Teste Baseado em Risco
Análise de Risco de Software

Auri Marcelo Rizzo Vincenzi[1], Márcio Eduardo Delamaro[2] e José Carlos Maldonado[2]

[1]Instituto de Informática
Universidade Federal de Goiás

[2]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Embora existam diferentes maneiras de se conduzir Análise de Risco, em diferentes fases do processo de desenvolvimento, este material foca na estratégia proposta por Craig e Jaskiel (2002) com foco na Análise de Risco relacionada as funções do produto em teste.

Organização

Introdução (1)

Tom Gilb

"Se você não atacar os riscos de forma ativa, eles o atacarão."

- Não há garantia que um sistema de software é correto
- Falhas podem surgir das mais variáveis direções
- Defeitos latentes permanecem imperceptíveis por anos e fazem o software falhar de modo inesperado
- Esse efeito é potencializado à medida que mudanças em interfaces ou protocolos, em uma parte do sistema, começam a interferir em outra

Introdução (2)

- Aumento no número de usuários podem sobrecarregar o sistema
- Alterações no modelo de negócios podem fazê-los a usar o sistema de modo nunca antes imaginado
- Alterações no ambiente de execução também impõem riscos para a aplicação
- Tudo isso pode minar um bom projeto de software, gerando problemas na implementação e operação do software

Introdução (3)

- O propósito da Análise de Risco de Software é determinar o que testar, a prioridade do teste e a abrangência dos testes

Introdução (3)

- O propósito da Análise de Risco de Software é determinar o que testar, a prioridade do teste e a abrangência dos testes
- **Quem** deve fazer?

Introdução (3)

- O propósito da Análise de Risco de Software é determinar o que testar, a prioridade do teste e a abrangência dos testes
- **Quem** deve fazer?
  - Idealmente, uma equipe de especialistas multidisciplinar

Introdução (3)

- O propósito da Análise de Risco de Software é determinar o que testar, a prioridade do teste e a abrangência dos testes
- **Quem** deve fazer?
  - Idealmente, uma equipe de especialistas multidisciplinar
  - Desenvolvedores, testadores, usuários, clientes, vendedores e outros interessados

Introdução (4)

- **Quando** deve ser feita?

Introdução (4)

- **Quando** deve ser feita?
  - O quanto antes no processo de desenvolvimento

Introdução (4)

- **Quando** deve ser feita?
  - O quanto antes no processo de desenvolvimento
  - Pode ser logo após os requisitos de alto nível serem definidos

Introdução (4)

- **Quando** deve ser feita?
  - O quanto antes no processo de desenvolvimento
  - Pode ser logo após os requisitos de alto nível serem definidos
  - Lembre-se que os resultados podem ser revistos durante o projeto

Introdução (4)

- **Quando** deve ser feita?
  - O quanto antes no processo de desenvolvimento
  - Pode ser logo após os requisitos de alto nível serem definidos
  - Lembre-se que os resultados podem ser revistos durante o projeto
- **Como** deve ser feita?

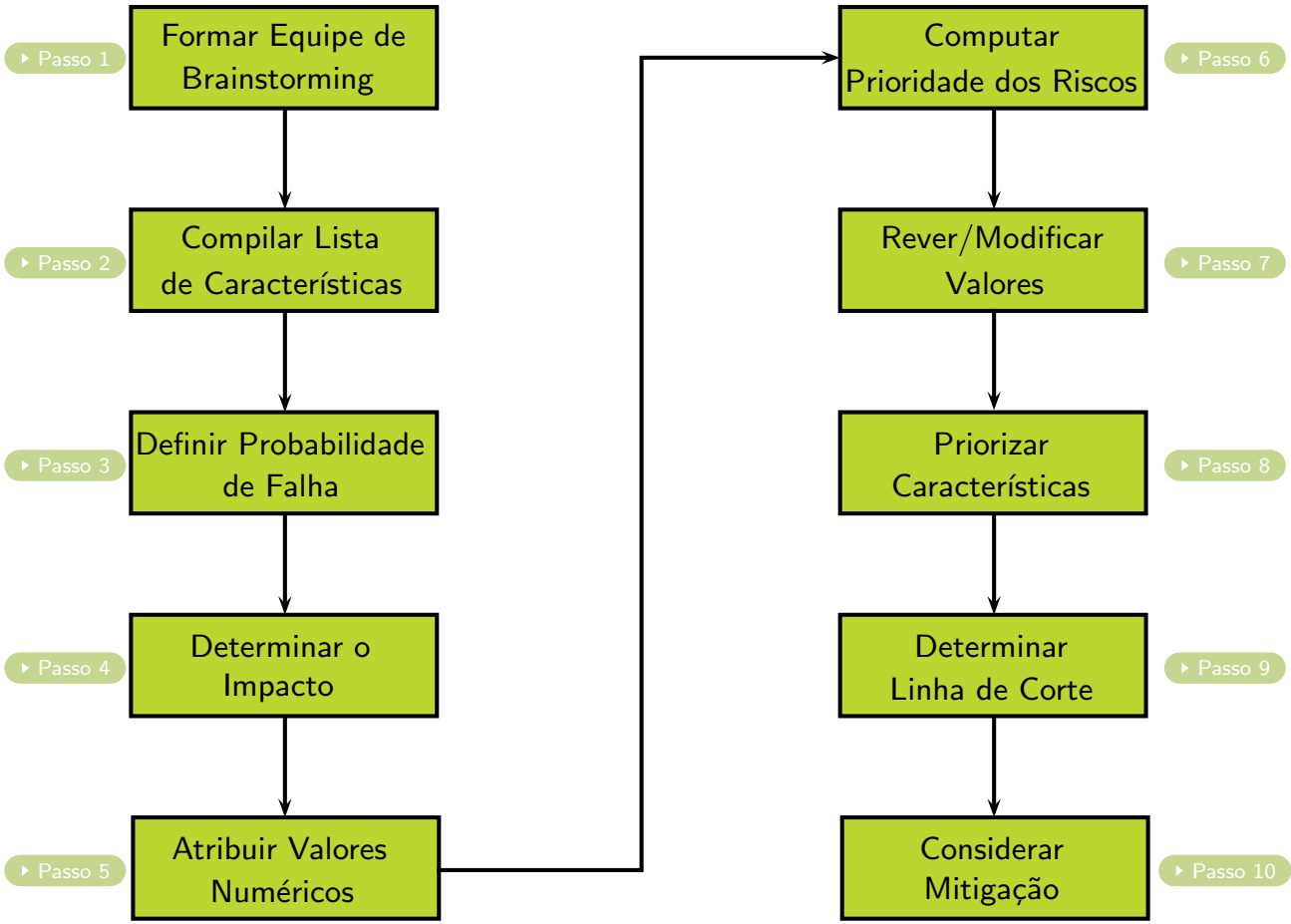
Introdução (4)

- **Quando** deve ser feita?
  - O quanto antes no processo de desenvolvimento
  - Pode ser logo após os requisitos de alto nível serem definidos
  - Lembre-se que os resultados podem ser revistos durante o projeto
- **Como** deve ser feita?
  - Craig e Jaskiel (2002) propõem uma sequência de dez passos para se conduzir a análise de risco

Introdução (4)

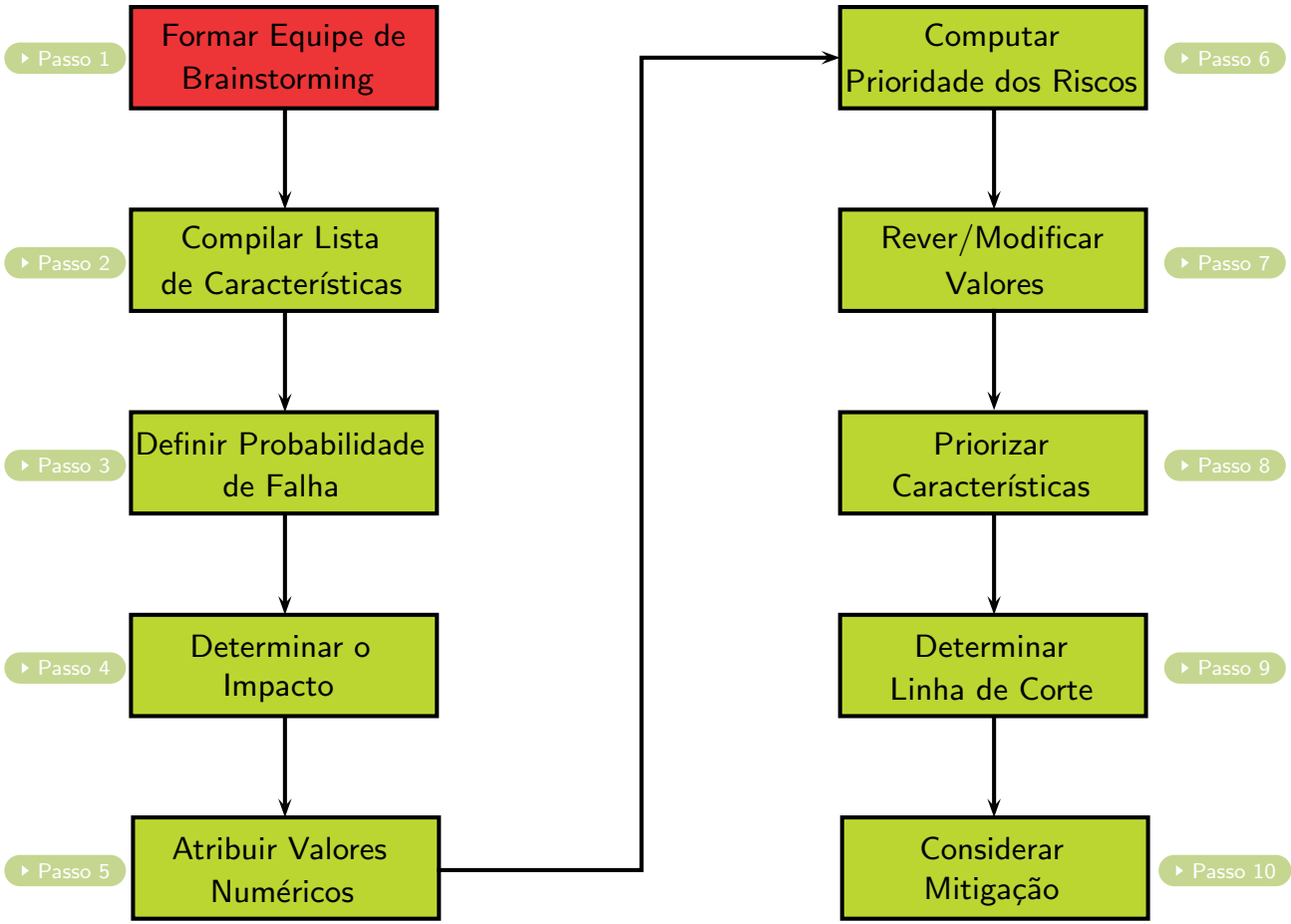
- **Quando** deve ser feita?
  - O quanto antes no processo de desenvolvimento
  - Pode ser logo após os requisitos de alto nível serem definidos
  - Lembre-se que os resultados podem ser revistos durante o projeto
- **Como** deve ser feita?
  - Craig e Jaskiel (2002) propõem uma sequência de dez passos para se conduzir a análise de risco
  - Algumas organizações combinam vários passos em um, enquanto outras podem incluir novos passos

Síntese do Processo



Adaptado de Craig e Jaskiel (2002)

Passo 1: Formar Equipe de Brainstorming



Adaptado de Craig e Jaskiel (2002)

Passo 1: Formar Equipe de Brainstorming

- Formar a equipe que irá conduzir o trabalho
- Importante incluir na equipe usuários, desenvolvedores, testadores, vendedores, representantes do cliente, pessoal do suporte e outros interessados com conhecimento do negócio e/ou produto
- Várias equipes falham na condução da análise de risco em função da escolha de pessoas erradas para sua composição

