


Laboratório de Engenharia de Software

Teste baseado em riscos

Arndt von Staa
Departamento de Informática
PUC-Rio
Abril 2017

Especificação



Laboratório de Engenharia de Software


- Objetivo desta aula
 - discutir algumas técnicas para a redução do risco de uso de um sistema.
- Justificativa
 - Testes devem enfatizar os casos de teste relevantes
 - procurar eliminar os defeitos e as vulnerabilidades de maior risco
 - ex. código inseguro

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

2

Defeito relevante




Laboratório de Engenharia de Software

- **Defeito relevante** é um que leva:
 - a **alto risco** de uso
 - danos materiais
 - danos pessoais
 - prejuízo financeiro
 - perda de oportunidade
 - vulnerabilidade a uso malicioso
 - . . .
 - a **alto risco** de desenvolvimento ou manutenção
 - custo ao desenvolver realizado muito maior do que o estimado
 - prazo para desenvolver realizado muito maior do que o estimado
 - projeto cancelado
 - resultado do desenvolvimento descartado
 - elevada frequência de manutenção corretiva
 - elevado custo de manutenção
 - . . .

Jones, Cp.; Bonsignour, O.; *The Economics of Software Quality*; Kindle edition; Pearson Education; 2011

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
3

Riscos



Laboratório de Engenharia de Software

- Riscos são em geral consequências da **ocorrência** de eventos **indesejáveis**, exemplos
 - execução de um defeito
 - precisa-se verificar se existe um defeito da classe correspondente ao risco e então eliminá-lo
 - exploração bem sucedida de uma vulnerabilidade, ver: defeito
 - consequência de uma ação incorreta do usuário
 - consequência de algum serviço ou artefato defeituoso provido por terceiros
 - raios cósmicos (!?) – i.e. causas virtualmente impossíveis de serem conhecidas e eliminadas
- Observação
 - podem existir eventos desejáveis, ou seja: **oportunidades**
 - exemplo: o custo do desenvolvimento ser significativamente menor do que o orçado

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
4

Risco, definição



- Especificação de um risco
 - **nome** do evento causador → a **ameaça**
 - extravasão de campo
 - não possui escalabilidade
 - hacker consegue invadir
 - ...
 - **probabilidade**, exemplos que aumentam a probabilidade
 - especificação ruim
 - engenharia ruim, ex. arquitetura, projeto ruins ou até ausentes
 - alta densidade de defeitos observada ao testar
 - teste mal feito
 - ...
 - **impacto**
 - dano direto
 - dano indireto, i.e. dano propagado para outros artefatos ou sistemas
 - ...
 - **relevância**, exemplos que aumentam a relevância
 - artefato é central para o negócio
 - artefato é usado com frequência
 - artefato é componente de ou interage com vários sistemas
 - ...

Abr 2017

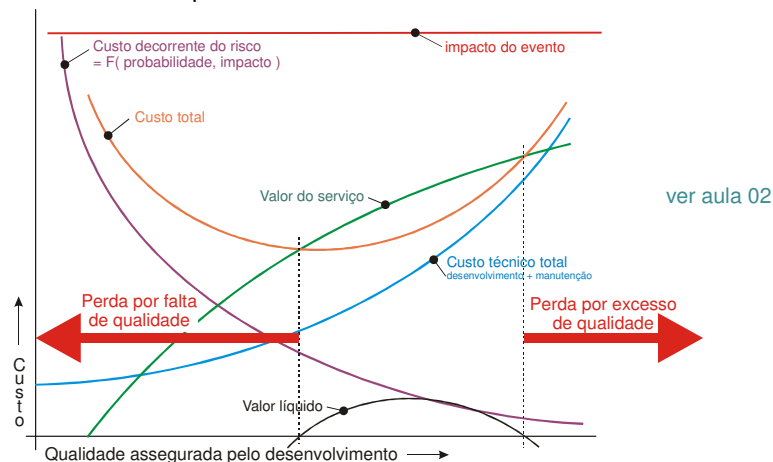
Arndt von Staa © LES/DI/PUC-Rio

5

Valor é função da qualidade, recordação



- O objetivo do desenvolvimento e da manutenção é assegurar que os **riscos sejam aceitáveis** e evitar a ocorrência de **lesões**
 - lesão é um impacto negativo que ocorreu e não foi observado
- Falhas relevantes possuem **riscos elevados**




Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

6

Laboratório de Engenharia de Software

Custo técnico total




- O **custo técnico total** leva em conta
 - o custo do desenvolvimento e da disponibilização
 - a soma dos custos de todos os eventos de manutenção e disponibilização
- **Manutenção:**
 - **corretiva** – elimina defeito causador de falha
 - **adaptativa** – altera o sistema sem afetar a sua funcionalidade
 - **perfectiva** – altera o sistema para introduzir melhorias de qualidade e funcionalidade
 - **preventiva** – altera o sistema para reduzir custos de manutenção **futuros**
- **Evolução:** desenvolvimento de uma nova versão

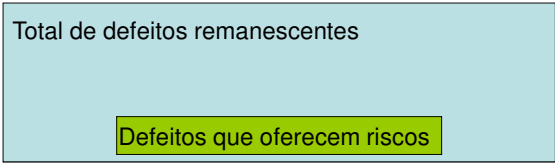
Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
7

Laboratório de Engenharia de Software

Observação da prática



- **Defeitos em sistemas usados em produção podem provocar repetidamente falhas**
 - quanto maior a frequência maior o interesse em removê-los
 - **amadurecimento**: eliminação correta de defeitos remanescentes
 - mas **manutenção** e **evolução** podem adicionar novos defeitos
- **Muitos defeitos remanescentes têm chance virtualmente 0 de serem exercitados**
 - poucos defeitos remanescentes possuem risco alto
 - estudo da IBM mostra que dos **defeitos remanescentes** somente 2% provocaram falhas recorrentes



Hatton, L.; "Exploring the Role of Diagnosis in Software Failure"; IEEE Software 18(4); Los Alamitos, CA: IEEE Computer Society; 2001; pags 34-39


Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
8

Laboratório de Engenharia de Software

Definição de qualidade, recordação

A **qualidade de um artefato** é um conjunto de **propriedades** a serem satisfeitas em **determinado grau**, de modo que o artefato **ofereça somente riscos aceitáveis** e **satisfaça** as **necessidades explícitas e implícitas** de todos os seus **interessados**

Adaptado da ABNT




9
Abr 2017
Arndt

Laboratório de Engenharia de Software

Objetivo do desenvolvimento


- Desenvolver e manter artefatos de qualidade satisfatória
 - a fidedignidade está relacionada com a capacidade de
 - **criar especificações** de qualidade satisfatória
 - **injetar muito poucos** defeitos **relevantes** ao desenvolver e manter
 - **detectar e eliminar** a (quase-) totalidade dos *defeitos relevantes* antes de por ou repor em uso
 - e conseguir isso de forma econômica:
 - alta **produtividade**
 - dimensão / esforço, ex. funcionalidades entregues por homem.hora
 - **custo compatível** com a *valia* (*value*) do artefato
 - nem sempre baixo custo de desenvolvimento é o desejável
 - » **baixo custo total = custo total técnico + custo total de uso**
 - o custo técnico total é fortemente afetado pela **densidade de defeitos inicial** ao desenvolver



valia - 3.Utilidade, préstimo, serventia, valência, valimento, valor; [Aurélio eletrônico]

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
10

Redução do custo técnico




Laboratório de Engenharia de Software

- **Injetar** poucos defeitos ao desenvolver e manter
 - elevada proficiência da equipe
 - contribui para uma significativa redução da injeção de defeitos nos variados artefatos gerados ao desenvolver ou manter
 - boa gestão e boas práticas
 - contribuem, por construção, para a redução da injeção de defeitos
 - boas ferramentas
 - contribuem para evitar ou para observar defeitos
 - padrões eficazes
 - eliminam ou reduzem, por construção, a frequência de determinadas classes de defeitos
 - desenvolvimento e integração incremental
 - contribuem para a redução de problemas nas especificações, na arquitetura e nos projetos
 - . . .

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
11

Controle da qualidade



Laboratório de Engenharia de Software

- Objetivo: identificar a existência de defeitos e inadequações
- Resultado: laudo de controle da qualidade
- Atividades:
 - revisão ou inspeção
 - medição e verificação estática
 - medição e verificação dinâmica
 - testes

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
12

Custo do teste



- Reduzir o **custo do teste**
 - concentrar nos casos de teste relevantes
 - reduzir o número de casos de teste pouco relevantes
 - aumentar a eficácia dos testes
 - prover capacidade de detectar falhas e diagnosticar os defeitos causadores
 - aumentar a eficiência dos testes
 - reduzir o número de vezes que testes são bem sucedidos ao serem **reexecutados**
 - um **teste bem sucedido** é um que encontra alguma falha
 - automatizar o que for possível
 - automação da execução dos casos de teste
 - elimina a necessidade de testadores humanos
 - automação da geração dos casos de teste úteis
 - reduz significativamente o esforço ao elaborar casos de teste úteis

Por que se preocupar com risco?



