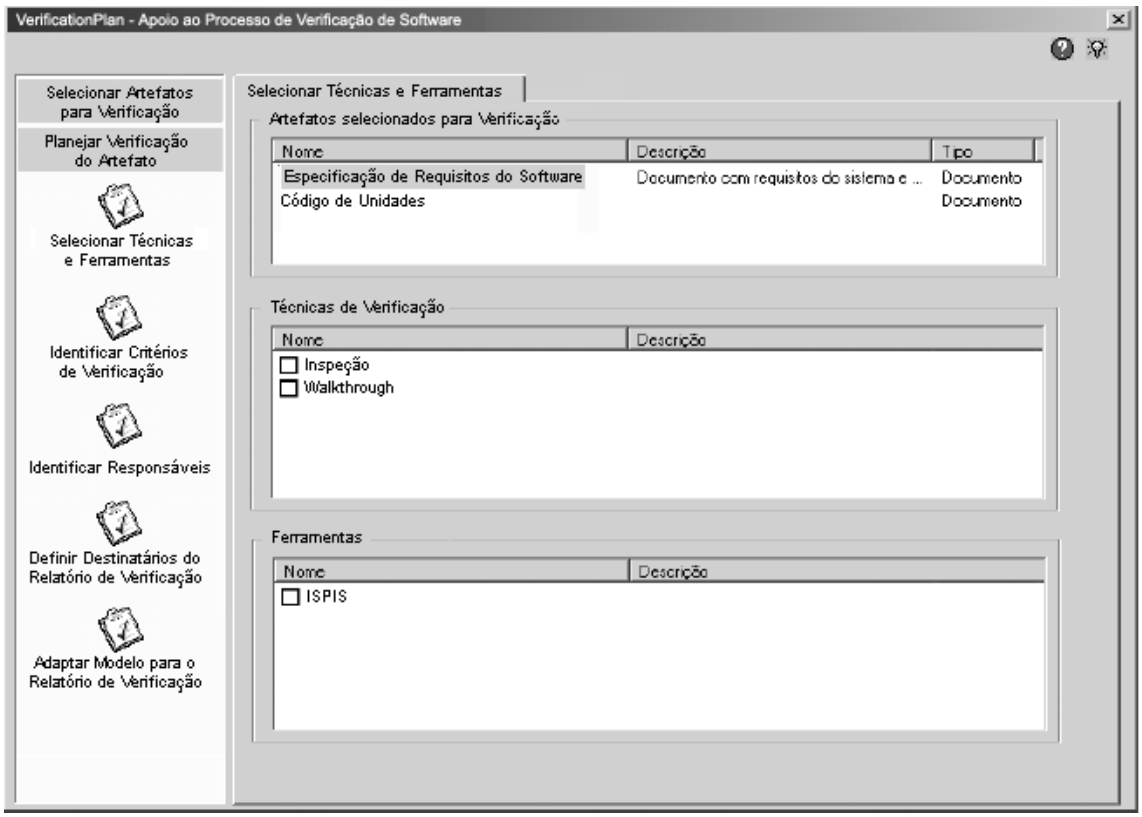

Figura 7. Atividade Selecionar Técnicas e Ferramentas

Na sub-atividade *Identificar Responsáveis*, a ferramenta apresenta as pessoas que fazem parte da equipe do projeto em questão e permite a identificação de alguns como responsáveis pela verificação daquele artefato.

Na sub-atividade *Definir Destinatários do Relatório de Verificação*, a ferramenta *VerificationPlan* apresenta os papéis existentes no projeto e permite a seleção daqueles que devem receber os relatórios de verificação. Para cada papel selecionado é possível identificar a pessoa que exerce aquele papel no projeto e que

deve receber os relatórios. Ao final da verificação, o ADSOrg envia por e-mail os relatórios para os destinatários identificados nessa atividade.

Finalmente, na sub-atividade Adaptar Modelo do Relatório de Verificação, a ferramenta apresenta os modelos definidos, especificamente para o artefato em questão, para o laudo de verificação e para o relatório de teste. É possível modificar esses modelos para o projeto, de acordo com as necessidades.

A qualquer momento durante o planejamento da verificação do artefato, é possível visualizar o plano de verificação gerado com as informações do planejamento de cada artefato.

7. Considerações Finais

Este artigo apresentou uma abordagem para a verificação de software fundamentada no conceito de Ambientes de Desenvolvimento de Software Orientados à Organização. Essa abordagem é baseada nas abordagens de verificação de software propostas pela ISO/IEC 12207 (1995) e pela área de verificação do CMMI (2002) e suporta as atividades do nível de maturidade 3 da Área de Verificação do CMMI (2002). O processo de verificação de software proposto foi descrito, assim como a ferramenta *VerificationPlan*, implementada com o intuito de apoiá-lo. Essa ferramenta está presente nos ADSOrgs instanciados pela Estação TABA.