Passo 1: Formar Equipe de Brainstorming

Parte 1

- O propósito da primeira parte de uma sessão de *brainstorming* é aumentar o número de ideias geradas pelo grupo
- Como regras gerais para o bom andamento do processo considere:
  - Não permitir crítica ou debate
  - Deixar a imaginação fluir
  - Prefira quantidade
  - Altere e combine ideias

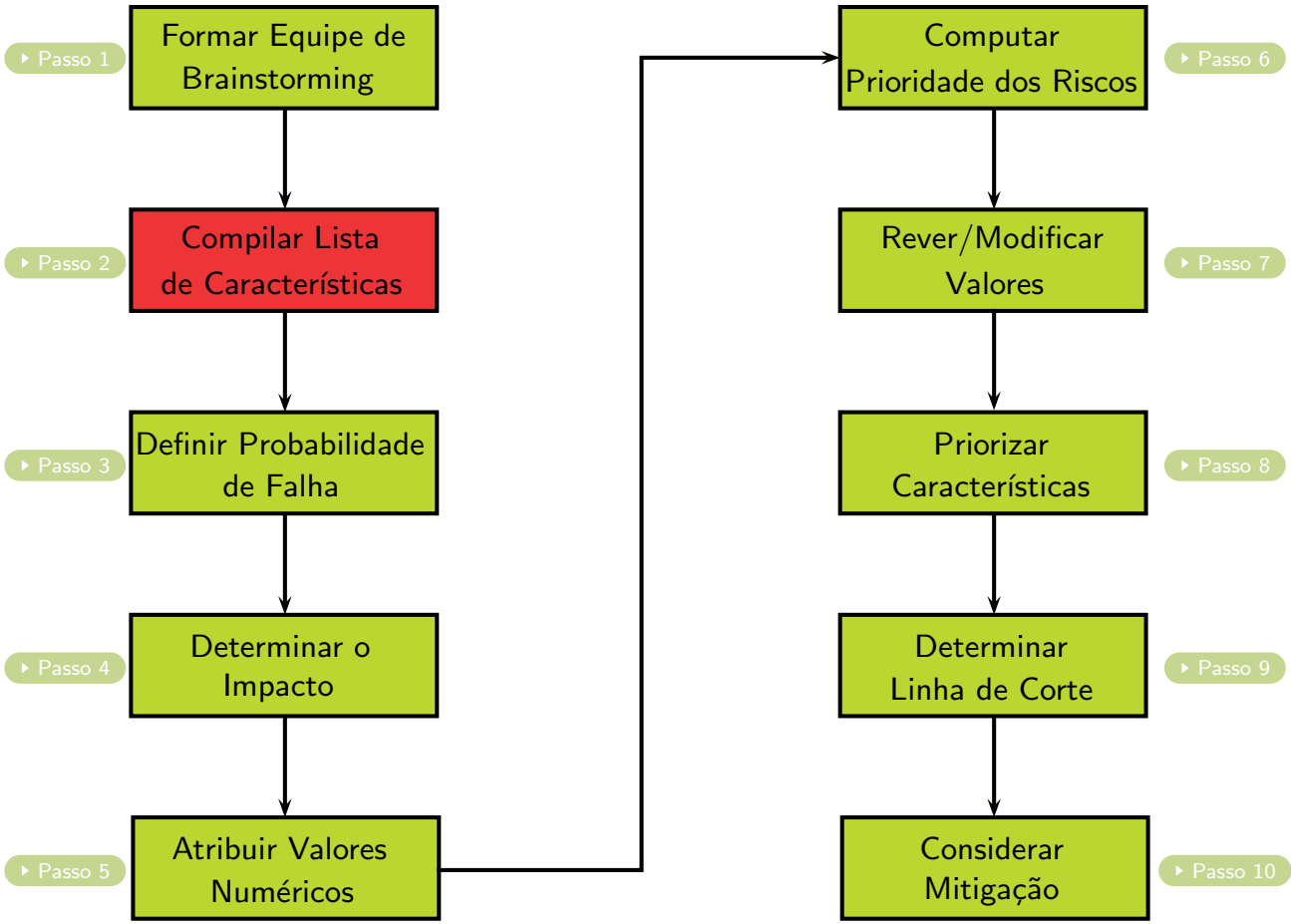
Passo 1: Formar Equipe de Brainstorming

Parte 2

- O propósito da segunda parte de uma sessão de *brainstorming* é reduzir a lista de ideia a um tamanho gerenciável
- Como regras gerais para o bom andaimando do processo considere:
  - Votação com defesas orais
  - Combinando ideias
  - Aplicando critérios
  - Pontuando e classificação
- No processo proposto, ênfase será dada à redução da lista da sessão de *brainstorming*

▸ Síntese do Processo

Passo 2: Compilar Lista de Característica



Adaptado de Craig e Jaskiel (2002)

Passo 2: Compilar Lista de Característica

A equipe de brainstorming deve:

- Obter toda documentação disponível: especificação de requisitos, especificação funcional, solicitações de mudanças, relatório de defeitos, documentos de projeto, dentre outros
- Compilar uma lista de características, atributos ou regras de negócio para todo sistema
- A lista de atributos pode incluir, por exemplo: acessibilidade, disponibilidade, compatibilidade, manutenibilidade, desempenho, confiabilidade, escalabilidade, segurança e usabilidade

Passo 2: Compilar Lista de Característica

A equipe de brainstorming deve:

- Posteriormente, se o tempo permitir, a lista pode ser complementada e detalhada
- Caso o esforço de desenvolvimento esteja concentrado em uma parte específica ou num subsistema, a análise pode focar apenas nessa área
- Entretanto, as interfaces com outras partes do sistema devem ser identificadas pois são importantes para os testes

Exemplos Utilizado (1)

Automated Teller Machine (ATM) - Caixa Eletrônico

- Para ilustrar o processo de análise de risco será utilizada a especificação de um ATM
- De conhecimento da maioria das pessoas
- Apresenta uma variedade de características

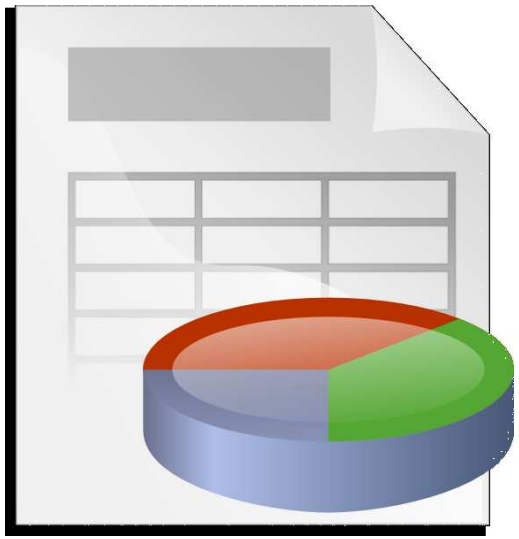
Exemplos Utilizado (2)

Automated Teller Machine (ATM) - Caixa Eletrônico

- Algumas características identificadas incluem:
  - saque, depósito, saldo, extrato, transferência, recargas de celular, pagamento
  - quais mais?
- Alguns atributos incluem:
  - usabilidade, desempenho e segurança
  - quais mais?

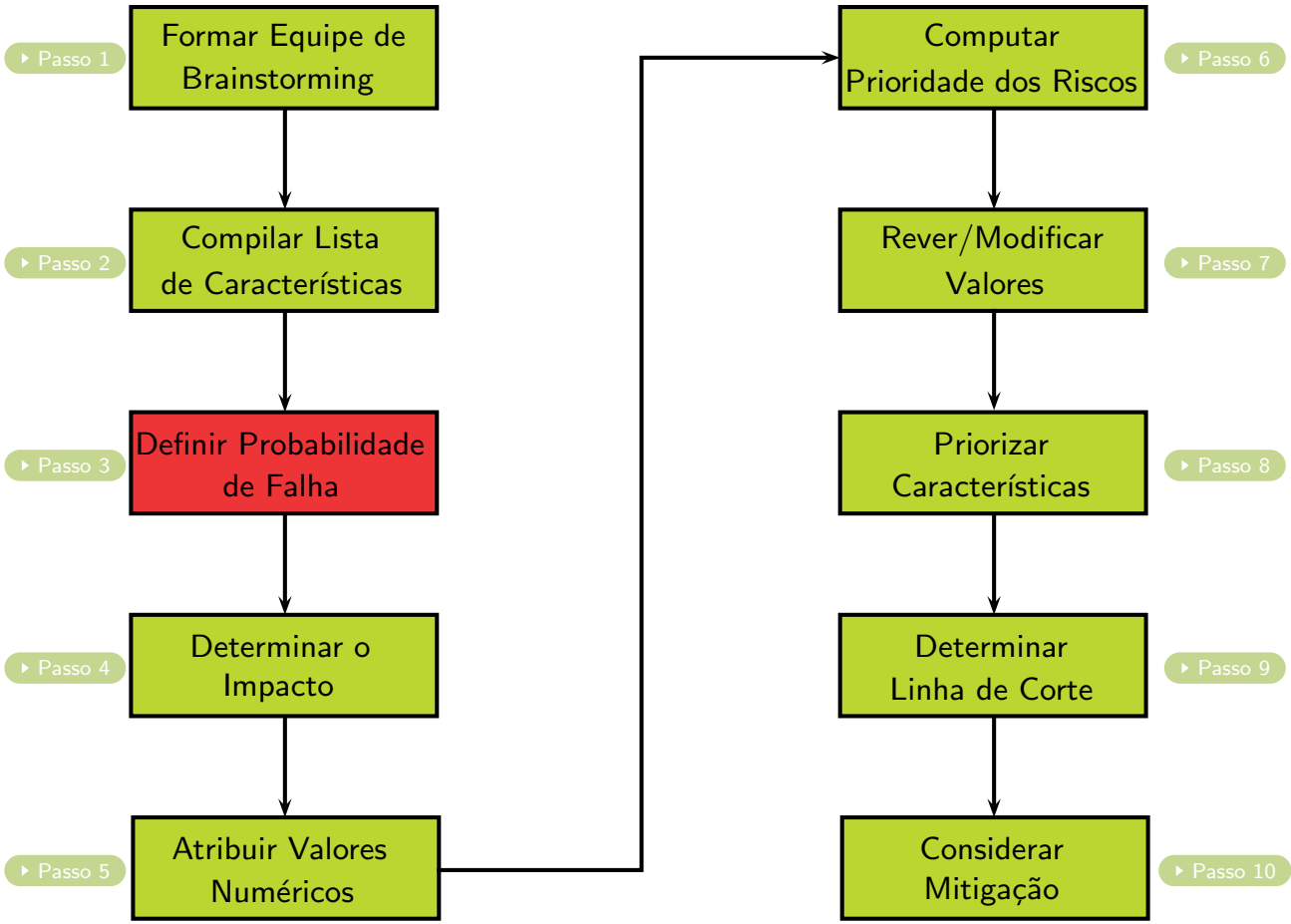
Passo 2: Compilar Lista de Característica

Exemplo ATM



▸ Síntese do Processo

Passo 3: Definir Probabilidade de Falha



Adaptado de Craig e Jaskiel (2002)

