VOLTAR

Testes de Software - Verificação e Validação

Estudar os conceitos sobre verificação e validação de software.

NESTE TÓPICO

Introdução – Conceitos de Verificação e Validação de Software

As atividades de Verificação e Validação (V&V) têm o propósito de certificar que o software esteja em conformidade com as especificações, atendendo aos requisitos dos usuários. É um abrangente conjunto de processos dentro das iniciativas de Garantia da Qualidade de Software (GQS). Elas ocorrem em cada etapa do processo de software, iniciando com as revisões de requisitos, sendo continuadas nas revisões de projetos e inspeções de código até os testes de produto (PRESSMAN, 2011; SOMMERVILLE, 2007; KOSCIANSKI e SOARES, 2007).

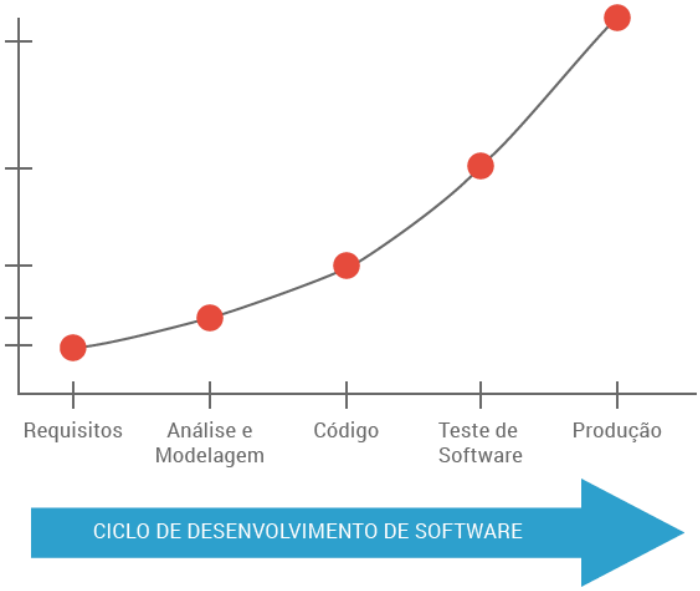
Embora os termos verificação e validação pareçam sinônimos, eles possuem significados diferentes. Dentro dos conceitos de engenharia de software, Boehm (1979) conseguiu definir cada um deles com uma pergunta:

- “Validação: Estamos construindo o produto correto?”
- “Verificação: Estamos construindo o produto corretamente?”

Verificação e Validação

Sommerville (2007) os diferencia primeiro dizendo que “o papel da verificação envolve verificar se o software está de acordo com as especificações”, ou seja, o atendimento aos requisitos funcionais e não funcionais. Por outro lado, o autor afirma que “a validação é um processo mais geral. A finalidade da validação é assegurar que o sistema de software atenda as expectativas do cliente”. Dessa forma, a validação extrapola a simples averiguação se o software está de acordo com as especificações, para confirmar se o software atende ao que o cliente espera.

Um dos objetivos mais importantes da V&V é a redução dos custos de desenvolvimento e evitar prejuízos com o mau funcionamento do software em produção. Quanto ao projeto de desenvolvimento de software, o gráfico abaixo ilustra que os problemas evidenciados nos estágios finais do desenvolvimento são mais custosos do que se tivessem sido descobertos nos estágios inicias.



O custo da propagação dos defeitos

Entre a ampla gama de atividades de Garantia da Qualidade de Software que V&V incluem, podemos destacar algumas:

- Revisões técnicas
- Inspeção de Programa
- Auditorias de qualidade e configuração
- Monitoramento de desempenho
- Simulação
- Estudo de viabilidade
- Revisão de documentação
- Revisão de base de dados
- Análise de algoritmo
- Processos de Teste

Como é possível perceber, embora a aplicação de teste tenha um papel extremamente importante em V&V, outras atividades também são necessárias.

> O teste proporciona a última oportunidade para que os erros possam ser descobertos antes da entrega do software. Mas o teste não deve ser visto como uma rede de segurança. Se a qualidade não está lá antes de um teste, ela não estará lá quando o teste terminar.

> A aplicação correta de métodos e ferramentas, de revisões técnicas eficazes, e de um sólido gerenciamento e avaliação conduzem todos à qualidade que é confirmada durante o teste.
>
> ((PRESSMAN, 2011))

Entre as atividades de V&V, vamos destacar nesse tópico os processos de revisão. Eles compreendem o exame, do ponto de vista técnico, de um documento, de um código ou de qualquer outro artefato produzido em um processo de desenvolvimento ou manutenção de software.