A interface entre a ferramenta *VerificationPlan* e a ferramenta de Aquisição do Conhecimento *Acknowledge* [Montoni *et al.* 2004] permite que o conhecimento sobre verificação de software acumulado pela organização e relevante ao desenvolvimento de software seja consultado e que novos conhecimentos adquiridos sejam armazenados.

As perspectivas futuras deste trabalho incluem o uso do processo e do apoio ferramental propostos por algumas empresas com o intuito de caracterizar o uso desta abordagem. O TABA vem sendo utilizado em diversas empresas e já é possível observar bons resultados disso, como, por exemplo, uma avaliação bem sucedida do CMMI nível 2 em uma das empresas [Nunes *et al.* 2005]. O processo e a ferramenta aqui propostos serão também utilizados pelas empresas em breve. Com isso, espera-se contribuir para que o TABA auxilie as empresas também na obtenção do CMMI nível 3 e será possível caracterizar a aplicabilidade desta proposta na prática São ainda perspectivas futuras, a definição e implementação de ferramentas que auxiliem a realização das revisões por pares e a realização dos testes.

Referências Bibliográficas

Andersson, C. (2003) “Exploring the Software Verification and Validation Process with Focus on Efficient Fault Detection”, Licentiate Thesis, Lund Institute of Technology (LTH), Lund University, Sweden.

Berling, T., Runeson, P. (2003) “Evaluation of Perspective Based Review Method Applied in an Industrial Setting”, IEEE Proceedings Software, vol. 150, n° 3, pp.177-184.

CMU/SEI (2002) "Capability Maturity Model Integration (CMMI)", Version 1.1 – Staged Representation, Carnegie Mellon University, Software Engineering Institute, Pittsburgh.

Golubic, S. (2003) "On Software Quality Verification in the Object-Oriented Development Environment", 7th International Conference on Telecommunications – ConTEL, pp.557-563, Zagreb, Croatia.

Humphrey, W. S. (1995) "A Discipline for Software Engineering. Reading", MA Addison-Wesley Publishing Company.

ISO/IEC 12207 (1995) "Information Technology – Software Life-Cycle Processes".

Juristo, N., Moreno, A. M., Vegas, S. (2003) "Limitations of Empirical Testing Technique Knowledge", Lecture Notes on Empirical Software Engineering, Series on Software Engineering and Knowledge Engineering - Vol. 12.

Kan, S. H. (2003) "Metrics and Models in Software Quality Engineering", Addison-Wesley Publishing Company.

Laitenberger, O., Vegas, S., Ciolkowoski, M. (2002) "The State of the Practice of Review and Inspection Technologies in Germany", Tech Report Number: ViSEK/011/E.

Montoni, M., Miranda, R., Rocha, A. R., Travassos, G. H. (2004) "Knowledge Acquisition and Communities of Practice: An Approach to Convert Individual Knowledge into Multi-Organizational Knowledge", Workshop Learning Software Organization, Banff.

mpsBr - Modelo Brasileiro de Melhoria de Processo de Software (2004) "Apresentação do Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira", XXX Conferência Latinoamericana de Informática (CLEI2004), Arequipa, Peru.

Myers, G. J. (1979) "The Art of Software Testing", Wiley.

Nunes, E. D., Silva, R., Rocha, A. R., Natali, A. C., Santos, G. (2005) "Uma Abordagem para Implantação de Processos de Software com ISO 9001 e CMMI", IV Simpósio Brasileiro de Qualidade de Software, Porto Alegre – RS.

Pfleeger, S. L. (2004) "Software Engineering – Theory and Practice", Nova Jersey, Prentice-Hall.

Porter, A. A., Votta, L. G. Jr., Basili, V. R. (1995) "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment", IEEE Transactions on Software Engineering, Vol. 21, No. 6, pp. 563-575.

Rocha, A. R. C., Aguiar, T. C., Souza, J. M. (1990) "TABA: A Heuristic Workstation for Software development", In: Proceedings of COMPEURO 90, Tel Aviv, Israel.

Rocha, A. R. C., Maldonado, J. C., Weber, K. C. (2001) "Qualidade de Software – Teoria e Prática", São Paulo, Prentice Hall.

Silva, L. F. S., Travassos, G. H. 2003, "Apoio Ferramental para Aplicação de Técnicas de Leitura Baseada em Perspectiva (PBR)", Workshop de Teses – XVII Simpósio Brasileiro de Engenharia de Software, Manaus – AM.

Singh, H., Bawa, H. S. (1995) “Management of Effective Verification and Validation”, IEEE Engineering Management Conference, pp.204-207.

Thelin, T. (2002) “Empirical Evaluations of Usage-Based Reading and Fault Content Estimation for Software Inspections”, Doctoral Thesis, Lund University, Sweden.

Travassos, G. H. (1994) “O Modelo de Integração de Ferramentas da Estação TABA”, Doctoral Thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

Villela, K. (2004) “Definição e Construção de Ambientes de Desenvolvimento de Software Orientados a Organização”, Doctoral Thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.