Passo 3: Definir Probabilidade de Falha

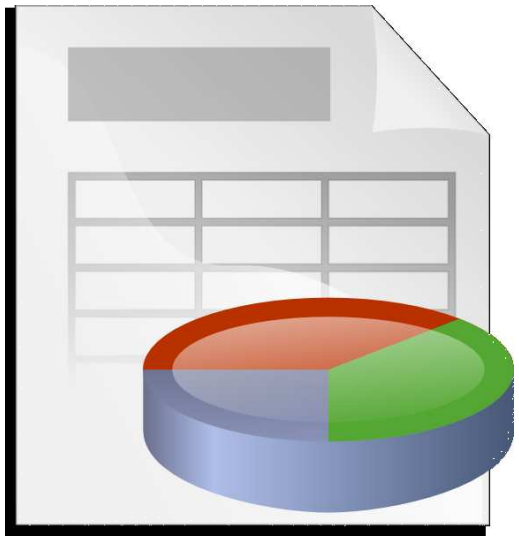
- Neste passo do processo da análise de risco são associados indicadores relacionados com a probabilidade que determinada característica ou atributo tem de falhar
- Uma forma de se fazer isso é associando valores de probabilidade: Alta, Média e Baixa
- A equipe deve atribuir valores respondendo a seguinte pergunta: **Com base no nosso conhecimento do sistema, qual a probabilidade desta característica ou atributo falhar?**
- Em geral, indicadores de probabilidade são relacionados a características sistêmicas como complexidade, número de interfaces, dentre outras

Passo 3: Definir Probabilidade de Falha

- Para todas as características ou atributos deve ser atribuída uma probabilidade
- Mesmo que alguns membros da equipe não concordem com o valor atribuído é importante ter algo anotado o quanto antes, estimulando o restante do processo
- Não é necessário que todos concordem com o valor nesse ponto do processo
- Posteriormente, caso haja tempo, é possível retornar e modificar tais valores

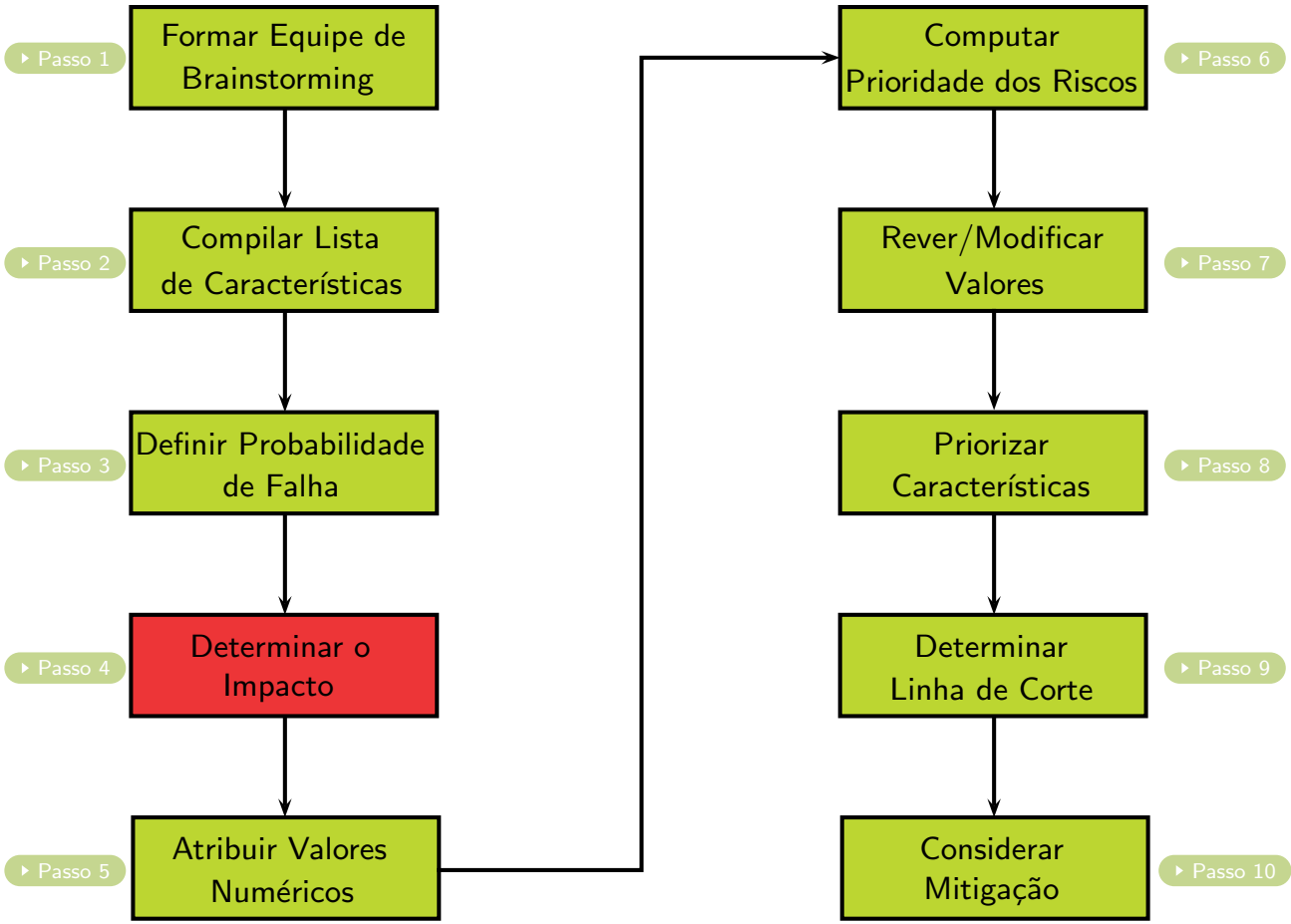
Passo 3: Definir Probabilidade de Falha

Exemplo ATM



▸ Síntese do Processo

Passo 4: Determinar o Impacto



Adaptado de Craig e Jaskiel (2002)

Passo 4: Determinar o Impacto

- Usando o mesmo esquema de ranqueamento, a equipe de brainstorming deve se fazer a seguinte pergunta: **Qual o impacto para o usuário se essa característica ou atributo não operar corretamente?**

Passo 4: Determinar o Impacto

- Usando o mesmo esquema de ranqueamento, a equipe de brainstorming deve se fazer a seguinte pergunta: **Qual o impacto para o usuário se essa característica ou atributo não operar corretamente?**
- Embora a equipe possa estar tentada em considerar o impacto da falha no desenvolvimento do resto do sistema, é importante resistir a isso nesse ponto da análise de risco

Passo 4: Determinar o Impacto

- Usando o mesmo esquema de ranqueamento, a equipe de brainstorming deve se fazer a seguinte pergunta: **Qual o impacto para o usuário se essa característica ou atributo não operar corretamente?**
- Embora a equipe possa estar tentada em considerar o impacto da falha no desenvolvimento do resto do sistema, é importante resistir a isso nesse ponto da análise de risco
- O foco deve ser na análise de características e atributos que afetam diretamente o usuário e não necessariamente o esforço de teste

Passo 4: Determinar o Impacto

- Usando o mesmo esquema de ranqueamento, a equipe de brainstorming deve se fazer a seguinte pergunta: **Qual o impacto para o usuário se essa característica ou atributo não operar corretamente?**
- Embora a equipe possa estar tentada em considerar o impacto da falha no desenvolvimento do resto do sistema, é importante resistir a isso nesse ponto da análise de risco
- O foco deve ser na análise de características e atributos que afetam diretamente o usuário e não necessariamente o esforço de teste
- O impacto no teste será resolvido posteriormente

Passo 4: Determinar o Impacto

- O usuário é especialmente importante nessa fase da análise de risco

Passo 4: Determinar o Impacto

- O usuário é especialmente importante nessa fase da análise de risco
- O problema é que ele, em geral, insiste que todas as características tem um alto impacto em caso de falha

Passo 4: Determinar o Impacto

- O usuário é especialmente importante nessa fase da análise de risco
- O problema é que ele, em geral, insiste que todas as características tem um alto impacto em caso de falha
- O problema aumenta à medida que vários usuários com diferentes interesses participam do processo

Passo 4: Determinar o Impacto

- O usuário é especialmente importante nessa fase da análise de risco
- O problema é que ele, em geral, insiste que todas as características tem um alto impacto em caso de falha
- O problema aumenta à medida que vários usuários com diferentes interesses participam do processo
- Entretanto, isso não ajuda em nada em priorizar os riscos já que todos teriam o mesmo nível de criticalidade

Passo 4: Determinar o Impacto

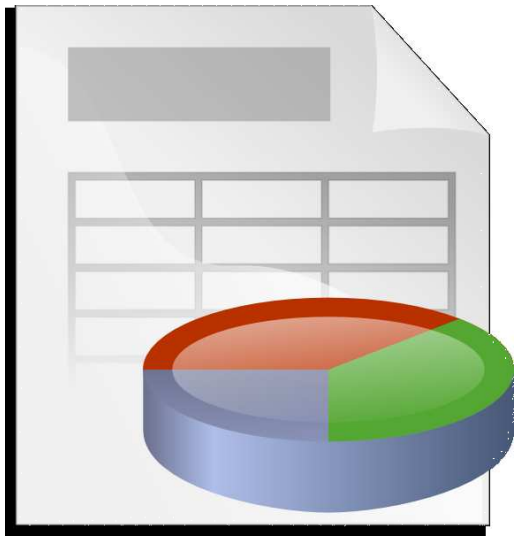
- O usuário é especialmente importante nessa fase da análise de risco
- O problema é que ele, em geral, insiste que todas as características tem um alto impacto em caso de falha
- O problema aumenta à medida que vários usuários com diferentes interesses participam do processo
- Entretanto, isso não ajuda em nada em priorizar os riscos já que todos teriam o mesmo nível de criticalidade
- Caso isso venha a ocorrer, uma alternativa é limitar o número de valores Alto, Médio e Baixo que podem ser atribuídos

Passo 4: Determinar o Impacto

- O usuário é especialmente importante nessa fase da análise de risco
- O problema é que ele, em geral, insiste que todas as características tem um alto impacto em caso de falha
- O problema aumenta à medida que vários usuários com diferentes interesses participam do processo
- Entretanto, isso não ajuda em nada em priorizar os riscos já que todos teriam o mesmo nível de criticalidade
- Caso isso venha a ocorrer, uma alternativa é limitar o número de valores Alto, Médio e Baixo que podem ser atribuídos
- Em geral, testadores experientes são excelentes em determinar o impacto de falhas

