

ADOPTION METRICS FOR VALIDATION OF SOFTWARE DEVELOPMENT PROCESS

T. M. R. Dias¹, P. M. Dias¹, G. F. Moita², B. Ferreira³

¹Instituto de Ensino Superior e Pesquisa – INESP, Fundação Educacional de Divinópolis – FUNEDI/UEMG

²Centro federal de Educação Tecnológica de Minas Gerais – CEFET-MG

³Instituto Federal Minas Gerais - IFMG

Abstract. *A process or methodology of software development is essential for creating quality systems. But for the institutions to make the most of their benefits, the process must be valid, that is, ensuring that participants are guided accurately and efficiently during the life cycle of the software. Some studies compare the formal model of the process with its execution, if the variance of the comparison is acceptable, the process is considered valid. But existing techniques do not consider issues such as satisfaction and efficiency. This paper proposes to adapt the software quality standards ISO9126 and ISO 14598 dealing with satisfaction metrics to use the software to measure the satisfaction of using the software development processes. Thus, together with comparisons of discrepancies, facilitated the validation of the development process including the variable of satisfaction of use.*

Keywords: *Validation Process, Satisfaction of Use, Software Development Process.*

Resumo: *Um processo ou metodologia de desenvolvimento de software é essencial para a criação de sistemas com qualidade. Mas para que as instituições possam aproveitar ao máximo seus benefícios, o processo deve ser válido, ou seja, garantir que os participantes sejam guiados de forma correta e eficiente durante o ciclo de vida do software. Alguns trabalhos comparam o modelo formal do processo com a sua execução, se a discrepância da comparação for aceitável, o processo é considerado válido. Mas as técnicas existentes não consideram questões como satisfação e eficiência. Este trabalho propõe adaptar as normas de qualidade de software ISO 9126 e ISO 14598 que tratam das métricas de satisfação de uso do software para medir a satisfação de uso dos processos de desenvolvimento de softwares. Desta forma, em conjunto com as comparações de discrepâncias, auxiliaria na validação do processo de desenvolvimento incluindo a variável de satisfação de uso.*

Palavras Chave: *Validação de Processo, Satisfação de Uso, Processo de desenvolvimento de Software.*

1. INTRODUÇÃO

Não é novidade para as empresas desenvolvedoras de sistemas que a busca por qualidade está diretamente ligada aos benefícios proporcionados pelos processos ou metodologia de desenvolvimento de software. O software não é mais o produto único e sim, parte de um processo que se preocupa com a independência de indivíduos, organização, aumento de produtividade e facilidade de manutenção.

A literatura apresenta vários processos de desenvolvimento que podem ser classificados de acordo com modelos preestabelecidos, mas geralmente, as empresas são obrigadas a customizar ou criar suas próprias metodologias, o que pode gerar certa insegurança no uso das etapas atividades e papéis propostos. Para amenizar as desconfiças iniciais, alguns autores propõem que a criação de um processo siga passos predeterminados, entre os quais, podemos citar a Validação do Processo Inicial e a etapa de Melhoraria de Processo.

A Validação do Processo Inicial é um passo importante e pode responder a uma preocupação natural da gerência de projetos – a garantia que o processo de desenvolvimento guia os participantes de forma correta e eficiente durante a criação do software. Burocracia excessiva ou uma orientação ambígua podem atrapalhar, ao invés de orientar, o desenvolvimento no ciclo de criação do produto.

Várias pesquisas vêm sendo propostas para a validação de processos, mas todas têm como ponto em comum, a comparação entre a execução do processo com o modelo formal proposto. Assim, a pesquisa destes autores foca em explorar as diferenças entre o modelo formal do processo e a prática, com o intuito de fornecer idéias úteis para melhorar os processos de engenharia de software e seus resultados práticos, sendo essa comparação essencial para detectar discrepâncias.

Mas a Validação é conceitualmente complexa, pois se refere a uma larga escala de questões, muitas vezes subjetivas. A execução do processo pode ser seguida como o modelo formal propõe e, mesmo assim, ser burocrática ou pouco esclarecedora. Com o intuito em minimizar a subjetividade, esse trabalho propõe adaptar as normas de qualidade de software ISO 9126 e ISO 14598 que tratam das métricas de satisfação de uso do software para medir a satisfação de uso dos processos de desenvolvimento de software. Esta adaptação será aplicada em conjunto com as técnicas de validação existentes, com o objetivo de melhorar os resultados obtidos durante a validação inicial do processo de desenvolvimento sob criação.

