

Engenharia de Software

Ficha T. Prática nº 6

Fonte: Eng. De Software, Coleção Schaum

Objectivo: Garantia da qualidade do software

1. Introdução

Existem muitas maneiras de definir a qualidade. Nenhuma é perfeita. Uma definição é “qualidade é a totalidade das características dum produto ou serviço que se baseia na sua habilidade de satisfazer uma dada necessidade”. Outra definição é que software de qualidade é aquele que faz o que se espera que ele faça. Outras definições baseiam-se na satisfação do utilizador, etc. A técnica principal para alcançar a qualidade é a revisão de software que tem por objecto encontrar erros no software. As abordagens formais têm demonstrado ser mais eficientes do que as abordagens informais. A métrica mais usada para avaliar a inspecção é erros encontrados/KLOC. A eficiência pode ser medida em termos dos erros encontrados/nº horas trabalhadas.

2. Inspeção formal e técnicas de revisão

A inspeção formal é uma actividade agendada na qual um projectista apresenta um material sobre um projecto e um selecto grupo de pares avalia os aspectos técnicos. Os detalhes de como a inspeção formal ou revisão técnica é feita podem variar muito. Os seguintes aspectos são geralmente aceites como os que distinguem a inspeção formal de outras revisões:

Pares conhecedores do assunto são usados

O desenvolvedor é um participante activo

Um produto explícito, completo, é inspeccionado

O propósito principal é encontrar defeitos

A inspeção formal é usada rotineiramente no desenvolvimento de software

Papéis específicos são designados

A inspeção usa passos específicos da inspeção formal

Pelo menos três pessoas estão envolvidas na inspeção

Papéis da inspeção

Moderador – o moderador selecciona a equipa, conduz a inspeção e relata os resultados

Leitor – o leitor é geralmente o não desenvolvedor do produto; entretanto, ele irá dirigir a equipa pelo trabalho do desenvolvedor durante o encontro da inspeção.

Arquivista – O arquivista mantém os arquivos da inspeção e relata minuciosamente cada defeito

Desenvolvedor – o desenvolvedor é quem originalmente produz o produto. O papel dele é responder a perguntas durante a inspeção. O desenvolvedor é também o responsável por corrigir qualquer problema identificado e relatar as correcções para o moderador

Passos da inspecção

Revisão – Quando o desenvolvedor satisfaz os critérios de entrada, a inspecção é agendada. O desenvolvedor, então, conduz uma revisão, o que contempla o restante da equipa de inspecção com o produto a ser inspecionado

Preparação – Os membros da equipa de inspecção estudam o produto. O tempo gasto na preparação é controlado, baseado no tamanho do produto em KLOC. Os membros podem usar uma checklist para focar em questões significativas.

Encontro de inspecção – O moderador supervisiona o encontro de inspecção. Algumas abordagens usam o leitor em vez do desenvolvedor para, eventualmente, conduzir a inspecção. O arquivista elabora um arquivo completo com as questões levantadas. Todos os membros da equipa de inspecção assinam o relatório. Qualquer membro da equipa pode gerar um relatório da minoria, se houver alguma discordância

Retrabalho – O desenvolvedor revê o relatório e corrige o produto

Próximos passos – O moderador revê o relatório e correção. Se satisfaz o critério de saída, a inspecção é finalizada; se não, o revisor pode exigir que o desenvolvedor refaça o produto ou uma inspecção pode ser agendada.

Checklists

Uma lista de verificação (checklist) é uma lista de itens que devem ser verificados durante a revisão. Algumas vezes os itens são colocados como perguntas para serem respondidas. A vantagem de uma checklist é que ela foca a atenção do revisor nos problemas potenciais. Qualquer falha encontrada deve ser analisada para verificar se o item da lista garante o foco neste problema.

(Lembro-me dum processo longo de detectar erro causado por um ponto-e-vírgula imediatamente após uma condição num comando IF em C++. Agora, qualquer checklist para programas em C++ que eu escrevo inclui a verificação de ponto-e-vírgula ao final das condições de decisão)

Itens da checklist que não são eficientes em encontrar erros durante a inspecção devem ser removidos, pois uma quantidade grande de itens na checklist irá prejudicar a eficiência da inspecção

3. Confiabilidade do software

Confiabilidade é a probabilidade do software não falhar num período de tempo específico. Geralmente é denominada por $R(n)$, onde n é um número de unidades de tempo. Se a unidade de tempo é dias, então $R(1)$ é a probabilidade de não falhar 1 dia. A probabilidade de falha no período de tempo específico é 1 menos a confiabilidade para o período ($F(n) = 1 - R(n)$).

Média de erros

Se um erro ocorre cada 2 dias, então a média de erros instantânea seria de 0,5 erros por dia. A média de erros é o inverso do tempo entre erros. A média de erros pode ser usada como uma estimativa da probabilidade de falha, $F(1)$. A menos que conheçamos alguma tendência a melhor estimativa para o comportamento do futuro próximo é o comportamento corrente. Então, se encontramos 20 erros num dia, a melhor estimativa para o próximo dia é 20 erros.

Exemplo: Se um erro ocorre após 2 dias, qual a probabilidade de que o sistema não irá falhar em 1,2,3 ou 4 dias? Se um erro ocorre cada 2 dias, podemos usar 0,5 como a média de erros instantânea, que pode também ser usada para estimar a probabilidade para 1 dia. Então, $F(1) = 0,5$. Então $R(1) = 1 - F(1) = 0,5$; $R(2) = 0,25$; $R(3) = 0,125$. $R(4) = 0,0625$.