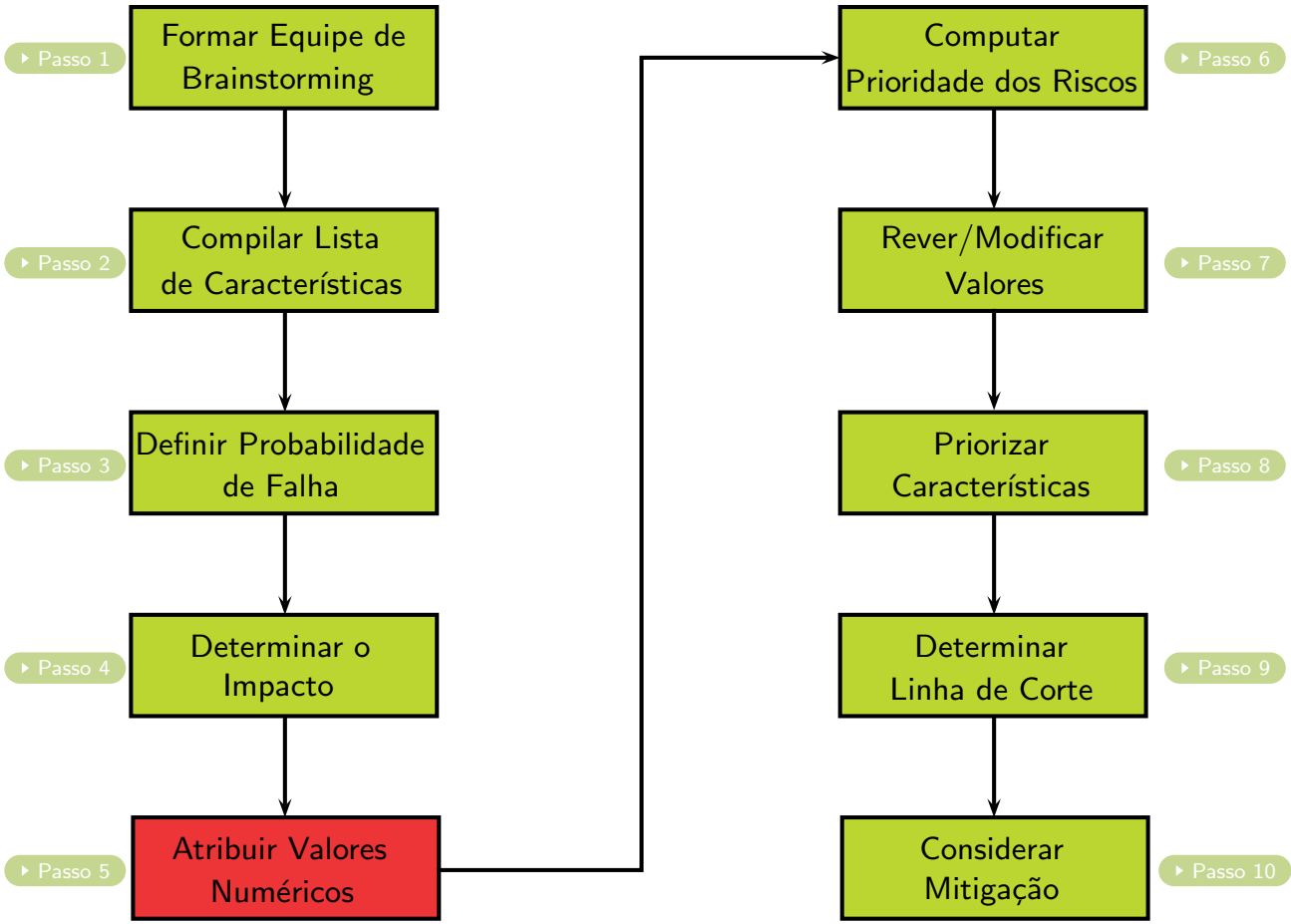
Passo 4: Determinar o Impacto

Exemplo ATM



▸ Síntese do Processo

Passo 5: Atribuir Valores Numéricos



Adaptado de Craig e Jaskiel (2002)

Passo 5: Atribuir Valores Numéricos

- Nesse passo, são atribuídos valores numéricos para os indicadores Alto, Médio e Baixo

Passo 5: Atribuir Valores Numéricos

- Nesse passo, são atribuídos valores numéricos para os indicadores Alto, Médio e Baixo
- Qualquer sequência de valores descendentes pode ser usada

Passo 5: Atribuir Valores Numéricos

- Nesse passo, são atribuídos valores numéricos para os indicadores Alto, Médio e Baixo
- Qualquer sequência de valores descendentes pode ser usada
- Por simplicidade, serão usados os valores 3, 2 e 1, respectivamente

Passo 5: Atribuir Valores Numéricos

- Nesse passo, são atribuídos valores numéricos para os indicadores Alto, Médio e Baixo
- Qualquer sequência de valores descendentes pode ser usada
- Por simplicidade, serão usados os valores 3, 2 e 1, respectivamente
- Algumas pessoas preferem atribuir 10, 3 e 1

Passo 5: Atribuir Valores Numéricos

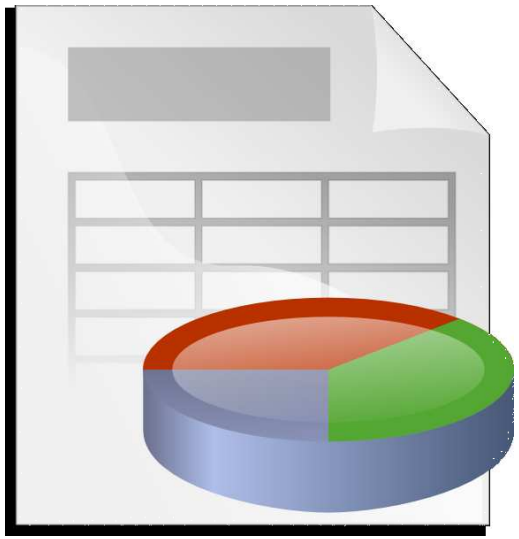
- Nesse passo, são atribuídos valores numéricos para os indicadores Alto, Médio e Baixo
- Qualquer sequência de valores descendentes pode ser usada
- Por simplicidade, serão usados os valores 3, 2 e 1, respectivamente
- Algumas pessoas preferem atribuir 10, 3 e 1
- Essa deve ser uma decisão organizacional e, uma vez determinada a escala ela deve ser mantida durante todo o processo de análise de risco

Passo 5: Atribuir Valores Numéricos

- Em caso de sistemas de segurança crítica é importante que aquelas características que podem resultar em morte ou perdas financeiras sempre recebam alta prioridade nos testes mesmo se o risco global foi baixo devido a uma excepcional baixa probabilidade de falha

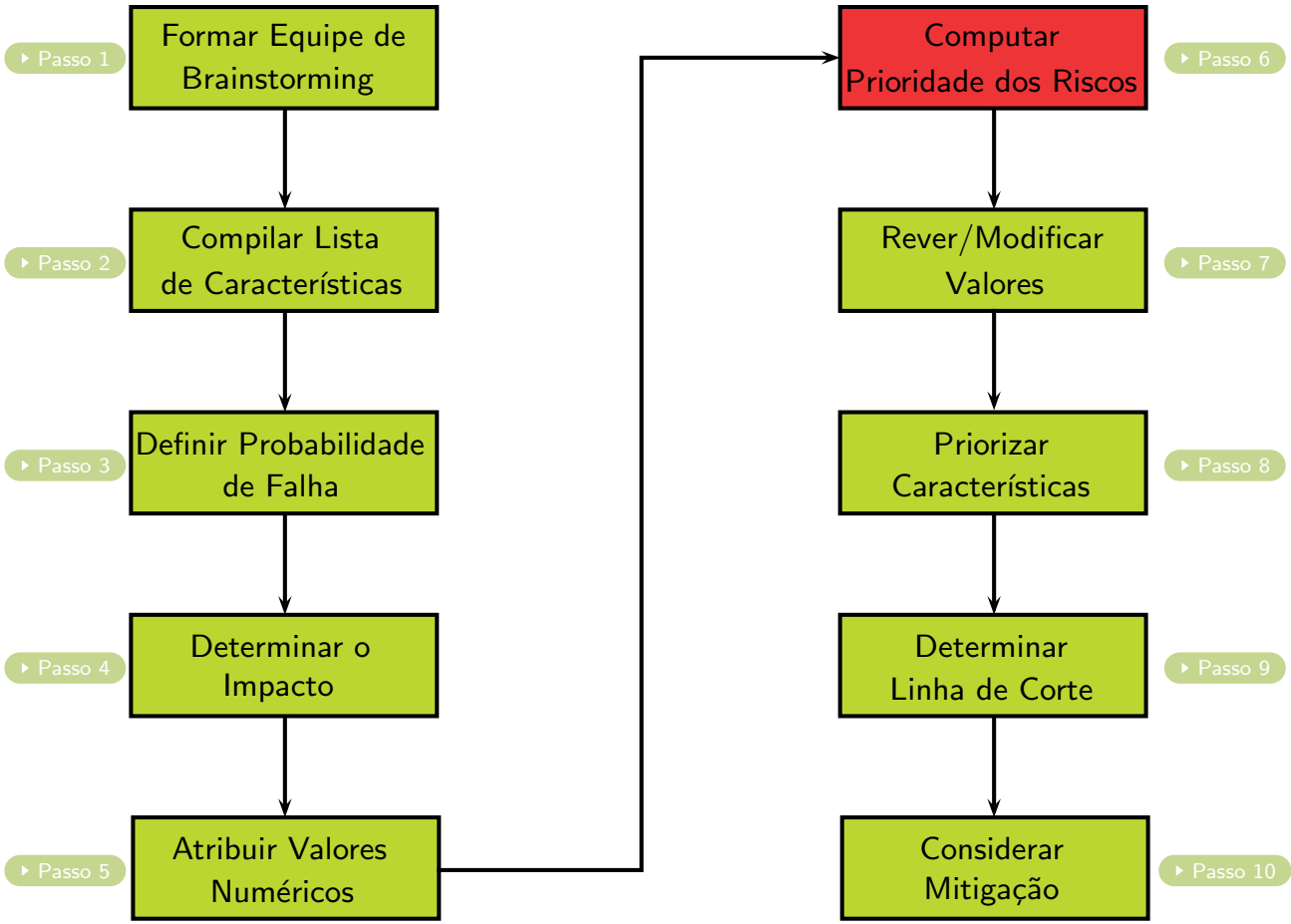
Passo 5: Atribuir Valores Numéricos

Exemplo ATM



▸ Síntese do Processo

Passo 6: Computar Prioridade dos Riscos



Adaptado de Craig e Jaskiel (2002)

Passo 6: Computar Prioridade dos Riscos

- Uma vez atribuídos valores para probabilidade de falha e impacto da falha, esses valores são somados

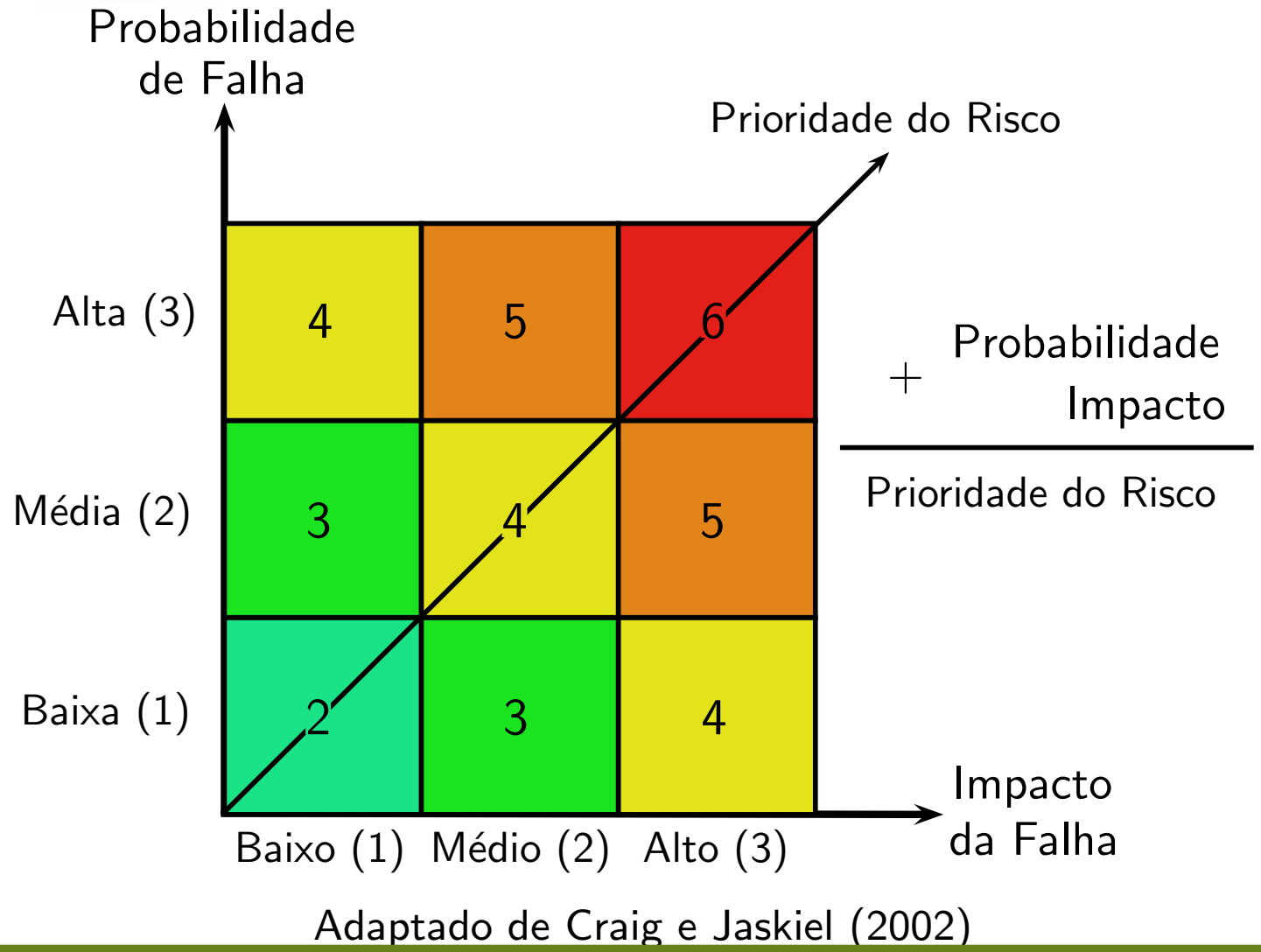
Passo 6: Computar Prioridade dos Riscos

- Uma vez atribuídos valores para probabilidade de falha e impacto da falha, esses valores são somados
- Considerando Alto=3, Médio=2 e Baixo=1 cinco níveis de prioridade de risco são possíveis: 6, 5, 4, 3 e 2