2. VALIDAÇÃO DE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Validar um processo de desenvolvimento não é uma tarefa simples. Um dos problemas é que a validação é conceitualmente complexa, por se referir a uma larga escala de questões, muitas vezes subjetivas. Outro problema é a ausência de diretrizes, técnicas ou padrões consolidados e documentados para auxiliar na validação de processos.

Cook e Wolf [1] e Moor e Delugach [7] adotam como princípio para a validação de processos a comparação entre modelo e prática, mas o fazem de formas distintas. Moor e Delugach [7] adotam um formalismo baseado na Teoria dos Grafos, de forma que o modelo e a prática sejam representados e comparados. O método adotado por Cook e Wolf [1] para medir

as diferenças entre o modelo e a prática é a Edição de String, onde o número de inserções, deleções e substituições simbólicas (tokens) necessárias para transformar uma string (seqüência de eventos do processo/prática) em outra são analisadas.

Mas, segundo Ferreira e Moita [2], os métodos apresentados pelos autores citados acima pagam um custo por utilizar conceitos de Teoria dos Grafos ou Edição de String, pois, podem inviabilizar a validação de processos de desenvolvimento ágeis, que pregam a flexibilidade e a facilidade de uso. Novos processos como o PESC, propostos por Pereira Jr [8] são criados com o intuito de prover qualidade no desenvolvimento de software sem a aplicação de grande quantidade de recursos humanos ou financeiros. Se a técnica de validação do processo não seguir esse intuito, ela pode inviabilizar sua utilização.

Ferreira e Moita [3] apontam que a técnica propostas por Cook e Wolf [1] e Moor e Delugach [7] não orientam como todo o processo de validação deve ser executado e apontam que as propostas são complexas e dependentes do contexto em que foram aplicadas, concluindo que é notória a ausência de uma técnica que tenha a preocupação de ser genérica, simples e extensível. Os autores ainda propõem uma técnica de Validação de Processo por Inspeção denominada VProcInsp.

2.1 VProcInsp

O VProcInsp é uma proposta de Ferreira e Moita [4] baseada nos conceitos propostos por Sauer et al. [11] para a inspeção de software e adaptação dos documentos propostos por Kalinowski [10] em sua infraestrutura computacional de apoio à inspeção de software que tem o objetivo de realizar revisões na prática de forma mais sistemática e explorar melhor o seu potencial, utilizando dados históricos e informações experimentalmente avaliadas sobre inspeções de software. A Figura 1 mostra o modelo proposto.

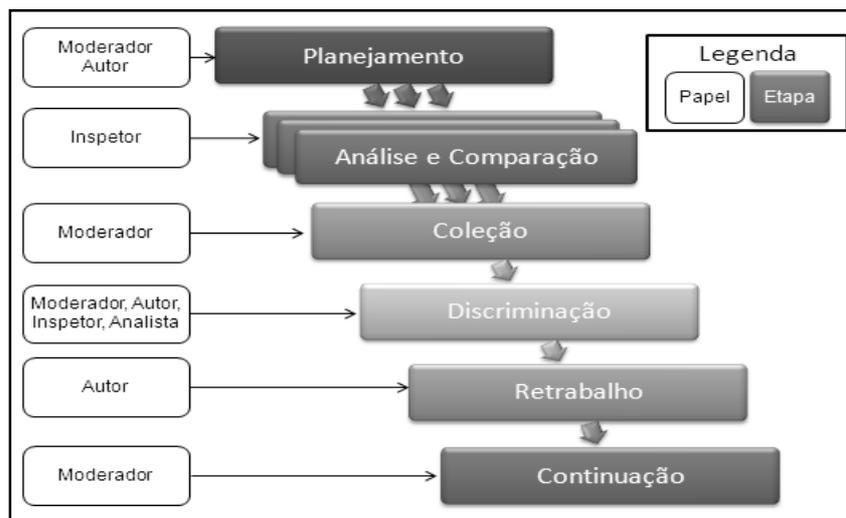


Figura 1. Modelo proposto pelo VProcInsp (Adaptado de [11]).