Revisões Técnicas

Durante todo o processo de software, seja em projetos de desenvolvimento ou na manutenção do software, cometemos uma série de erros. Isso é inerente aos tipos de atividades que envolvem a produção e atualização do software. Por esse motivo, existem mecanismos para revelar esses erros, de forma que eles possam ser corrigidos o mais cedo possível.

A descoberta prematura de erros permite que eles sejam menos custosos para serem corrigidos. Pense na seguinte situação: um analista faz uma reunião com o cliente e define um conjunto de requisitos funcionais para um determinado software. No dia seguinte, esse mesmo analista se reúne com outro analista mais sênior para que esse faça uma revisão dos requisitos levantados, onde são descobertas inconsistências. Desse modo, é feita nova reunião com o cliente, onde são refinados os requisitos e resolvidos os problemas de inconsistências, os quais gerariam um software com algumas funções inadequadas aos usuários.

A descoberta dos erros constante logo na primeira versão do documento de requisitos da situação hipotética, graças a um processo de revisão técnica nesse documento, com certeza geraria um custo menor do que se o erro se propagasse no projeto e sua descoberta acontecesse somente em fases mais avançadas de testes no sistema.

As revisões podem ter uma abordagem mais ou menos formal. A formalização da revisão, muitas vezes, se torna importante para garantir que evidências de revisão sejam geradas para atender a exigências de auditoria e um melhor gerenciamento do projeto. Porém, muitas vezes em uma “conversa de corredor”, em um encontro informal entre membros da equipe, descobrem-se possíveis problemas para o projeto. Essa oportunidade de antecipação a uma futura falha no produto de software não pode ser desprezada e deve ser corretamente endereçada a uma correção.

Principal objetivo das revisões técnicas: “Encontrar erros durante o processo, de modo a não se tornarem defeitos depois da liberação do software” (PRESSMAN, 2011).

Revisões Técnicas Formais

A Revisão Técnica Formal (RTF) é estruturada e visa exercer o controle da qualidade de software pela equipe de projeto. Seus objetivos são (PRESSMAN, 2011):

- Descobrir erros na função, lógica ou implementação para qualquer representação do software;
- Verificar se o software que está sendo revisado atende aos requisitos;
- Garantir que o software foi representado de acordo com padrões predefinidos;
- Obter software que seja desenvolvido de maneira uniforme;
- Tornar os projetos mais gerenciáveis.

As reuniões de revisão devem ter planejamento antecipado e ter foco em uma parte específica do software. Desse modo, ela terá maior probabilidade de revelar erros. É importante que os revisores convidados recebam com antecedência o material a ser revisado, invistam tempo suficiente para a revisão (recomenda-se em média cerca de duas horas) e levem à reunião suas anotações.

No final da revisão, os participantes da RTF decidem se:

- Aceitam o artefato sem as modificações adicionais,
- Rejeitam o artefato devido a erros graves (uma vez corrigidos, deve ser realizada outra revisão) ou
- Aceitam o artefato provisoriamente (foram encontrados erros secundários que devem ser corrigidos, mas não haverá nenhuma outra revisão).

Uma vez tomada a decisão, os participantes da RTF assinam um documento de aprovação, indicando sua participação na revisão e concordância com as descobertas da equipe de revisão.

Diretrizes para uma Revisão Técnica Formal

Pressman (2011) apresenta 10 diretrizes para uma RTF:

1. Revisar o produto, não o produtor.
2. Estabelecer uma agenda e mantê-la.
3. Limitar debates e refutação. Mesmo sem um acordo sobre um item levantado, a reunião não é momento para debater a questão – deve limitar-se a anotar as áreas de possíveis problemas para posterior averiguação.
4. Enunciar as áreas do problema mas não tentar resolver todo o problema registrado.
5. Tomar notas. Se possível anotar em um quadro para que todos os revisores avaliem os potenciais problemas e prioridades.
6. Limitar o número de participantes e insistir na preparação antecipada.
7. Desenvolver uma lista de verificação para cada artefato que provavelmente será revisado.
8. Alocar os recursos e programar o tempo para as RTFs
9. Realizar treinamento significativo para todos os revisores.
10. Revisar revisões iniciais. Os primeiros artefatos a ser revisados devem ser as próprias diretrizes de revisão.

