

## 1. Introdução

O presente estudo possui como objetivo a formalização de um processo de Verificação e Validação de *software* para o contexto de Fábrica de *Software*, contemplando além do estágio de desenvolvimento do produto, todo o ciclo de vida da mesma.

Para este contexto, entende-se como verificação de *software* a análise que confirma que um produto está ou foi desenvolvido corretamente, de forma que os resultados de produtos estejam em conformidade com as especificações impostas em atividades anteriores. (Bourque & Fairley, 2014). Entende-se como validação de *software* a análise que confirma que o produto está ou foi desenvolvido, de forma que o produto satisfaça seu propósito esperado (Bourque & Fairley, 2014). Ambos os aspectos de verificação e de validação de *software* buscam garantir a qualidade de um produto de *software* e a eficiência de um processo de desenvolvimento de *software* (Ghezzi et al., 2002).

Fábrica de *software* é um ambiente de desenvolvimento configurado para apoiar um desenvolvimento eficiente de um tipo específico de aplicação (Greenfield et al., 2004). O conceito de fábrica de *software* foi inicialmente proposto em 1968 por R. W. Bemer, onde estabeleceu diretrizes para aprimorar o desenvolvimento de *software* através de ferramentas padronizadas e o uso de um banco de dados com o histórico de desenvolvimento anteriores. Para este contexto, fábrica de *software* é um paradigma que promove o projeto de um esforço cooperativo de desenvolvimento de uma infraestrutura capaz de produzir *software* de alta qualidade e em grande quantidade (Cusumano, 1989).

## 2. Verificação e validação no MPS-BR

O modelo MPS baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de software e serviços correlatos e também para a melhoria da qualidade e produtividade dos serviços prestados (MPS, 2012). O guia MPS pode ser utilizado em empresas públicas e privadas, tal que a maturidade é dividida em sete níveis, de A a G.

Dentro do MPS-BR os processos de verificação e validação estão dentro do nível D de maturidade (largamente definido). No caso da validação, são esperados sete resultados, enquanto que na verificação são apenas seis.

## 2.1. Fábrica de Software

### Validação

O objetivo da validação é validar que um produto de software atenderá a seu objetivo quando colocado no ambiente para o qual foi desenvolvido [Sommerville, 2007].

De forma geral o processo de validação tem seu foco em como avaliar a qualidade de um produto ou componente de produto, assegurando que os objetivos e ou necessidades dos clientes sejam atendidas quando colocado em seu ambiente de produção, ou seja, o objetivo da validação é garantir que o produto correto está sendo desenvolvido [SOFTEX, 2011].

### Verificação

O propósito do processo Verificação é confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados [Pressman, 1995].

O processo Verificação trata de como avaliar produtos de trabalho e serviços, garantindo que atendam a seus requisitos, por meio da identificação dos itens a serem verificados, do planejamento da verificação de cada um destes itens e da execução da verificação conforme planejado ao longo do desenvolvimento do produto [SOFTEX, 2011].

Em uma fábrica de software podemos utilizar das VALs e VERs. Sendo elas:

### VALs

**VAL1 - Produtos de trabalho a serem validados são identificados** Este resultado esperado visa garantir que sejam identificados os produtos ou componentes de produto que serão validados ao longo do projeto.

**VAL2 - Uma estratégia de validação é desenvolvida e implementada, estabelecendo cronograma, participantes envolvidos, métodos para validação e qualquer material a ser utilizado na validação** Este resultado esperado tem como objetivo garantir que as atividades de validação sejam planejadas com a definição de procedimentos, infra-estrutura necessária, cronograma, recursos e responsabilidades. Os métodos a serem usados nas atividades de validação também devem ser identificados.

**VAL3 - Critérios e procedimentos para validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido** O objetivo deste resultado esperado é garantir que os critérios e procedimentos a serem utilizados para a validação foram identificados e que foi estabelecido um ambiente para validação, semelhante ao ambiente operacional.

