




Laboratório de Engenharia de Software

Plano de Teste

Arndt von Staa
Departamento de Informática
PUC-Rio
Maio 2014



Especificação

Laboratório de Engenharia de Software


- Objetivo desse módulo
 - apresentar e discutir planos de teste
- Justificativa
 - para realizar testes de forma confiável é necessário trabalhar de forma disciplinada
 - o plano estabelece os princípios a serem observados durante todo desenvolvimento e auxilia em determinar esforço (custos) e tempos requeridos para as atividades relacionadas com controle e garantia da qualidade
- Texto
 - Bastos, A. 2; Rios, E.; Cristalli, R.; Moreira, T.; *Base de Conhecimento em Teste de Software*; São Paulo: Martins; 2007
 - Pezzè, M.; Young, M.; *Teste e Análise de Software*; Porto Alegre, RS: Bookman; 2008, capítulo 20

Mai 2014

Arndt von Staa © LES/DI/PUC-Rio

2

Organização do plano




Laboratório de Engenharia de Software

- O plano de teste é organizado em uma série de seções.
- Existem várias propostas na literatura e que, de maneira geral, diferem muito pouco uma da outra
- As transparências a seguir descrevem as seções de uma dessas propostas

Mai 2014Arndt von Staa © LES/DI/PUC-Rio3

Plano de teste




Laboratório de Engenharia de Software

- Introdução
 - foco nos objetivos que motivaram o desenvolvimento
 - melhoria da imagem
 - redução de falhas operacionais
 - aceleração do processo empresarial
 - maior acuidade dos resultados
 - . . .
- Definição dos requisitos de alto nível
 - ex. sistema de registro e acompanhamento
 - ex. sistema embarcado que monitora ...

Mai 2014Arndt von Staa © LES/DI/PUC-Rio4

Plano de teste




Laboratório de Engenharia de Software

- Produtos do teste a serem entregues (*deliverables*)
 - o histórico de problemas registrados
 - falhas, adaptações, evoluções, melhorias
 - o próprio plano de teste
 - o projeto dos casos de teste
 - objetivos específicos do teste
 - processo de teste utilizado
 - evidência que os objetivos foram alcançados
 - cenários de teste
 - métodos de seleção dos casos de teste utilizados
 - ...
 - as suítes de teste, i.e. coletâneas de massas de teste
 - os logs de execução dos casos de teste
 - os relatórios de medição
 - os relatórios de falhas (laudos)
 - ideal: falhas por versão intermediária conforme o controle de versões
 - um relatório resumido (retrospectiva)

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
5

Plano de teste




Laboratório de Engenharia de Software

- Processo de teste
 - como será realizada a garantia da qualidade
 - quando e como realizar revisões ou inspeções
 - padrões a serem observados
 - arquitetura
 - projeto
 - codificação
 - uso de assertivas
 - documentação
 - interação entre teste e desenvolvimento
 - desenvolvimento visando testabilidade
 - desenvolvimento dirigido por testes
 - automação da geração de casos de teste
 - ferramentas de apoio ao teste automatizado a serem utilizadas

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
6

Plano de teste




Laboratório de Engenharia de Software

- Identificação dos tipos de teste
 - onde será utilizado o teste manual
 - como será realizado o teste manual
 - onde será utilizado o teste automatizado
 - como será realizado o teste automatizado
 - onde será utilizada a geração automatizada de casos de teste
 - como será realizada a geração automatizada de casos de teste

Mai 2014Arndt von Staa © LES/DI/PUC-Rio7

Plano de teste




Laboratório de Engenharia de Software

- Identificação dos critérios de conclusão dos testes, ex.
 - todos testes são baseados em critérios de seleção rigorosos e concluem corretamente
 - testes são automatizados sempre que possível
 - evidência do teste é visível
 - falhas por determinada unidade de tempo abaixo de X
 - beta teste
 - esgotamento de recursos para o teste
 - evite usar este tipo de critério!

Mai 2014Arndt von Staa © LES/DI/PUC-Rio8

Plano de teste




Laboratório de Engenharia de Software

- Estratégia de teste de regressão
 - quais os testes que serão efetuados de novo
 - testes manuais
 - parcela dos testes automatizados
 - matriz de reteste
 - *make* de reteste
 - estratégia de desenvolvimento que leva à criação de um subsistema de apoio ao teste durante a manutenção

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
9

Plano de teste




Laboratório de Engenharia de Software

- Ambiente de teste
 - espaço físico para o trabalho
 - equipamentos
 - ferramentas, lista
 - tecnologias
 - material de apoio, treinamento
- Ferramentas de apoio aos testes
 - quais são, inclusive a versão
 - onde estão
 - material de treinamento
 - tanto ferramentas de apoio aos testes como ferramentas de apoio à gestão, ex. registro de pendências, controle de versão, automação do teste de integração contínuo, ...