A proposta de criação do VProcInsp seguiu os passos descritos por Laitenberger e Atkinson (1998) e definiu os tipos do artefatos de software/processo a serem avaliados; a forma como a equipe de profissionais será definida para realizar a inspeção; a técnica de avaliação utilizada e o processo de execução seguido na inspeção.

Na fase de Planejamento, o primeiro passo é identificar um profissional que exercerá o papel do Moderador da inspeção. O Planejamento é similar a uma Gerência de Projetos, pois nessa atividade o Moderador define a equipe, prazos, custos, entre outras funções cabíveis à gerência.

Durante essa fase, o Moderador deve definir a abrangência em que a inspeção de validação do processo vai ser realizada (por etapa, por iteração, ao final do ciclo de desenvolvimento, entre outros), além de informações como datas e tamanho da equipe. O Moderador define também os participantes, com base na experiência de desenvolvimento, conhecimento do domínio e experiência com validação de processos.

A principal atividade do Planejamento é a configuração do checklist para a inspeção de validação. Para configurar o checklist, o Moderador conta com a documentação ou modelo formal do processo sob validação e com o apoio do Autor do processo. Em uma reunião, os profissionais que assumiram os papéis de Moderador e Autor criam os checklists seguindo diretrizes propostas na atividade de Definição do Checklist.

Na fase de Análise e Comparação, os inspetores, individualmente analisam os artefatos e os comparam com o modelo formal do processo. Eles são guiados pelos checklists, e identificam discrepâncias entre os dois conjuntos de documentos.

Na etapa de Coleção, o Moderador da inspeção tem acesso a todas as listas de discrepâncias através dos Relatórios de Discrepâncias produzidos pelos Inspectores na atividade de Análise e Comparação. O Moderador poderá, então, selecionar discrepâncias destas listas e as descartar ou classificar como replicadas, caso haja mais de uma discrepância representando o mesmo problema. Quando uma discrepância é descartada, ela é retirada das próximas atividades do VProcInsp.

O Moderador, o Autor e os Inspectores têm participação na Discriminação. Durante esta etapa, as discrepâncias são tratadas como tópicos de discussão. Cada participante pode acrescentar seus comentários relativos a cada uma das discrepâncias, que fica disponível, como tópico de discussão enquanto o Moderador e o Autor não decidem se o item representa realmente uma discrepância ou não.

Por fim, se de fato a discrepância for constatada, o Moderador deve marcar no Relatório de Discrepâncias que ela foi realmente aceita e devem-se atualizar as estatísticas do inspetor responsável por essa discrepância.

É importante ressaltar que a solução para as discrepâncias não são discutidas nessa etapa. Esta é uma tarefa do Autor do processo. Logo, cabe ao Autor gerar, da forma que lhe convier, a documentação para que alterações no processo sejam feitas.

No Retrabalho, o Autor corrige as discrepâncias detectadas durante a inspeção. Já na fase de Continuação, o material corrigido pelos autores é repassado para o Moderador, que tem a liberdade de decidir se uma nova inspeção deve ocorrer ou não.

3. QUALIDADE DE SOFTWARE

No estudo de caso apresentado por Ferreira e Moita [5] o VProcInsp foi aplicado em uma das maiores empresas brasileiras de desenvolvimento de Software. O estudo levantou discrepâncias no processo e mostrou resultados interessantes segundo a equipe de engenharia de software da empresa. Mas, apesar de mostrar indícios que a técnica pode descobrir informações sobre o processo comparando o modelo formal com a execução, várias questões ainda podem ser abordadas.

Isto se deve ao fato que validar um produto, processo ou fenômeno é conceitualmente complexo e se refere a muitas questões subjetivas: (1) Um processo pode não ser seguido e produzir um resultado satisfatório, ou a prática e o processo podem ser coincidentes, mas o produto final não corresponder às expectativas. Logo, percebe-se a necessidade de abordar outras variáveis no processo de validação, como analisar a satisfação dos desenvolvedores, usuários e da gerência da empresa produtora de software ou, (2) a Eficiência, aonde recurso e tempo envolvido durante o processo são analisados se são compatíveis com o nível de desempenho esperado.