4. Padrões IEEE para o plano de SQA

Uma parte importante na obtenção da qualidade é planejar para a qualidade, isto é, planejar as actividades que irão auxiliar na obtenção da qualidade. A IEEE desenvolveu um padrão para planos de garantia da qualidade de software. A seguir estão parte das secções especificadas no IEEE

Propósito – Esta secção deverá listar o software e as partes contempladas do seu ciclo de vida

Documentos de referência – Esta secção deverá listar todos os documentos referenciados no plano

Gestão

Organização – Esta secção deverá descrever a estrutura da organização e as responsabilidades e geralmente inclui um organograma

Tarefas – esta secção deverá listar todas as tarefas a serem executadas, os relacionamentos entre tarefas e os pontos de verificação e a sequencia de tarefas

Responsabilidades – esta secção deverá listar as responsabilidades de cada unidade organizacional

Documentação

Propósito – esta secção deverá listar todos os documentos necessários a estabelecer como os documentos serão avaliados

Documentos mínimos – Esta secção deverá descrever a documentação mínima necessária, geralmente incluindo:

Especificação de requisitos

Descrição do projecto

Plano de verificação e validação de software

Relatório de verificação e validação de software

Documentação do utilizador

Plano de gestão da configuração do software

Padrões, práticas, convenções e métricas – esta secção deverá identificar padrões, práticas, convenções e métricas a serem aplicadas e como serão monitorizadas e

garantidas. O conteúdo mínimo deverá incluir padrões de documentação, de estrutura lógica, de codificação, de teste, produtos de SQA seleccionados e métricas do processo

Revisões e auditoria – esta secção deverá definir quais revisões/auditorias serão feitas, como serão acompanhadas e quais acções serão realizadas após

Testes – esta secção deverá incluir todos os testes não incluídos no plano de verificação e validação do software

Relatório do problema – Esta secção deverá definir práticas e procedimentos para relatar, acompanhar e resolver, incluindo responsabilidades organizacionais

Ferramentas, técnicas e metodologias – esta secção deverá identificar as ferramentas, técnicas e metodologias especiais do software e descrever os seus usos

Controlo de código – esta secção deverá definir os métodos e facilidades para manter o controlo de versões do software

Controlo dos meios – esta secção deverá definir os métodos e facilidades para identificar, armazenar e proteger os meios físicos

Controlo de fornecedor – esta secção deverá estabelecer providências para garantir que o software entregue pelos fornecedores atenda aos padrões

Registos – esta secção deverá identificar a documentação a ser retida e os métodos para recolher, manter e salvaguardar essa documentação

Treino – Esta secção deverá identificar as actividades de treino necessárias

Gestão de risco – esta secção deverá especificar métodos e procedimentos para a gestão de risco

Exemplo:

Desenvolva a secção 3 e a secção 8 dum plano SQA para um projecto de desenvolvimento de software. Considere um gerente de projecto chamado Carlos; uma equipa externa de teste composta por Tina a líder, Dora e Helena; uma equipa de gestão da configuração composta por Marcos (líder), Paulo e Lucas e uma equipa de garantia da qualidade composta por Jorge (líder) e Bruno

Secção 3 – Organização

Tarefas – Todos os documentos serão revistos. Uma ferramenta de gerência de configuração irá gerir todos os documentos e módulos do código fonte. Todos os planos de teste serão feitos durante a fase de requisitos e incluirão um número adequado de casos de teste. Inspeções formais serão conduzidas no final de cada fase.

Responsabilidades

A equipa do projecto é responsável por todo o desenvolvimento, incluindo requisitos, projecto e implementação e produz o plano de teste como parte dos requisitos. Eles também são responsáveis por toda a documentação, incluindo o manual do utilizador e documentos de acompanhamento

A equipa de teste é responsável por testar a versão básica do código fonte. A equipa de teste irá usar o plano de teste desenvolvido durante os requisitos. Casos adicionais de teste serão desenvolvidos para satisfazer cada comando do código. Quaisquer discrepâncias no plano de teste e/ou requisitos ou nos testes serão relatados para o gerente geral.

A equipa de gerência de configuração será responsável por aceitar os itens de configuração do software e atribuir números às versões. A equipa de garantia da qualidade será responsável por supervisionar as revisões e inspeções e irá acompanhar os problemas relatados.

Secção 8 – Relatório do problema

Todos os problemas identificados fora da unidade de desenvolvimento devem ser relatados para a equipa de garantia da qualidade, que designará um número para o problema. O gerente de cada equipa irá aprovar as correções dos problemas mencionados e designados para esta equipa. A equipa de garantia da qualidade será responsável por acompanhar todos os problemas e semanalmente relatar para o gerente geral.

Problemas:

1. Construa uma checklist para a revisão de programas escritos em código C
2. Construa uma checklist para a revisão dum projecto de software
3. Calcule a confiabilidade dum software que teve 10 erros em 200 casos de teste.
4. Assuma que uma técnica ABC requer 2 horas/KLOC de preparação e que tenha sido alocada 1 hora/KLOC para revisão e a técnica XYZ requer 1 hora/KLOC de preparação e 4 horas para a revisão. Também assumo que a técnica ABC encontra 12 erros/KLOC e a técnica XYZ encontra 14 erros/KLOC. Compare a eficiência de ambas técnicas.
5. Se o software teve 5 falhas em 100 testes durante 10 dias, o que seria uma boa estimativa dum boa confiabilidade para os próximos 5 dias? 2 semanas?
6. Desenvolva um plano de SQA para o problema da pousada, eleja as secções que apliquem ao problema
7. Um erro 1 ocorre após 4 dias e o erro 2 ocorre 5 dias depois. Desenhe um gráfico de média de erros versus o número de erros e a média de erros versus o tempo e estime o número de erros no sistema e o tempo para remover todos os erros.