Passo 6: Computar Prioridade dos Riscos

- Uma vez atribuídos valores para probabilidade de falha e impacto da falha, esses valores são somados
- Considerando Alto=3, Médio=2 e Baixo=1 cinco níveis de prioridade de risco são possíveis: 6, 5, 4, 3 e 2
- Algumas organizações preferem multiplicar os valores ao invés de somar, ampliando a área de risco

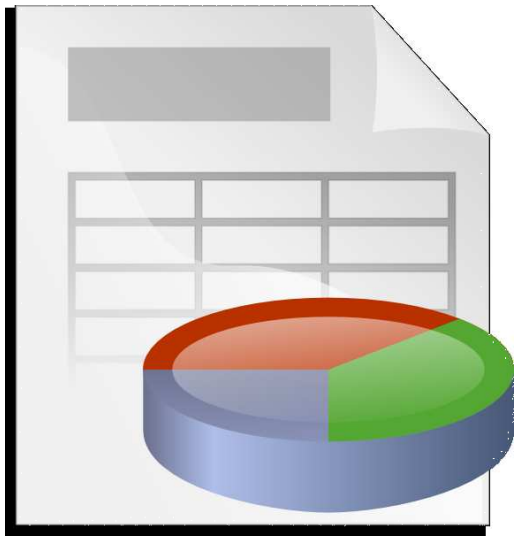
Passo 6: Computar Prioridade dos Riscos



Adaptado de Craig e Jaskiel (2002)

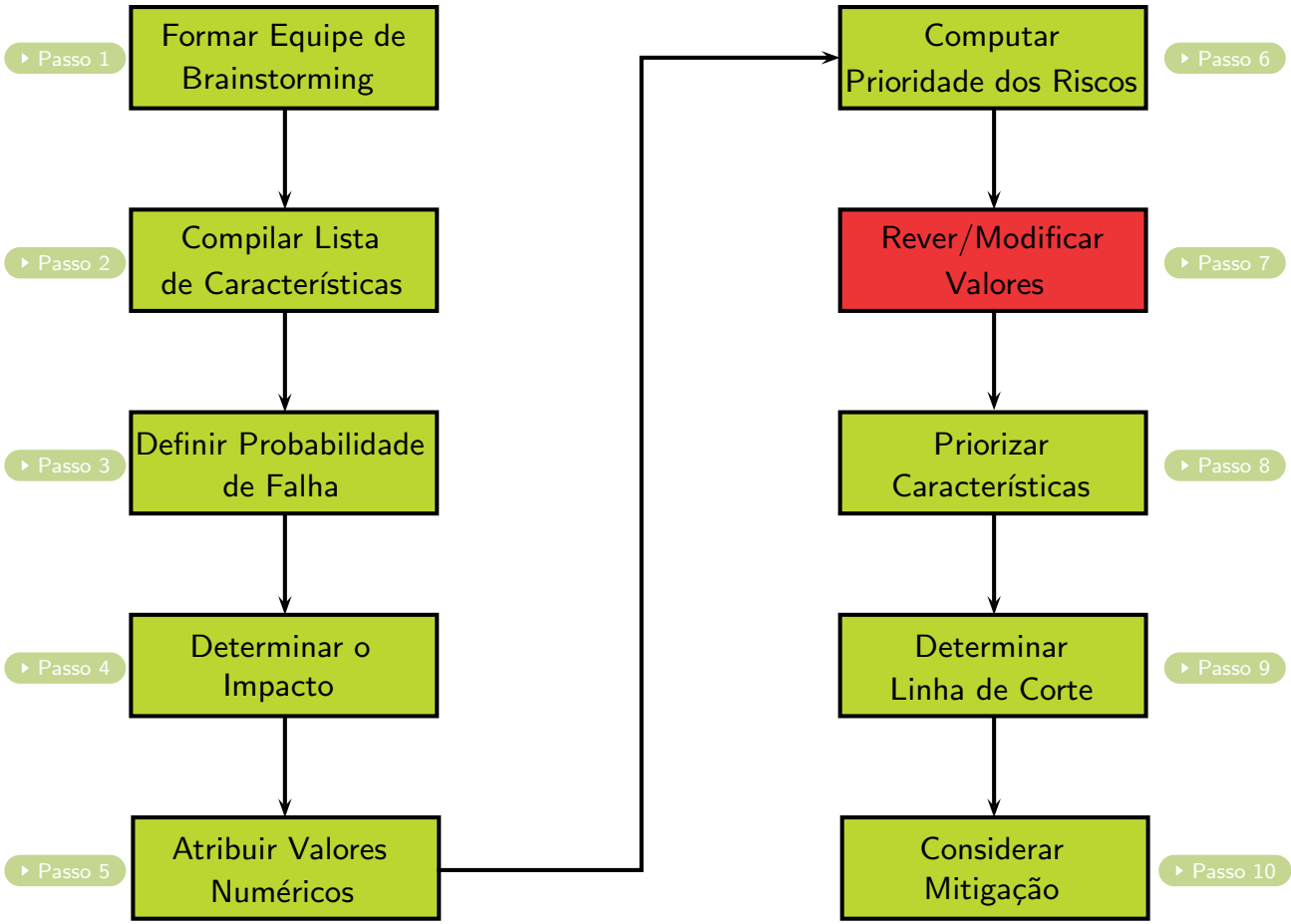
Passo 6: Computar Prioridade dos Riscos

Exemplo ATM



▸ Síntese do Processo

Passo 7: Rever/Modificar Valores



Adaptado de Craig e Jaskiel (2002)

Passo 7: Rever/Modificar Valores

- No ‣ Passo 3, valores foram atribuídos às características e aos atributos do software

Passo 7: Rever/Modificar Valores

- No ▸ Passo 3, valores foram atribuídos às características e aos atributos do software
- Atingido o consenso sobre os valores atribuídos durante a sessão de brainstorming, é possível alterar esses valores **com base em informações adicionais** ou **análises** obtidas

Passo 7: Rever/Modificar Valores

- No Passo 3, valores foram atribuídos às características e aos atributos do software
- Atingido o consenso sobre os valores atribuídos durante a sessão de brainstorming, é possível alterar esses valores **com base em informações adicionais** ou **análises** obtidas
- Exemplos de indicadores de probabilidade de falha incluem:

Passo 7: Rever/Modificar Valores

- No Passo 3, valores foram atribuídos às características e aos atributos do software
- Atingido o consenso sobre os valores atribuídos durante a sessão de brainstorming, é possível alterar esses valores **com base em informações adicionais** ou **análises** obtidas
- Exemplos de indicadores de probabilidade de falha incluem:
  - histórico da equipe, complexidade, usabilidade, características novas ou modificadas, características desenvolvidas com novas tecnologias, histórico de defeitos, e aquelas características difíceis de testar devido a restrições do ambiente

Passo 7: Rever/Modificar Valores

- Por exemplo, sabe-se que determinados desenvolvedores produzem melhores códigos do que outros

Passo 7: Rever/Modificar Valores

- Por exemplo, sabe-se que determinados desenvolvedores produzem melhores códigos do que outros
- Conhecendo a experiência dos desenvolvedores é prudente planejar um teste mais rigoroso das características desenvolvidas pelos menos experientes

Passo 7: Rever/Modificar Valores

- Por exemplo, sabe-se que determinados desenvolvedores produzem melhores códigos do que outros
- Conhecendo a experiência dos desenvolvedores é prudente planejar um teste mais rigoroso das características desenvolvidas pelos menos experientes
- Assim sendo, a experiência dos desenvolvedores pode afetar as prioridades de teste

Passo 7: Rever/Modificar Valores

- Por exemplo, sabe-se que determinados desenvolvedores produzem melhores códigos do que outros
- Conhecendo a experiência dos desenvolvedores é prudente planejar um teste mais rigoroso das características desenvolvidas pelos menos experientes
- Assim sendo, a experiência dos desenvolvedores pode afetar as prioridades de teste
- Alterados os valores, a prioridade de risco é recalculada

Passo 7: Rever/Modificar Valores

- Por exemplo, sabe-se que determinados desenvolvedores produzem melhores códigos do que outros
- Conhecendo a experiência dos desenvolvedores é prudente planejar um teste mais rigoroso das características desenvolvidas pelos menos experientes
- Assim sendo, a experiência dos desenvolvedores pode afetar as prioridades de teste
- Alterados os valores, a prioridade de risco é recalculada
- Cuidado especial é exigido aqui quando pessoas são avaliadas para não causar indisposições pessoais

Passo 7: Rever/Modificar Valores

- Outro indicador pode ser a complexidade relativa de componentes do sistema

Passo 7: Rever/Modificar Valores

- Outro indicador pode ser a complexidade relativa de componentes do sistema
- Métricas como Complexidade Ciclomática, Pontos por Função, Pontos por Caso de Uso, dentro outras podem ser usadas

Passo 7: Rever/Modificar Valores

- Outro indicador pode ser a complexidade relativa de componentes do sistema
- Métricas como Complexidade Ciclomática, Pontos por Função, Pontos por Caso de Uso, dentro outras podem ser usadas
- A complexidade é um dos indicadores de probabilidade de falha mais utilizada

Passo 7: Rever/Modificar Valores

- Outro indicador pode ser a complexidade relativa de componentes do sistema
- Métricas como Complexidade Ciclomática, Pontos por Função, Pontos por Caso de Uso, dentro outras podem ser usadas
- A complexidade é um dos indicadores de probabilidade de falha mais utilizada
- Entretanto, no caso da Complexidade Ciclomática, é necessária a existência de código para seu cálculo preciso, ou então a complexidade poderá ser estimada

Passo 7: Rever/Modificar Valores

- Aumento da facilidade de uso do sistema, em geral, aumenta a complexidade do mesmo e, consequentemente, a probabilidade de falha

Passo 7: Rever/Modificar Valores

- Aumento da facilidade de uso do sistema, em geral, aumenta a complexidade do mesmo e, consequentemente, a probabilidade de falha
- Esse problema é amplificado pela dificuldade em reproduzir o ambiente do usuário

Passo 7: Rever/Modificar Valores

- Aumento da facilidade de uso do sistema, em geral, aumenta a complexidade do mesmo e, consequentemente, a probabilidade de falha
- Esse problema é amplificado pela dificuldade em reproduzir o ambiente do usuário
- Embora exista a máxima de que "o usuário sempre está correto", é preciso, primeiramente, determinar "quem é o usuário".