A análise de satisfação e eficiência são características já estudadas pela área de Engenharia de Software, mas com o objetivo de buscar a qualidade do produto. Logo, passa a ser uma alternativa para a validação de processo conhecer como essas características são alcançadas no produto para que seja analisada uma possível adaptação para avaliar a satisfação e eficiência do processo. Isso fornecerá outro ponto de vista na validação.

O principal objetivo da Engenharia de Software é produzir software de qualidade. O termo qualidade é definido por Pressman [9] como a conformidade de requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvidos. A avaliação dos produtos de software segundo normas têm sido uma das formas empregadas por diversas organizações.

Segundo Koscianski et al. [11], normatização é a ação de aplicar regras estabelecidas e executar uma atividade de maneira ordenada. Objetiva-se com a utilização de normas no desenvolvimento e teste de software propiciar benefícios quantitativos como redução de custo, tempo e erros e qualitativos como adequação, facilidade de uso e uma melhor percepção pelos usuários, através de uma linguagem como estabelecida entre fornecedores e consumidores. Segundo esses autores é importante destacar também que o uso de normas auxilia na obediência ao Código do Consumidor e a maior competitividade internacional. As normas podem ser internacionais, regionais, nacionais e organizacionais em função da sua área de aplicação e os organismos internacionais mais importantes para o setor de software são a ISSO e a IEC.

A *International Organization for Standardization* – ISO é uma organização não-governamental estabelecida em 1947. Sua missão é promover o desenvolvimento da normatização e atividades relacionadas a nível mundial. O seu trabalho resulta em acordos entre países e que são publicados como normas internacionais.

A *International Electrotechnical Commission* – IEC, fundada em 1906 é a organização mundial que publica as normas internacionais relacionadas com eletricidade, eletrônica e áreas relacionadas. Conta com a participação de mais de 50 países.

Existe uma série de padrões, caracterizados por ser um conjunto de documentos que trabalham com sistemas de qualidade e abrange diversas áreas como as normas ISO 9000, as quais podem ser subdivididas em:

ISO 9001 – Norma que visa à garantia de qualidade no desenvolvimento, aplicada as organizações que projetam/desenvolver um produto/serviço, abrange o ciclo de vida completo destes.

ISO 9002 – Norma que abrange a qualidade na fabricação, montagem e suporte.

ISO 9003 – Esta norma é utilizada em organizações que fazem uso de teste de produto, abrangendo a qualidade na inspeção final e testes.

ISO 9004 – Responsável por estabelecer diretrizes para melhoria do desempenho.

Por sua importância, a ISO e a IEC criam normas específicas ao desenvolvimento de sistemas. Essas normas são referenciadas pelos números 9126 e 14598.

3.1 Normas ISO/IEC 9126 e 14598

As famílias de normas ISO/IEC 9126 (partes 1 a 4) e 14598 (1 a 6) descrevem um modelo de qualidade, um processo de avaliação e alguns exemplos de métricas que podem ser utilizadas por organizações que pretendam fazer avaliação de produto de software. A ISO 14598 refere-se ao conjunto de planos para avaliação do produto de software, utilizada em parceria (suporte) com a ISO/IEC 9126 que define métricas, ou seja, medições para atributos quantificáveis internos ou externos do software. A norma ISO 14598 é dividida em:

ISO/IEC 14598-1 – Visão Geral: Apresenta definições técnicas, estrutura do funcionamento das normas, requisitos para métodos de medição, como intuito de avaliar a qualidade do produto de software.

ISO/IEC 14598-2 – Planejamento/Gerenciamento: Estabelece os requisitos, orientações para a função de suporte do processo de avaliação dos produtos de software, determinando quais as métricas serão utilizadas para a avaliação.

ISO/IEC 14598-3 - Processo para equipe de desenvolvedores: É utilizada durante o desenvolvimento e manutenção do software, sendo realizadas as medições e avaliações no produto intermediário para ver antecipadamente a qualidade do produto final.

ISO/IEC 14598-4 – Processo para compradores: Determina as atividades para aceitação, seleção e aquisição dos produtos de software.