- Um artefato possui qualidade satisfatória caso satisfaça **plenamente** os anseios de todos os interessados, oferecendo **riscos justificavelmente aceitáveis** para cada propriedade
- Quanto maior for o **impacto**
 - muito menor deve ser a **probabilidade** dos defeitos causadores persistirem no sistema
 - logo: muito menor deve ser a **probabilidade** de deixar de observar o erro consequente desses defeitos

Por que se preocupar com risco?



- Potenciais falhas são
 - desconhecidas
 - se fossem conhecidas, poderiam ter sido removidas, óbvio
 - intrinsecamente inevitáveis
 - usuários são humanos e, portanto, podem
 - usar o sistema (ou o artefato) de forma incorreta
 - fornecer dados de forma incorreta
 - o falta de adequada proficiência e/ou a falibilidade dos desenvolvedores injeta defeitos
 - errar é humano, logo não se pode esperar que trabalho humano seja asseguradamente perfeito
 - especificações defeituosas levam a sistemas contendo defeitos
 - hardware e software de terceiros pode falhar
 - transmissão de dados entre equipamentos pode falhar
 - sistemas são sujeitos a “raios cósmicos”
 - falhas decorrentes de causas externas imprevisíveis ou desconhecidas

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

15

Tratamento do risco que se materializou

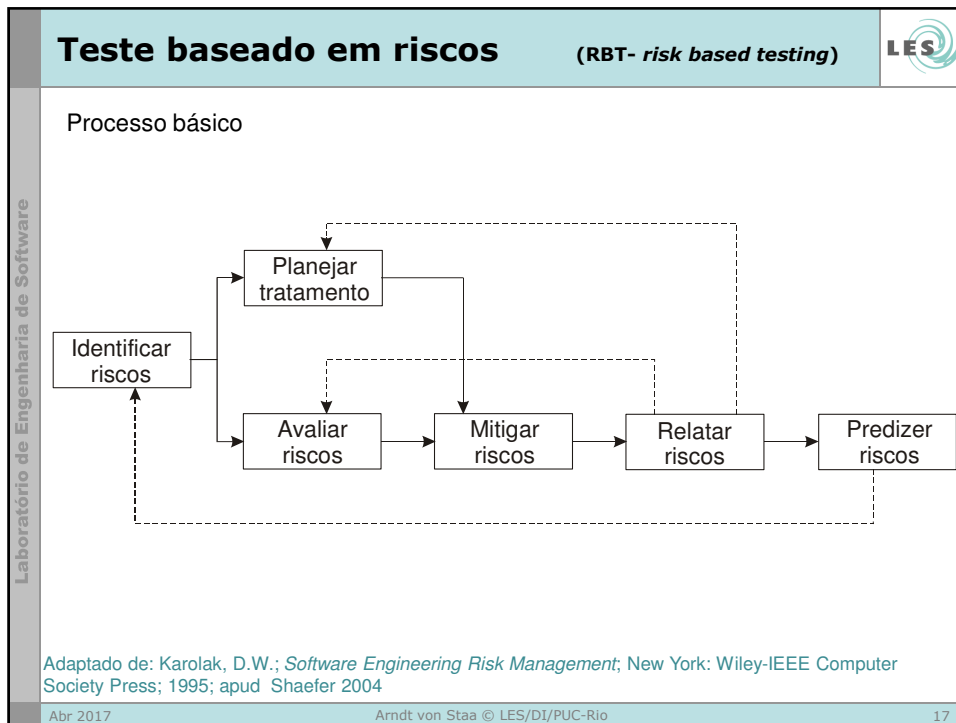



- Para tratar riscos ocorridos é necessário
 - ser capaz de **observar a correspondente falha**
 - os eventos de risco correspondem a algum comportamento ou estado diferente do que deveria ser segundo a especificação ou as expectativas do usuário, ou seja, um erro
 - ser capaz de **diagnosticar a causa da falha**
 - defeito contido no artefato
 - erro de uso não controlado
 - defeito em artefato disponibilizado por terceiros
 - agressão possível devido a alguma vulnerabilidade
 - ...
 - ser capaz de **controlar as consequências (dano) da falha**
 - ser capaz de **eliminar completamente a causa**
 - ser capaz de **por a versão corrigida em operação**

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

16




Identificar riscos 

- Criar um **catálogo genérico** dos potenciais riscos
 - este catálogo é utilizado para dirigir a identificação dos riscos a serem controlados no software em questão, exemplos:
 - é possível extravasar campos?
 - é possível injetar comandos SQL ao fornecer dados em um browser?
 - arquivos temporários permanecem disponíveis?
 - o conteúdo de arquivos excluídos continua disponível?
 - é possível acessar arquivos de outras aplicações?
 - é possível acessar arquivos na versão errada?
 - dados confidenciais podem ser acessados por não autorizados?
 - exemplo não computacional: o lixo gerado contém dados confidenciais
 - é possível vender mais de uma vez um mesmo produto?
 - é possível informar falta de estoque quando o item ainda existe em estoque?
 - ...