Passo 7: Rever/Modificar Valores

- Partes novas do sistema ou aquelas que receberam alterações recentes são, usualmente, mais propícias a apresentarem falhas do que as demais

Passo 7: Rever/Modificar Valores

- Partes novas do sistema ou aquelas que receberam alterações recentes são, usualmente, mais propícias a apresentarem falhas do que as demais
- Ironicamente, características com alterações em uma ou duas linhas de código apresentam taxas de introdução de defeitos maiores do que módulos com alterações mais substanciais

Passo 7: Rever/Modificar Valores

- Partes novas do sistema ou aquelas que receberam alterações recentes são, usualmente, mais propícias a apresentarem falhas do que as demais
- Ironicamente, características com alterações em uma ou duas linhas de código apresentam taxas de introdução de defeitos maiores do que módulos com alterações mais substanciais
- Isso ocorre, provavelmente, porque no caso de pequenas mudanças menos testes de regressão são executados

Passo 7: Rever/Modificar Valores

- Partes novas do sistema ou aquelas que receberam alterações recentes são, usualmente, mais propícias a apresentarem falhas do que as demais
- Ironicamente, características com alterações em uma ou duas linhas de código apresentam taxas de introdução de defeitos maiores do que módulos com alterações mais substanciais
- Isso ocorre, provavelmente, porque no caso de pequenas mudanças menos testes de regressão são executados
- É necessário um estudo específico para cada organização para determinar se isso realmente ocorre

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde
  - NTND – número total de novos defeitos

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde
  - NTND – número total de novos defeitos
  - NTDC – número total de defeitos corrigidos nesta versão

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde
  - NTND – número total de novos defeitos
  - NTDC – número total de defeitos corrigidos nesta versão
  - NTD – número total de defeitos

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde
  - NTND – número total de novos defeitos
  - NTDC – número total de defeitos corrigidos nesta versão
  - NTD – número total de defeitos
  - NTLC – número total de linhas de código alteradas

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde
  - NTND – número total de novos defeitos
  - NTDC – número total de defeitos corrigidos nesta versão
  - NTD – número total de defeitos
  - NTLC – número total de linhas de código alteradas
- Infelizmente, corrigir um defeito não implica necessariamente em um software mais confiável

Passo 7: Rever/Modificar Valores

- A proporção de características novas ou modificadas pode ser calculada como:

$$\frac{NTND}{NTDC} \quad \text{ou} \quad \frac{NTD}{NTLC}$$

- onde
  - NTND – número total de novos defeitos
  - NTDC – número total de defeitos corrigidos nesta versão
  - NTD – número total de defeitos
  - NTLC – número total de linhas de código alteradas
- Infelizmente, corrigir um defeito não implica necessariamente em um software mais confiável
- A cada defeito corrigido, novos podem ser introduzidos

Passo 7: Rever/Modificar Valores

- Além disso, algumas empresas introduzem múltiplas mudanças tecnológicas no mesmo projeto e ao mesmo tempo

Passo 7: Rever/Modificar Valores

- Além disso, algumas empresas introduzem múltiplas mudanças tecnológicas no mesmo projeto e ao mesmo tempo
- Se esse for o caso, os testadores devem ficar em alerta

Passo 7: Rever/Modificar Valores

- Além disso, algumas empresas introduzem múltiplas mudanças tecnológicas no mesmo projeto e ao mesmo tempo
- Se esse for o caso, os testadores devem ficar em alerta
- O ponto chave é que características desenvolvidas usando novas tecnologias, métodos, técnicas ou linguagens podem exigir esforço adicional de teste

Passo 7: Rever/Modificar Valores

- Outro problema está na dificuldade de replicar o ambiente de produção no laboratório de testes

Passo 7: Rever/Modificar Valores

- Outro problema está na dificuldade de replicar o ambiente de produção no laboratório de testes
- Isso aumenta o risco de certas características, atributos ou do projeto todo

Passo 7: Rever/Modificar Valores

- Outro problema está na dificuldade de replicar o ambiente de produção no laboratório de testes
- Isso aumenta o risco de certas características, atributos ou do projeto todo
- Desse modo, o realismo do ambiente pode afetar a prioridade dos teste

Passo 7: Rever/Modificar Valores

- Outro problema está na dificuldade de replicar o ambiente de produção no laboratório de testes
- Isso aumenta o risco de certas características, atributos ou do projeto todo
- Desse modo, o realismo do ambiente pode afetar a prioridade dos teste
- Assim sendo, para aquelas características difíceis de serem testadas devido à acessibilidade ao ambiente adequado deve ser atribuída uma maior probabilidade de falha e uma maior prioridade de teste, permitindo que um tempo adicional seja dado para sua avaliação

Passo 7: Rever/Modificar Valores

- Outro problema está na dificuldade de replicar o ambiente de produção no laboratório de testes
- Isso aumenta o risco de certas características, atributos ou do projeto todo
- Desse modo, o realismo do ambiente pode afetar a prioridade dos teste
- Assim sendo, para aquelas características difíceis de serem testadas devido à acessibilidade ao ambiente adequado deve ser atribuída uma maior probabilidade de falha e uma maior prioridade de teste, permitindo que um tempo adicional seja dado para sua avaliação
- Tais características são fortes candidatas para **mitigação**, abordada no Passo 10

Passo 7: Rever/Modificar Valores

- Uma análise fortemente recomendada é a de tendência e padrões de defeitos

Passo 7: Rever/Modificar Valores

- Uma análise fortemente recomendada é a de tendência e padrões de defeitos
- Se defeitos encontrados versões anteriores ou níveis iniciais de teste e/ou inspeção são documentados, eles podem ser usados para determinar áreas do software que merecem maior atenção e esforço de teste

Passo 7: Rever/Modificar Valores

- Uma análise fortemente recomendada é a de tendência e padrões de defeitos
- Se defeitos encontrados versões anteriores ou níveis iniciais de teste e/ou inspeção são documentados, eles podem ser usados para determinar áreas do software que merecem maior atenção e esforço de teste
- Testadores denominam esse tipo de análise de **Análise de Pareto**

Passo 7: Rever/Modificar Valores

- Muitos fenômenos de software seguem a distribuição de Pareto: 80% dos resultados são decorrentes de 20% do todo

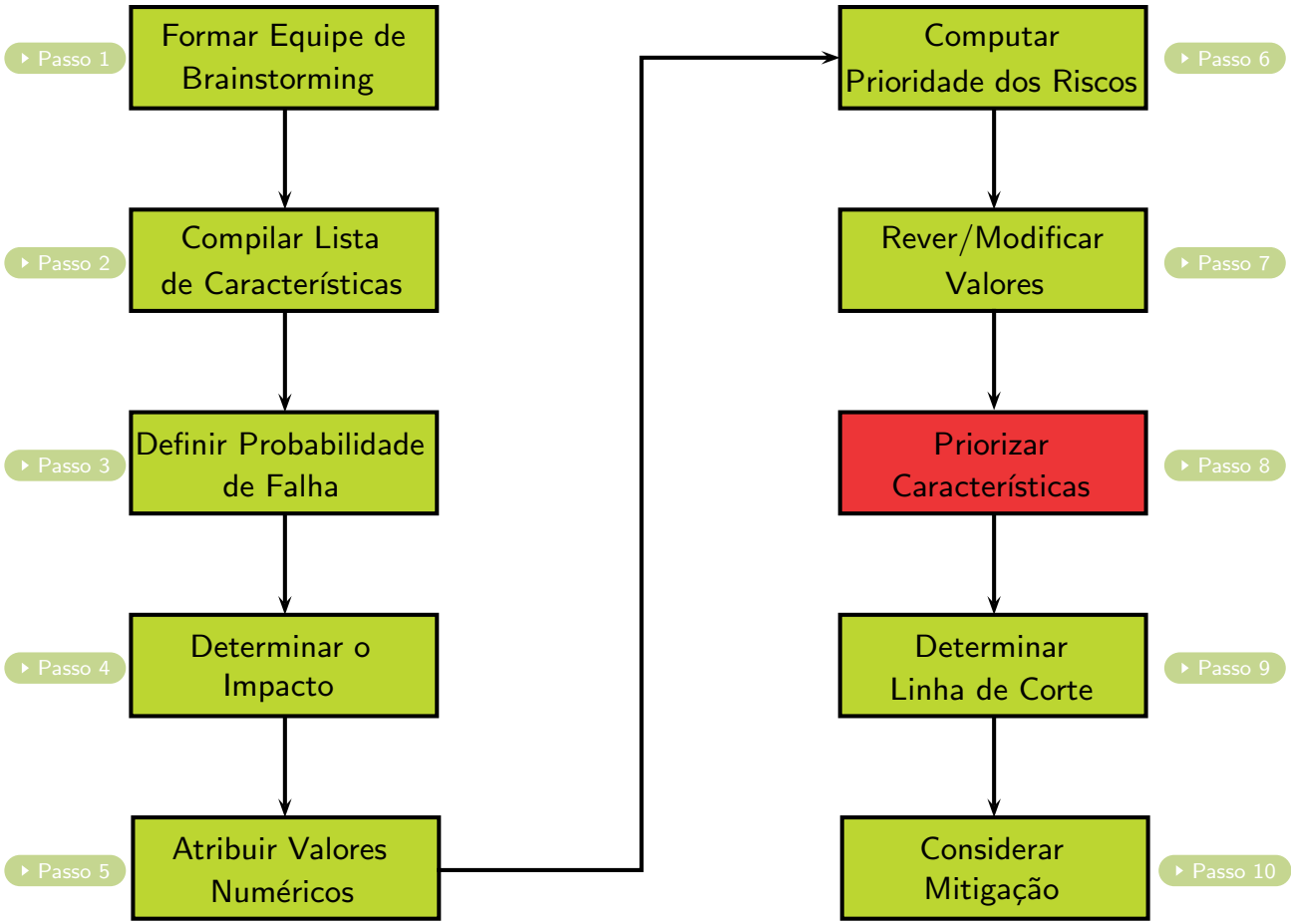
Passo 7: Rever/Modificar Valores

- Muitos fenômenos de software seguem a distribuição de Pareto: 80% dos resultados são decorrentes de 20% do todo
- Por exemplo: 80% dos defeitos (não necessariamente os mesmos) vem de 20% dos módulos

Passo 7: Rever/Modificar Valores

- Muitos fenômenos de software seguem a distribuição de Pareto: 80% dos resultados são decorrentes de 20% do todo
- Por exemplo: 80% dos defeitos (não necessariamente os mesmos) vem de 20% dos módulos
- Isso geralmente ocorre pois existem partes do sistema que são usualmente mais complexas ou escritas a partir de uma especificação mais pobre, incompleta ou inconsistente

Passo 8: Priorizar Características



Adaptado de Craig e Jaskiel (2002)

Passo 8: Priorizar Características

- Nesse passo, a lista de características e atributos deve ser reorganizada

Passo 8: Priorizar Características

- Nesse passo, a lista de características e atributos deve ser reorganizada
- A lista ordenada por prioridade fornece uma visão clara de quais riscos merecem mais atenção

Passo 8: Priorizar Características

- Nesse passo, a lista de características e atributos deve ser reorganizada
- A lista ordenada por prioridade fornece uma visão clara de quais riscos merecem mais atenção
- Como pode ser observado, uma deficiência desta técnica de priorização é que ela não leva em conta as dependências de teste