ISO/IEC 14598-5 – Processos para avaliadores: Determina as atividades necessárias para analisar os requisitos de avaliação, guiando-os para a avaliação do produto de software.

ISO/IEC 14598-6 – Módulo de Avaliação: Descreve os módulos de avaliação (conjunto de itens onde se encontram a tecnologia utilizada para avaliação das características de um software), determinando a estrutura e o conteúdo da documentação.

A Figura 2 mostra todo o processo de avaliação proposto pela norma 14598-1. Os números entre parêntese ao longo da Seção 4 correspondem aos itens da norma 14598-1 que tratam do assunto.

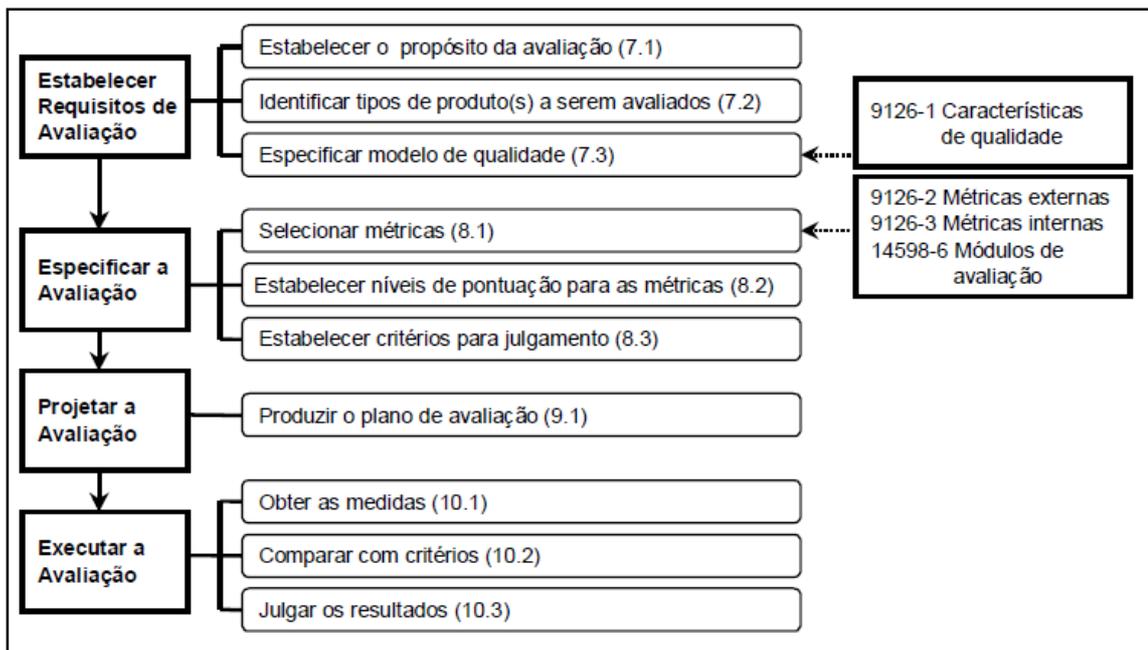


Figura 2 – Processo de avaliação do produto [11]

O processo de avaliação do produto pode ser interpretado por três perspectivas diferentes: (1) Desenvolvedores de softwares que pretendem melhorar seus produtos; (2) Pessoas ou organizações que pretendem utilizar a avaliação de qualidade como um processo de seleção ou, (3) organizações que executam avaliações independentes de produtos de software, o qual é o ponto de vista mais relevante para técnicas de validação de processo que tenha como objetivos ser genérica, ou seja, aplicável a várias instituições distintas.

4. DISCUSSÃO

Assim como a avaliação do produto (Figura 2), na validação de processos é necessário Estabelecer Requisitos de Avaliação com o propósito de dar subsídios para dizer que o processo de desenvolvimento é bem visto por seus usuários (7.1) e, será aplicado durante todo o ciclo de uso do mesmo (7.2). Equivalente à validação do produto, também é necessário especificar o modelo de qualidade adotado (7.3), que significa levantar as características internas (funcionalidade, confiabilidade, usabilidade, eficiência e manutenibilidade), ou seja, uma visão interna ao software ou características externas conhecidas como qualidade de uso (efetividade, produtividade, segurança e satisfação) que retrata a visão do usuário do sistema. Neste caso, como já levantado por Ferreira e Moita [4], deverá ser avaliada a eficiência e satisfação do processo.