Abr 2017 Arndt von Staa © LES/DI/PUC-Rio 18

Identificar riscos




Laboratório de Engenharia de Software

- Selecionar o **catálogo do sistema** a partir dos casos de uso
 - perguntas do gênero:
 - para cada dado a ser fornecido pelo usuário
 - o que ocorre se o usuário fornece um dado errado?
 - o que vem a ser um dado errado?
 - o que vem a ser um dado **não plausível**?
 - usuário tenta fraudar
 - usuário tenta invadir
 - mais de N usuários tentam usar simultaneamente, qual é N ?
 - para cada link
 - link para URL não existente
 - para cada ação computacional
 - atividade realiza um cálculo errado
 - atividade cancela o processamento
 - usuário interrompe a transação antes de concluir
 - processamento é interrompido antes de concluir
 - » falta de energia, quebra ou falha de equipamento, logout
 - ...

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
19

Identificar riscos



Laboratório de Engenharia de Software


- Selecionar o **catálogo do sistema** baseado em critérios de qualidade
 - requisitos de disponibilidade
 - requisitos de desempenho
 - requisitos de escalabilidade
 - requisitos de capacidade
 - requisitos de manutenibilidade
 - requisitos de localizabilidade
 - requisitos de qualidade de engenharia
 - o que vem a ser qualidade de engenharia?
 - ...

Localizar um software: traduzir todas as mensagens, diálogos, menus, para um novo idioma de determinada sociedade (país), e adaptar o software à cultura correspondente

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
20

Laboratório de Engenharia de Software

Avaliar riscos




- Identificar os riscos relevantes do software em questão
 - reunião com cliente e usuários
 - deve resultar em um catálogo necessário e suficiente
- Proposta: criar uma planilha com as colunas:
 1. Nome da vulnerabilidade que caracteriza o risco
 2. Probabilidade potencial da ocorrência do risco
 - muito baixo 1, baixo 2, normal 3, alto 4, muito alto 5
 3. Impacto estimado
 - muito baixo 1, baixo 2, normal 3, alto 4, muito alto 5
 4. Relevância estimada
 - muito baixo 1, baixo 2, normal 3, alto 4, muito alto 5

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
21

Laboratório de Engenharia de Software

Avaliar riscos



- Modelo **pseudo** matemático ☺ do valor agregado

$$VA = \text{probabilidade} * \text{impacto} * \text{relevância}$$
- Probabilidade ::
 - {5 – muito alta , 4 – alta , 3 – aceitável , 2 – baixa , 1 – muito baixa}
- Impacto
 - {5 – muito alto , 4 – alto , 3 – aceitável , 2 – baixo , 1 – muito baixo}
- Relevância
 - {5 – muito alta , 4 – alta , 3 – aceitável , 2 – baixa , 1 – muito baixa}
- Nível do risco
 - {125 – 101 desastroso , 100 – 76 altíssimo , 75 – 51 muito alto , 50 – 26 alto , 25 – 10 aceitável , 9 – 1 irrelevante }


Chutologia? Qual seria um modelo melhor?

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
22

Laboratório de Engenharia de Software

Avaliar riscos

- Ordenar a planilha em ordem decrescente de VA
- Determinar os pontos de corte
 - Evitar sempre $VA \geq 75$,
 - Evitar se possível $25 < VA < 75$
 - Ignorar $VA \leq 25$
- Desenvolver testes cuidadosos
 - para os riscos a “evitar sempre”
 - se existirem recursos
 - para os riscos a “evitar se possível”
- É conveniente a planilha existir antes de se iniciar o desenvolvimento
 - reduz o esforço para controlar coisas de baixo risco
 - conhecer o **risco influencia a arquitetura e o projeto**




Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
23

Laboratório de Engenharia de Software

Planejar tratamento


- Desenvolver planos de contingência
 - o que fazer se o evento identificado pelo risco ocorrer?
 - quando em teste, ex.
 - gerar uma solicitação de correção emergencial
 - ou gerar uma solicitação para uma futura versão
 - quando em uso
 - procedimentos de registro e tratamento de incidentes (falhas) observados
- Desenvolver planos para **mitigar**
 - como proceder para manter o dano sob controle no caso do evento de um risco ocorrer?
 - quando em teste
 - quando em uso



Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
24

Laboratório de Engenharia de Software

Mitigar riscos




- Mitigar o risco tem por objetivo
 - prevenir (impedir) a ocorrência do evento que oferece risco
 - ou, se isto não for possível, manter sob controle o impacto consequente da ocorrência do evento
- Para poder mitigar é necessário ser capaz de observar a ocorrência do evento
 - caso a mitigação seja feita corretamente, não ocorrerão lesões
 - recordação: lesão é a ocorrência de um dano não conhecido
 - mas poderão ocorrer impactos observáveis ou mensuráveis

Mitigar: abrandar, suavizar, diminuir (o impacto) [Aurélio eletrônico]

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
25

Laboratório de Engenharia de Software

Mitigar riscos



- Para poder mitigar é necessário ser capaz de observar erros
 - como observar que o evento ocorreu ou está prestes a ocorrer?
 - como proceder para evitar a ocorrência do evento?
 - procurar tornar inexistentes os defeitos associados ao risco
 - como proceder para mostrar que o evento efetivamente tem baixa probabilidade de ocorrer?
 - testar, inspecionar
 - para fins de teste, como proceder para simular ou provocar a ocorrência do evento?
 - caso ocorra o evento, proceder como planejado
 - os grandes desastres tendem a ser a consequência da composição de vários eventos e de erros humanos ao tratar alguns deles
 - muitas vezes espera-se de humanos comportamento não humano, por exemplo, humanos podem errar, erram mais quando sob stress
 - ou seja, erro humano pode ser induzido por sistema com usabilidade inadequada

Abr 2017
Arndt von Staa © LES/DI/PUC-Rio
26

Relatar riscos



- Relatam-se os
 - eventos que ocorreram
 - podem-se observar novos riscos
 - a frequência com que ocorreram
 - as consequências da ocorrência
 - impactos (danos) observados
- Durante os testes devem ser medidos:
 - número de defeitos e vulnerabilidades encontrados
 - número de defeitos por funcionalidade
 - número de ocorrências de cada evento
 - tempo (horas, ou fração) gastas para encontrar a falha
 - tempo (horas, ou fração) gastas para eliminar o defeito
 - classificação do defeito
 - o evento a que corresponde

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

27

Predizer riscos



- Baseado em medições e observações realizadas (passado) prediz-se a possibilidade da ocorrência dos eventos associados a riscos
 - como consequência da predição pode-se concluir que determinadas funcionalidades (ou componentes) devem ser revistas
 - a mesma coisa aplica-se à arquitetura e aos projetos


Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

28

Laboratório de Engenharia de Software

Manutenção do catálogo



- Um catálogo incompleto leva à perda de confiança
 - durante as diversas etapas do processo podem ser identificados novos riscos
 - devem ser incorporados ao catálogo
 - ao ler literatura sobre riscos, novos riscos podem ser identificados
 - devem ser incorporados ao catálogo
- Um catálogo muito extenso torna-se um estorvo
 - de tempos em tempos o catálogo deve ser revisto
 - riscos não mais observáveis devem ser transferidos para uma região de riscos “obsoletos” (deprecados)
 - riscos similares devem ser fundidos em um único
 - descrições de riscos devem ser revistas para assegurar atualidade e coerência com a terminologia atual

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

29

Laboratório de Engenharia de Software

Bibliografia



- A presente aula foi fortemente baseada nos textos a seguir
 - Amland, S.; *Risk Based Testing and Metrics*; 5th International Conference EuroSTAR '99; 1999, Barcelona, Spain
 - Bach, J.; *Heuristic Risk-Based Testing*; Software Testing and Quality Engineering Magazine, 11/99
 - Schaefer, H.; *Risk based testing, how to choose what to test more and less*; Notas de palestra; Software Test Consulting, Norway; 2004
 - Teunissen, R.; *Risk Based Test Strategy*; Notas de palestra; São Paulo; 2010; Polteq IT Services BV
- Arnuphaptrairong, T.; “Top Ten Lists of Software Project Risks: Evidence from the Literature Survey”; Volume I; IMECS 2011 International MultiConference of Engineers and Computer Scientists; 2011; online
- James R. Persse, J.R.; A Basic Approach to ITIL Service Operation; Atlanta: Tree Of Press; 2010; Kindle Edition
- Steven Christey Coley, Ryan P. Glenn, Janis E. Kenderdine, and John M. Mazella; editors; CWE Common Weakness Enumeration; version 2.8; 2014

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

30

Laboratório de Engenharia de Software

LES

Fim

Abr 2017

Arndt von Staa © LES/DI/PUC-Rio

31