**VAL4 - Atividades de validação são executadas para garantir que o produto esteja pronto para uso no ambiente operacional pretendido** Este resultado esperado visa garantir que as atividades de validação foram realizadas nos produtos e componentes de produto conforme o planejado.

**VAL5 - Problemas são identificados e registrados** Este resultado esperado visa garantir que os problemas identificados durante a execução das atividades de validação foram documentados e que foram definidos quais problemas serão tratados. Estes problemas devem ser acompanhados até sua conclusão.

**VAL6 - Resultados de atividades de validação são analisados e disponibilizados para as partes interessadas** O alcance deste resultado esperado envolve realizar uma análise dos resultados obtidos em decorrência da execução das atividades relacionadas a validação e disponibilizar estes resultados para o cliente, ou seu representante na execução das atividades, e outras partes interessadas.

**VAL7 - Evidências de que os produtos de software desenvolvidos estão prontos para o uso pretendido são fornecidas** Quando as atividades de teste são realizadas e há evidências que o produto satisfaz os requisitos e as expectativas do cliente, o produto pode ser considerado validado. Para isso o produto deve ser testado em seu ambiente real de uso ou em uma reprodução deste ambiente.

## **VERs**

**VER1 - Produtos de trabalho a serem verificados são identificados** Para atender a este resultado esperado deve-se analisar os produtos de trabalho que serão produzidos ao longo do projeto e selecionar aqueles a serem verificados.

**VER2 - Uma estratégia de verificação é desenvolvida e implementada, estabelecendo cronograma, revisores envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação** O alcance deste resultado esperado envolve definir uma estratégia de verificação

descrevendo os procedimentos, a infra-estrutura necessária e as responsabilidades pelas atividades de verificação.

**VER3 - Critérios e procedimentos para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é estabelecido** O alcance deste resultado esperado implica na definição dos critérios e procedimentos que serão utilizados para a verificação de cada produto de trabalho e na preparação do ambiente para verificação, disponibilizando ferramentas, recursos de hardware, infra-estrutura de rede e outros recursos necessários à execução das atividades planejadas.

**VER4 - Atividades de verificação, incluindo testes e revisões por pares, são executadas** Este resultado esperado visa garantir que as atividades de verificação são executadas conforme planejado, o que inclui, obrigatoriamente, a realização de revisão por pares e testes.

**VER5 - Defeitos são identificados e registrados** Este resultado esperado visa garantir que os defeitos identificados durante a execução da verificação são documentados e registrados. Para registro dos defeitos identificados pode-se usar uma classificação de defeitos, por exemplo, por severidade (crítico, sério, moderado) ou por origem (requisitos, projeto (design), código, testes).

**VER6 - Resultados de atividades de verificação são analisados e disponibilizados para as partes interessadas** O alcance deste resultado esperado envolve realizar uma análise dos resultados obtidos em cada atividade de verificação e disponibilizar estes resultados para as partes interessadas.

## **VALs e VERs aplicadas ao contexto de Fábrica de software**

Das VALs definidas pelo guia, podemos utilizar as seguinte atividades em um processo de VV no contexto de fábrica de software:

**Revisão dos componentes** Após o desenvolvedor/testador implementar o código e testá-lo, então tem-se em mãos uma versão do componente codificado. Assim, terminada essa tarefa, o papel Revisor, que é uma pessoa externa da equipe de desenvolvimento do componente, será responsável pela revisão de todo o código e, então, produzirá um relatório que será encaminhado para o Líder, o qual pode aprová-lo ou não.

**Verificação de Compilação** Clareza, facilidade e agilidade na compilação do componente.

**Verificação do Modelo de Componentes** Verificar se o componente desenvolvido pelo parceiro está de acordo com o modelo

**Verificação de conformidade com a arquitetura pré-definida** Verificar se todas as classes necessárias, definidas na arquitetura, de acesso ao componente foram implementadas

**Verificação do seguimento dos padrões de codificação** Verificar se o código/arquitetura segue os padrões definidos pelo cliente/time de desenvolvimento

**Verificação da implementação e cobertura dos testes de unidade** Verificar se os testes unitários atingiram um grau de cobertura mínimo de X% do código do componente.