É necessário Especificar a Avaliação, que se inicia selecionando as métricas. Todo atributo interno quantificável do software e todo atributo externo quantificável do software interagindo com seu ambiente e que se correlacione com uma característica, pode ser definido como uma métrica levando em consideração pontos como significância, custo e complexidade, repetibilidade, entre outras (8.1). Deve-se em seguida, estabelecer os níveis de pontuação das métricas. O resultado, isto é, o valor medido é mapeado em uma escala. A Figura 3 mostra um exemplo de níveis de satisfação (8.2). Estes níveis servem como critérios de jul-

gamento (8.3).

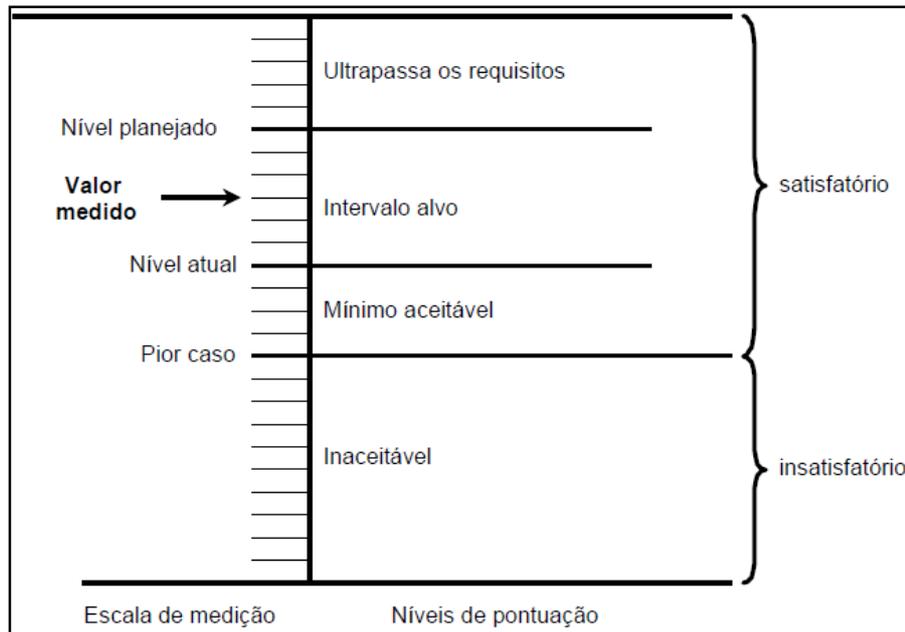


Figura 3 – Níveis de pontuação para métricas [11]

A próxima etapa é Projetar a Avaliação. Neste ponto devem ser especificadas ações para coleta dos dados a serem executadas para obter valores reais para cada métrica, isto inclui a especificação de cronogramas, responsabilidades e o uso de ferramentas de coleta de dados e de análise. Ao final, uma entrevista com os participantes do processo e o cruzamento de dados como a data prevista de entrega e a data realizada de cada artefato proposto pelo processo podem respectivamente analisar satisfação e eficiência.

Na Execução da Avaliação as métricas selecionadas são aplicadas ao processo. Como resultado obtém-se os valores nas escalas das métricas (10.1). O valor medido é comparado com os critérios predeterminados (10.2) e finalmente julgados (10.4).

Todo o processo de avaliação é descrito por Koscianski et al. [11] como um procedimento passo-a-passo, de tal forma que os requisitos de avaliação possam ser expressos quantitativamente e mostram resultados compreensíveis, aceitáveis e confiáveis.

5. CONCLUSÃO

Com o aumento da demanda e da complexidade dos sistemas, os processos de desenvolvimento de software vêm sendo considerados fundamentais para que um produto de qualidade seja obtido. Atualmente, empresas de todos os portes e instituições científicas fazem uso dos benefícios adquiridos pelo uso desses processos. Contudo, os vários processos de desenvolvimento vêm sendo customizados e outros propostos para atender as necessidades de cada meio produtor de software, como é o caso do processo da RM Sistemas e do PESC. Isto faz surgir a necessidade de validar os processos para garantir que são tangíveis de serem seguidos, descartando burocracias desnecessárias ou constatando a ausência de uma orientação

mais metódica.