Passo 8: Priorizar Características

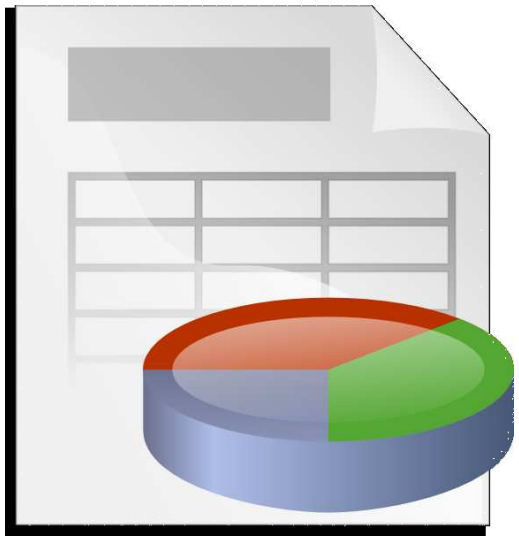
- Nesse passo, a lista de características e atributos deve ser reorganizada
- A lista ordenada por prioridade fornece uma visão clara de quais riscos merecem mais atenção
- Como pode ser observado, uma deficiência desta técnica de priorização é que ela não leva em conta as dependências de teste
- Por exemplo, no caso do ATM, embora a função de **Saldo** tenha recebido baixa prioridade, ela deve ser testada antes uma vez que para realizar um **Saque**, o **Saldo** deve ser verificado

Passo 8: Priorizar Características

- Nesse passo, a lista de características e atributos deve ser reorganizada
- A lista ordenada por prioridade fornece uma visão clara de quais riscos merecem mais atenção
- Como pode ser observado, uma deficiência desta técnica de priorização é que ela não leva em conta as dependências de teste
- Por exemplo, no caso do ATM, embora a função de **Saldo** tenha recebido baixa prioridade, ela deve ser testada antes uma vez que para realizar um **Saque**, o **Saldo** deve ser verificado
- É importante ignorar essas dependências até que o primeiro rascunho da análise de risco esteja finalizado

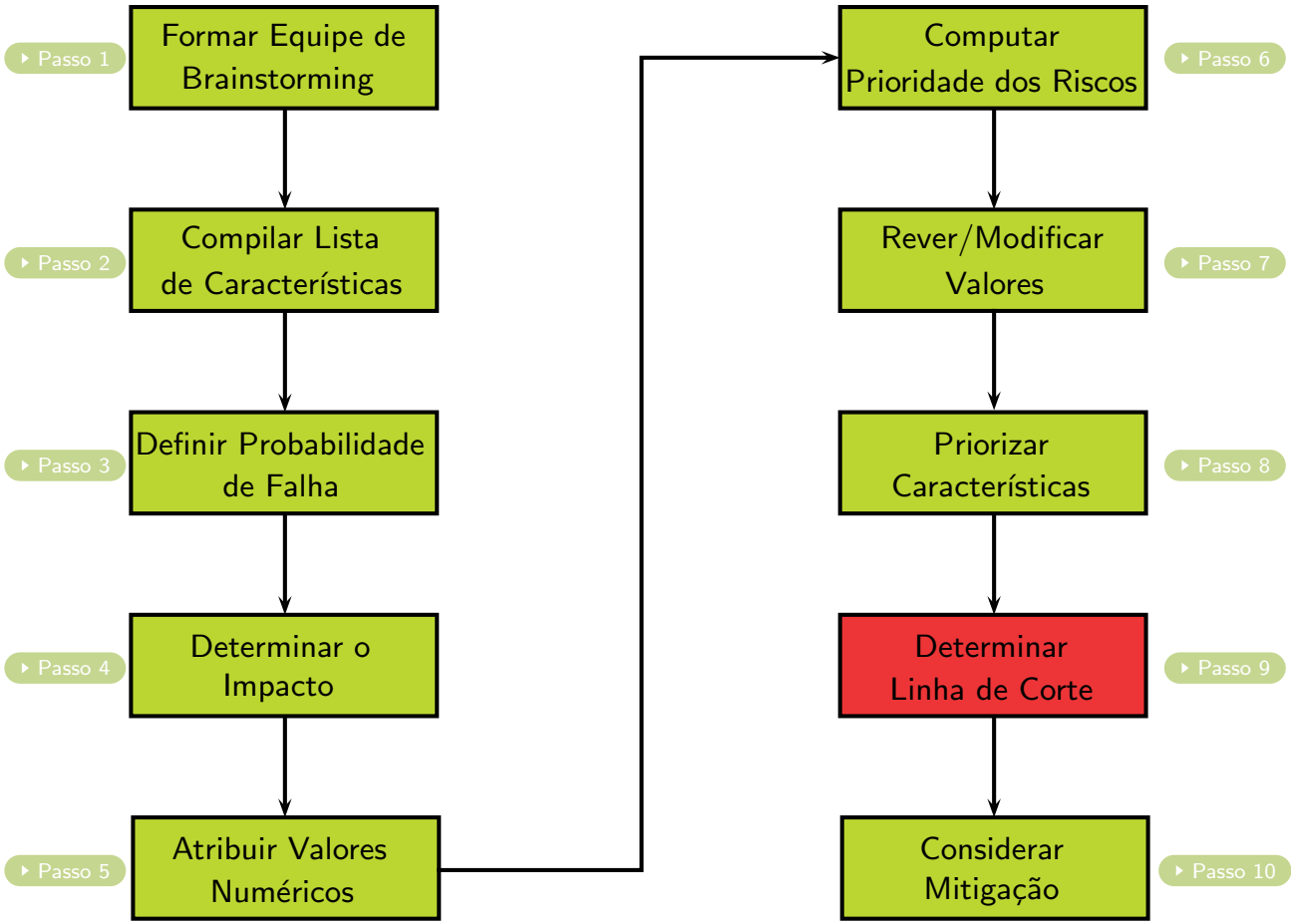
Passo 8: Priorizar Características

Exemplo ATM



▸ Síntese do Processo

Passo 9: Determinar Linha de Corte



Adaptado de Craig e Jaskiel (2002)

Passo 9: Determinar Linha de Corte

- Após as prioridades ordenadas é possível estabelecer uma linha de corte delimitando quais características ou atributos serão testados e quais não serão, ou receberão menos testes

Passo 9: Determinar Linha de Corte

- Após as prioridades ordenadas é possível estabelecer uma linha de corte delimitando quais características ou atributos serão testados e quais não serão, ou receberão menos testes
- Obviamente, à medida que o tempo passa e as estimativas são revistas, a linha de corte pode ser movida para cima ou para baixo

Passo 9: Determinar Linha de Corte

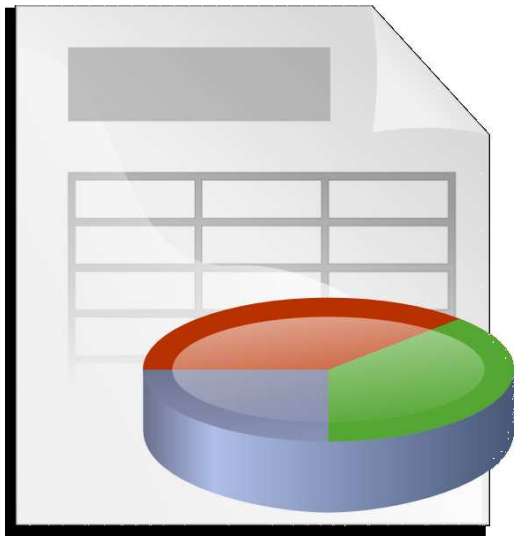
- Após as prioridades ordenadas é possível estabelecer uma linha de corte delimitando quais características ou atributos serão testados e quais não serão, ou receberão menos testes
- Obviamente, à medida que o tempo passa e as estimativas são revistas, a linha de corte pode ser movida para cima ou para baixo
- Para sistemas críticos é inconcebível características ou atributos não testados

Passo 9: Determinar Linha de Corte

- Após as prioridades ordenadas é possível estabelecer uma linha de corte delimitando quais características ou atributos serão testados e quais não serão, ou receberão menos testes
- Obviamente, à medida que o tempo passa e as estimativas são revistas, a linha de corte pode ser movida para cima ou para baixo
- Para sistemas críticos é inconcebível características ou atributos não testados
- É papel do gerente de teste decidir, com os recursos disponíveis, o que será priorizado e o que pode ser feito

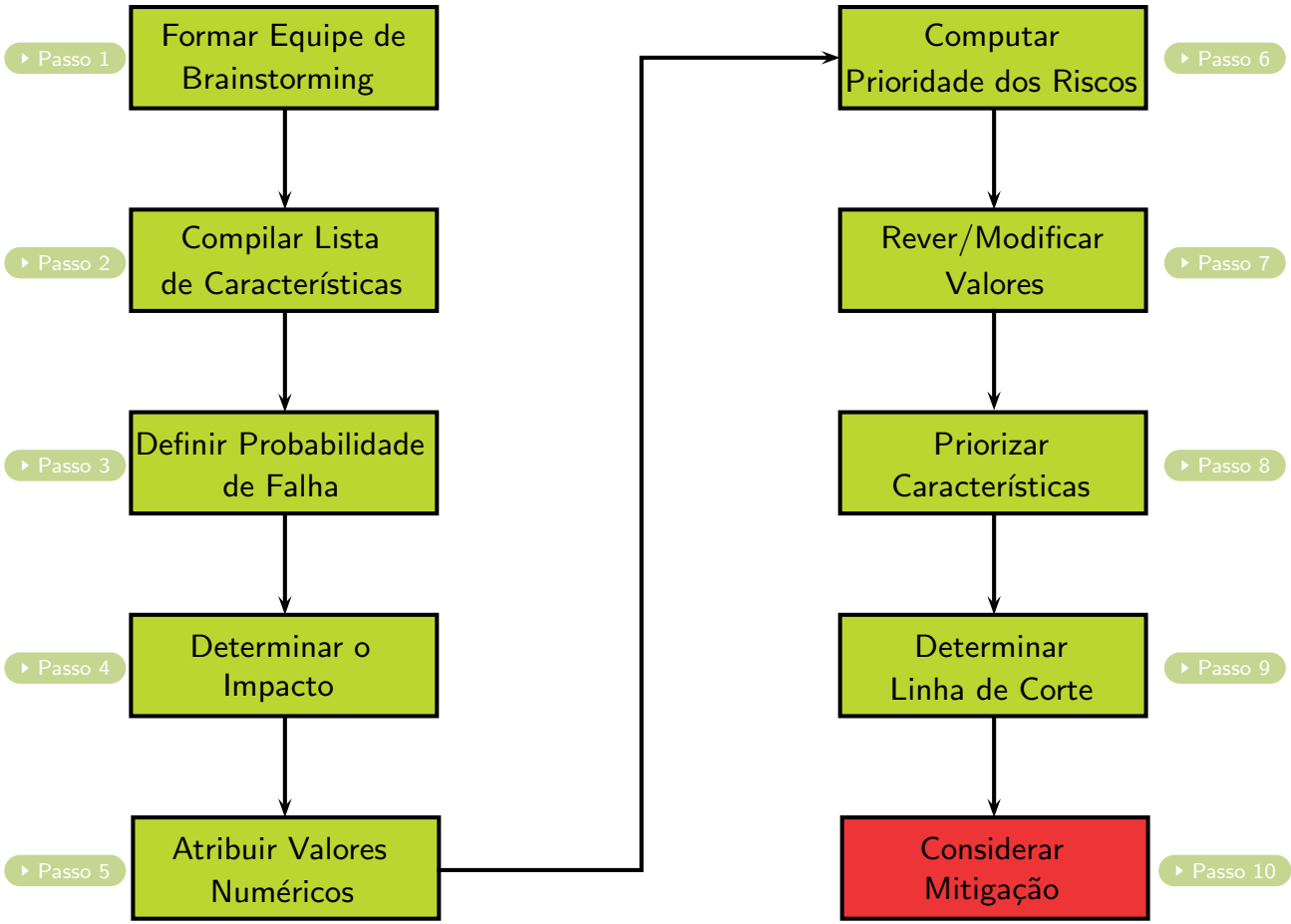
Passo 9: Determinar Linha de Corte

Exemplo ATM



▸ Síntese do Processo

Passo 10: Considerar Mitigação



Adaptado de Craig e Jaskiel (2002)