O produto final do processo de revisão é uma lista de possíveis problemas e/ou erros descobertos, assim como a indicação do estado técnico do objeto revisado. Sendo a revisão técnica mais formal ou mais informal, ela tem o objetivo de revelar problemas no projeto que, o quanto antes forem descobertos, menor a probabilidade de propagação para outros artefatos que serão produzidos.

Inspeção de Programa

São também revisões, com o objetivo específico de encontrar defeitos em programas. Os defeitos podem ser erros de lógica ou irregularidades no código que possam estar em não-conformidade com os padrões de desenvolvimento da organização, ou mesmo com boas práticas recomendadas pela engenharia de software.

Como qualquer revisão técnica, ela pode ser formal ou informal e as principais diretrizes vistas para uma revisão técnica formal são válidas também para uma revisão (inspeção) de programa.

Revisão Técnica

No quadro abaixo, Sommerville (2007) traça algumas classes de defeitos e possíveis verificações que podem ser feitas nessas classes, funcionando como um guia base (checklist) para estabelecer critérios para um processo de inspeção de programas.

CLASSES DE DEFEITOS	VERIFICAÇÃO E INSPEÇÃO
Defeitos de Dados	• Todas as variáveis de programa são iniciadas antes que seus valores sejam usados? • Todas as constantes foram nomeadas? • O limite superior de vetores deve ser igual ao tamanho do vetor ou ao tamanho -1? • Se as strings de caracteres são usadas, um delimitador é explicitamente atribuído? • Existe alguma possibilidade de overflow de buffer?
Defeitos de Controle	• Para cada instrução condicional, a condição está correta? • É certo que cada loop vai terminar? • As declarações compostas estão posicionadas corretamente entre colchetes? • Em declaração case, todos os cases possíveis são considerados? • Se um break é requerido após cada case em declarações case, esse foi incluído?
Defeitos de Entrada/ Saída	• Todas as variáveis de entrada são usadas? • Todas as variáveis de saída receberam um valor antes de serem emitidas? • Entradas inesperadas podem causar corrupção de dados?
Defeitos de Interface	• Todas as chamadas de funções e métodos têm o número correto de parâmetros? • Os parâmetros formais e reais correspondem? • Os parâmetros estão na ordem correta? • Se os componentes acessam memória compartilhada, eles têm o mesmo modelo de estrutura de memória compartilhada?
Defeitos de Gerenciamento de Armazenamento	• Se uma estrutura ligada é modificada, todas as ligações foram corretamente retribuídas? • Se o armazenamento dinâmico é usado, o espaço foi alocado corretamente? • O espaço de memória é liberado depois de não ser mais necessário?
Defeitos de Gerenciamento de Exceções	• Todas as possíveis condições de erro foram consideradas?

Checklist de Inspeção de Programa

Fonte: (SOMMERVILLE, 2007)

Os checklists são ferramentas muito úteis nos processos de revisão, pois alertam a pontos importantes que devem ser observados, além de ajudar a estabelecer quais os critérios de qualidade que serão abordados. Porém, é importante salientar que um checklist é um guia genérico que deve ser adaptado (customizado) para os processos organizacionais padrão, para o projeto que está em execução, para a tecnologia que está sendo aplicada, ou mesmo para as necessidades específicas da inspeção que está sendo feita. O checklist proposto apresentado no quadro acima, por exemplo, deve ser adaptado à linguagem de programação que está sendo utilizada.

As revisões e inspeções, quando aplicadas em conjunto com um consistente gerenciamento das atividades, conduzem todos à qualidade. Qualidade essa que é confirmada durante o teste.

Quiz

Exercício Final

Testes de Software - Verificação e Validação

INICIAR

Referências

BOEHM, B. W. Software engineering: R&D trends and defense needs. In: *Research directions in software technology*. Cambridge: MA: MIT Press, 1979. Cap. 22, p. 44-86.

KOSCIANSKI, A.; SOARES, M. D. S. *Qualidade de software*: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2007.

PRESSMAN, R. S. *Engenharia de Software*: Uma Abordagem Profissional. 7. ed. Porto Alegre: McGraw Hill, 2011.

SOMMERVILLE, I. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

Avalie este tópico

uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove (http://www.uninove.br)
Mapa do Site