Durante a pesquisa foram identificadas, através de uma revisão sistemática, algumas técnicas de validação de processos descritas na literatura. Com a caracterização dessas técnicas, foi possível identificar seus benefícios e limitações bem como, sugerir uma técnica de validação que permite minimizar as limitações e, suprir a ausência constatada de técnicas de validação de processos bem documentadas e genéricas.

Assim, pode-se concluir que:

1) Dizer que um processo é seguido à risca não quer dizer que ele irá produzir softwares corretos ou com o menor custo. O processo pode conter burocracias desnecessárias e submeter os envolvidos a executar tarefas sem necessidade ou encobrir outras que são necessárias. Logo, uma possível saída é analisar outras variáveis subjetivas ou passíveis de medições como eficiência, satisfação, usabilidade, confiabilidade.

2) Assim como Ferreira Moita [5] usaram os princípios e técnicas de Verificação e Validação de Software para criar uma técnica de Validação de Processo, o uso de normas ISO/IEC de satisfação de uso e eficiência de software dá indícios que podem ser adaptadas e servirem como mais um parâmetro de comparação para a validação de processos. Ou seja, a idéia é usar diretrizes, conceitos e atividades utilizadas como métricas de software para validar processos de software.

6. REFERÊNCIAS

- [1] Cook, Jonathan E., Wolf, Alexander L. Software Validation: Quantitatively Measuring the Correspondence of a Process to a Model. ACM Transactions on Software Engineering and Methodology. vol 8 No 2, p. 147-176,1999.
- [2] Ferreira, B.; Moita, G. F. Avaliação de técnicas para validação em processos de desenvolvimento de software. In: VIII Simpósio de Mecânica Computacional, Belo Horizonte: PUC Minas,. vol. 1. p. 1-15, 2008 a
- [3] Ferreira, B.; Moita, G. F. The evaluation of different validation techniques for software development process. In: 8th World Congress on Computational Mechanics WCCM8, Veneza. Proceedings of the 8th World Congress on Computational Mechanics, v. 1. p. 1-2, 2008 b.
- [4] Ferreira, B.; Moita, G. F. Técnica de Inspeção para a Validação de um Processo de Desenvolvimento de Software. In: 30º Congresso Íbero Latino Americano de Métodos Computacionais em Engenharia, Búzios - RJ: UFRJ,. vol. 1. p. 16-26, 2009 a.
- [5] Ferreira, B.; Moita, G. F. Uma Proposta de Técnica por Inspeção para a Validação de Processos de Desenvolvimento de Softwares. In: X Encontro de Modelagem Computacional, Rio de Janeiro - RJ: IME,. vol. 1. p. 120-130, 2009 b.

- [6] Laitengerger, O., Atkinson, C. Generalized Perspective Based Inspection to handle Object Oriented Development Artifacts, Proceedings of ICSE 99, p. 494-503., 1998
- [7] Moor, A., Delugach H. Software Process Validation: Comparing Process and Practice Models. In Contributions to ICCS 2005, Kassel, Germany. p 533-540, 2006
- [8] Pereira Jr, M. Concepção de um Processo Específico para Software Científico. Dissertação de Mestrado. Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG. Belo Horizonte, 2007.
- [9] Pressman, R. S. Engenharia de Software. 6ª ed. Rio de Janeiro: McGraw-Hill, 2006.
- [10] Kalinowsh k, M. Infra-Estrutura Computacional de Apoio ao Processo de Inspeção de Software, Dissertação de Mestrado, Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ, Rio de Janeiro, 2004.
- [11] Koscianski et al., Guia para a utilização das normas sobre avaliação de qualidade de produto de software ISO/IEC 9126 e ISO/IEC 14598, 1999.
- [12] Sauer, C., Jeffery, D.R., Land, L., Yetton, P. The Effectiveness of Software Development Technical Review: A Behaviorally Motivated Program of Research, IEEE Transactions on Software Engineering 26, 1-14, 2000.