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
10

Plano de teste




Laboratório de Engenharia de Software

- Gerência de configuração
 - inicialização, e.g. solicitação de alteração
 - avaliação técnica
 - avaliação do impacto sobre o negócio
 - avaliação do impacto sobre a gestão do desenvolvimento
 - acompanhamento dos testes
 - procedimento de instalação ou divulgação

Mai 2014Arndt von Staa © LES/DI/PUC-Rio11

Plano de teste




Laboratório de Engenharia de Software

- Organização da equipe de teste
 - nível das pessoas
 - habilitações, treinamento
 - papéis, responsabilidades
 - quem são as pessoas e quais os papéis que desempenham
 - integração de testadores com desenvolvedores
 - processos ágeis

Mai 2014Arndt von Staa © LES/DI/PUC-Rio12

Laboratório de Engenharia de Software

Plano de teste




- Identificação das precedências
 - código
 - projeto
 - componentes precedentes
 - disponibilidade de pessoas
 - disponibilidade de ferramentas
- Cronograma de teste
 - coleta de informações
 - planejamento
 - projeto dos roteiros de teste
 - desenvolvimento dos casos de teste
 - execução dos testes
 - integração
 - teste de sistema
 - testes de requisitos não funcionais
 - teste de aceitação
 - produção do resumo

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
13


Laboratório de Engenharia de Software

Plano de teste



- Procedimentos de composição de construto versionado
 - ferramenta de controle de versão utilizada
- Sistema de registro e acompanhamento de problemas
 - especialmente voltado para a etapa de teste
 - *issue tracker*
- Processo de decisão com relação a problemas registrados
 - máquina de estados de evolução de uma FAP amarrada às ferramentas
- Procedimentos de relato
 - evolução do estado de progresso dos testes
 - resumos dos testes
- Procedimentos de aprovação
 - quem aprova o que
 - para cada produto do teste (*deliverable*)


Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
14

Plano de teste


Laboratório de Engenharia de Software

- Métricas a serem coletadas (GQM – Goal, Question, Metric)
 - objetivos da medição: prover melhorias para:
 - ex. reduzir o número de falhas em programas entregues
 - ex. reduzir o esforço gasto em retrabalho desnecessário
 - quais as perguntas que se relacionam com os objetivos
 - ex. quantas falhas por dimensão e complexidade percebida foram detectadas em produtos já entregues
 - ex. quanto tempo é gasto em alteração de artefatos já desenvolvidos
 - ex. qual o percentual deste tempo que seria desnecessário caso fossem adotados princípios mais eficazes
 - medições a realizar
 - número de falhas reportadas após a aceitação
 - dimensão do artefato
 - complexidade percebida
- Não meça só para colecionar medições!
- Meça para **identificar, resolver ou melhorar** um problema.

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
15

Plano de teste


Laboratório de Engenharia de Software

- Exemplos de métricas
 - natureza das causas de falhas
 - distribuição das causas no tempo
 - modos de observação das falhas
 - ex. teste
 - ex. uso
 - classe das falhas, ex. desastre, grave, normal, aborrecimento
 - distribuição de falhas por artefato
 - tempo para eliminar completamente cada problema
 - esforço para eliminar completamente cada problema
 - volume de automação dos testes
 - distribuição do esforço para diagnosticar cada problema encontrado
 - distribuição de uso dos recursos de apoio aos testes
 - cobertura de código
 - cobertura de sequências de ativação

Mai 2014
Arndt von Staa © LES/DI/PUC-Rio
16

Bibliografia



- Bastos, A. 2; Rios, E.; Cristalli, R.; Moreira, T.; *Base de Conhecimento em Teste de Software*; São Paulo: Martins; 2007
- Kaner, C.; Falk, J.; Nguyen, H.Q.; *Testing Computer Software*; International Thompson Computer Press; 1993
- Kaner, C.; *Bug Advocacy: How to Win Friends and Stomp BUGs*; 2000; www.kaner.com (testing website) www.badsoftware.com (legal website)
- Kemerer, C.F.; Slaughter, S.; "An Empirical Approach to Studying Software Evolution"; *IEEE Transactions on Software Engineering* 28(4); 1999; pages 493-509
- Lewis, W.E.; *Software Testing and Continuous Quality Improvement*; Boca Raton: Auerbach; 2000
- Pezzè, M.; Young, M.; *Teste e Análise de Software*; Porto Alegre, RS: Bookman; 2008

FIM