### 3. Processo

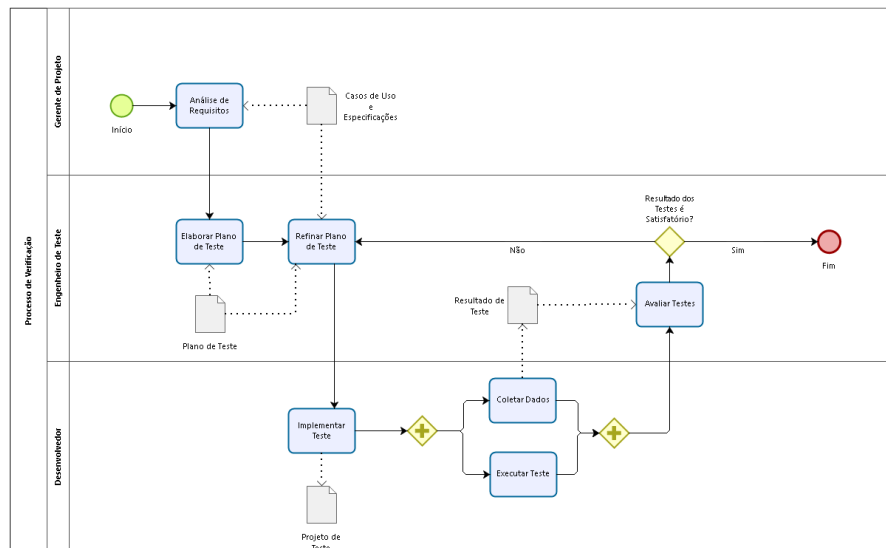


Figure 1: Processo

### Papéis:

- Gerente de requisitos: Responsável pela elicitação dos requisitos junto ao cliente e sua conformidade com os documentos de testes.
- Gerente de teste: Responsável pelos documentos de teste, elaboração da estrutura(suite) de testes que será implementada posteriormente pelo desenvolvedor. A análise dos resultados dos testes também será analisada pelo gerente de testes.
- Desenvolvedor: Responsável por implementar os testes e coletar os resultados e entrega-los ao gerente de teste.

### Artefatos:

- Caso de Usos e especificações: Narrativas de texto descrevendo unidade funcional de valor que tem algum valor para o cliente.
- Plano de Teste: Terá o planejamento dos testes juntamente com os casos de testes, regras de negócio, tudo que deve ser testado, juntamente com valores de referências e expectativas sobre os resultados.
- Resultados do teste: Resultados do teste que serão avaliados pelo gerente de teste.

## 4. Conclusão

Com base nos resultados obtidos no estudo das atividades gerais, de um processo de verificação e validação de *software* dentro do MPS-Br, podemos afirmar que é possível aplicar no contexto de Fábrica de *Software* as atividades inerentes a esse processo. Como resultado, esse estudo apresentou um processo de Verificação e Validação de *software* para o contexto de Fábrica de *Software*, contemplando todo o ciclo de vida do desenvolvimento do produto.

## Referências

- Bourque, Pierre, & Fairley, Richard E. (2014). *Guide to the Software Engineering Body of Knowledge*. IEEE.
- Ghezzi, Carlo, & Jazayeri, Mehdi, & Mandrioli, Dino (2002). *Fundamentals of Software Engineering*. Prentice Hall.
- Greenfield, Jack, & Short, Keith, & Cook, Steve, & Kent, Stuart (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. Microsoft Corporation.
- Bemer, R. W. (1969). *Position papers for Panel Discussion: The economics of Program Production*. A. J. H. Morrell.

- Cusumano, M. A. (1989). *The Software Factory: A Historical Interpretation*. IEEE.
- Sommerville, Ian. Engenharia de Software - 8ª Edição 2007;
- Pressman, Roger s. Engenharia de Software – 1995;
- SOFTEX. Guia de Implementação – Parte 9: Implementação do MR-MPS - 2011;