Passo 10: Considerar Mitigação

- Algumas empresas adotam uma coluna adicional na planilha de análise de risco denominada **mitigação**

Passo 10: Considerar Mitigação

- Algumas empresas adotam uma coluna adicional na planilha de análise de risco denominada **mitigação**
- Para aquelas características ou atributos com alta prioridade, o grupo de desenvolvimento ou de teste pode decidir identificar formas para minimizar ou eliminar (mitigar) o risco

Passo 10: Considerar Mitigação

- Algumas empresas adotam uma coluna adicional na planilha de análise de risco denominada **mitigação**
- Para aquelas características ou atributos com alta prioridade, o grupo de desenvolvimento ou de teste pode decidir identificar formas para minimizar ou eliminar (mitigar) o risco
- Como exemplos de mitigação podem-se citar: atividades de revisão ou inspeção, uso de prototipação, incluir mais níveis de teste

Passo 10: Considerar Mitigação

- Algumas empresas adotam uma coluna adicional na planilha de análise de risco denominada **mitigação**
- Para aquelas características ou atributos com alta prioridade, o grupo de desenvolvimento ou de teste pode decidir identificar formas para minimizar ou eliminar (mitigar) o risco
- Como exemplos de mitigação podem-se citar: atividades de revisão ou inspeção, uso de prototipação, incluir mais níveis de teste
- É importante observar que a mitigação de risco reduz a probabilidade de falha, mas não afeta o seu impacto

Passo 10: Considerar Mitigação

- Finalizado esse passo, a equipe atingiu a primeira versão da **Análise de Risco do Software**, o que a maioria dos grupos não consegue fazer

Passo 10: Considerar Mitigação

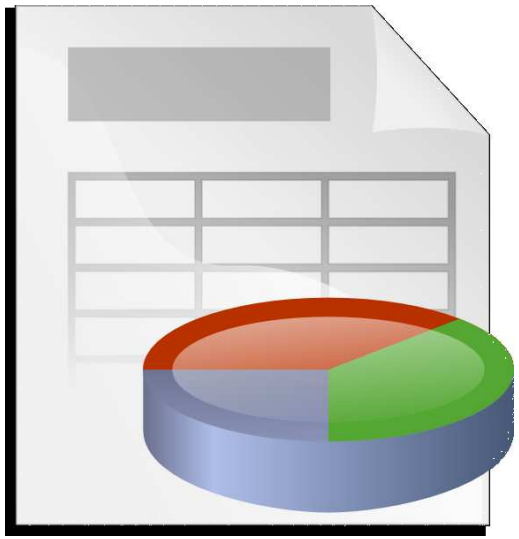
- Finalizado esse passo, a equipe atingiu a primeira versão da **Análise de Risco do Software**, o que a maioria dos grupos não consegue fazer
- Esse documento pode ser revisado ocasionalmente em função de mudanças nos requisitos, escopo, projeto, cronograma, ou outros fatores

Passo 10: Considerar Mitigação

- Finalizado esse passo, a equipe atingiu a primeira versão da **Análise de Risco do Software**, o que a maioria dos grupos não consegue fazer
- Esse documento pode ser revisado ocasionalmente em função de mudanças nos requisitos, escopo, projeto, cronograma, ou outros fatores
- Para uma nova versão do software, a análise de risco anterior pode ser utilizada como base para a nova análise

Passo 10: Considerar Mitigação

Exemplo ATM



▸ Síntese do Processo

Planejando Riscos e Contingências (1)

- Outro aspecto da gerência de risco diz respeito ao planejamento de risco

Planejando Riscos e Contingências (1)

- Outro aspecto da gerência de risco diz respeito ao planejamento de risco
- Eventos ou atividades não programadas que podem comprometer o cronograma de teste

Planejando Riscos e Contingências (1)

- Outro aspecto da gerência de risco diz respeito ao planejamento de risco
- Eventos ou atividades não programadas que podem comprometer o cronograma de teste
- Planejamento de risco é tudo aquilo que interfere no esforço de teste programado

Planejando Riscos e Contingências (1)

- Outro aspecto da gerência de risco diz respeito ao planejamento de risco
- Eventos ou atividades não programadas que podem comprometer o cronograma de teste
- Planejamento de risco é tudo aquilo que interfere no esforço de teste programado
- Alguns exemplos comuns de planejamento de risco incluem: datas de entrega, disponibilidade da equipe, orçamento, opções de ambiente, disponibilidade de ferramentas, cronograma de aquisição, necessidade de treinamento, falta de requisitos, baixa qualidade do software, dentre outras

Planejando Riscos e Contingências (1)

- Outro aspecto da gerência de risco diz respeito ao planejamento de risco
- Eventos ou atividades não programadas que podem comprometer o cronograma de teste
- Planejamento de risco é tudo aquilo que interfere no esforço de teste programado
- Alguns exemplos comuns de planejamento de risco incluem: datas de entrega, disponibilidade da equipe, orçamento, opções de ambiente, disponibilidade de ferramentas, cronograma de aquisição, necessidade de treinamento, falta de requisitos, baixa qualidade do software, dentre outras
- O propósito desta análise de risco é determinar o melhor contingenciamento caso um dos eventos do planejamento de risco vier a ocorrer

Planejando Riscos e Contingências (2)

- A maioria dos gerentes de teste considera que durante as fases de planejamento a equipe está mais motivada a sentar e tomar decisões sobre o que deve ser feito caso algum dos riscos planejados venha a ocorrer

Planejando Riscos e Contingências (2)

- A maioria dos gerentes de teste considera que durante as fases de planejamento a equipe está mais motivada a sentar e tomar decisões sobre o que deve ser feito caso algum dos riscos planejados venha a ocorrer
- Iniciado o projeto, a tomada de decisão em cima da hora pode levar a enganos fatais

Planejando Riscos e Contingências (2)

- A maioria dos gerentes de teste considera que durante as fases de planejamento a equipe está mais motivada a sentar e tomar decisões sobre o que deve ser feito caso algum dos riscos planejados venha a ocorrer
- Iniciado o projeto, a tomada de decisão em cima da hora pode levar a enganos fatais
- Por exemplo, imagine o início de um projeto com pequeno atraso e um grande vendedor de software lança uma nova versão de seu sistema operacional no qual a aplicação irá executar

Planejando Riscos e Contingências (2)

- A maioria dos gerentes de teste considera que durante as fases de planejamento a equipe está mais motivada a sentar e tomar decisões sobre o que deve ser feito caso algum dos riscos planejados venha a ocorrer
- Iniciado o projeto, a tomada de decisão em cima da hora pode levar a enganos fatais
- Por exemplo, imagine o início de um projeto com pequeno atraso e um grande vendedor de software lança uma nova versão de seu sistema operacional no qual a aplicação irá executar
- O objetivo aqui não é analisar as razões do porque isso ocorre e sim o que pode ser feito a respeito

Estudos de Caso

Estudo de Caso 1

A nova versão do produto em desenvolvimento foi prometida para uma data bem ambiciosa. A data parece bem agressiva frente a quantidade de recursos disponíveis e há necessidade de entregar um produto de alta qualidade (a última versão foi um fracasso). O inesperado acontece. Um dos principais membros da equipe recebe uma proposta do concorrente e abandona o projeto deixando uma grande lacuna na base de conhecimento sobre o projeto. Agora a data que era agressiva parece impossível. Quais são as escolhas ou contingências possíveis?

1. Alterar o cronograma – o que a equipe de vendas diz que não pode ser feito
2. Reduzir o escopo – mas nos comprometemos com nosso cliente
3. Reduzir a qualidade (reduzir teste e deixar mais defeitos no produto final) – a última versão foi um fracasso
4. Adicionar mais recursos (incluindo horas extras) – mas não há o que adicionar e toda a equipe já está sobrecarregada
5. ...

Situação complicada. Inicialmente deve-se tentar adicionar mais recursos. Se não for possível, a equipe vai acabar tomando alguns atalhos, eliminando um documento aqui, uma revisão ali, ou um conjunto de teste por completo.
Caso o projeto ainda esteja ameaçado, funcionalidade que não seja extremamente necessária será reagendada para versões posteriores ou a data de lançamento terá que ser adiada.

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros
- O objetivo é decidir antecipadamente o que fazer caso algum desses riscos planejados ocorra

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros
- O objetivo é decidir antecipadamente o que fazer caso algum desses riscos planejados ocorra
- Na opinião de Craig e Jaskiel (2002) existem apenas quatro possibilidades de contingência:

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros
- O objetivo é decidir antecipadamente o que fazer caso algum desses riscos planejados ocorra
- Na opinião de Craig e Jaskiel (2002) existem apenas quatro possibilidades de contingência:
  - reduzir o escopo

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros
- O objetivo é decidir antecipadamente o que fazer caso algum desses riscos planejados ocorra
- Na opinião de Craig e Jaskiel (2002) existem apenas quatro possibilidades de contingência:
  - reduzir o escopo
  - atrasar a implementação

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros
- O objetivo é decidir antecipadamente o que fazer caso algum desses riscos planejados ocorra
- Na opinião de Craig e Jaskiel (2002) existem apenas quatro possibilidades de contingência:
  - reduzir o escopo
  - atrasar a implementação
  - adicionar recursos

Planejando Riscos e Contingências (3)

- Identificando planejamento de risco e contingências auxilia na tomada inteligente de decisões
- Quase toda equipe de projeto é capaz de planejar riscos preocupantes: requisitos instáveis e definidos tardiamente, problemas no ambiente de teste, atraso na entrega do software, dentro outros
- O objetivo é decidir antecipadamente o que fazer caso algum desses riscos planejados ocorra
- Na opinião de Craig e Jaskiel (2002) existem apenas quatro possibilidades de contingência:
  - reduzir o escopo
  - atrasar a implementação
  - adicionar recursos
  - reduzir a qualidade do processo

Planejando Riscos e Contingências (4)

- Entretanto, diferentes organizações implementam essas contingências de diferentes formas

Planejando Riscos e Contingências (4)

- Entretanto, diferentes organizações implementam essas contingências de diferentes formas
- Por exemplo, adicionar recursos pode significar horas extras ou mais testadores

Estudos de Caso

Estudo de Caso 2

O usuário introduz uma alteração significativa nos requisitos em uma fase adiantada do ciclo de vida

Exemplo de Contingência #1
Solicitar ao grupo de usuários para contribuir com mais usuários visando a aumentar o esforço de teste, ou seja, adicionar recursos

Exemplo de Contingência #2
Decidir não implementar características de baixa prioridade, relegando-as para versões posteriores, ou seja, reduzir escopo

Exemplo de Contingência #3
Decidir não testar (ou testar menos) algumas características de baixo risco, identificadas por meio da análise de risco, ou seja, reduzir a qualidade do processo

Estudos de Caso

Estudo de Caso 3

O tamanho do projeto não para de crescer. Isso, por um lado, requer um aumento nos recursos de teste. Por outro, sabe-se que à medida que o tamanho do projeto aumenta, a taxa de produtividade da equipe de desenvolvimento e de teste decai

Exemplo de Contingência #1
Adicionar recursos (terceirizar desenvolvimento, adicionar usuários, adicionar desenvolvedores, autorizar horas extras, dentre outras)

Exemplo de Contingência #2
Reduzir o escopo do projeto. Definir uma estratégia de entrega incremental com o cliente

Exemplo de Contingência #3
Reduzir a qualidade do processo. Reduzir o esforço de teste em módulos de baixa criticalidade

Exemplo de Contingência #4
Atrasar a implementação

Planejando Riscos e Contingências

- Como pode ser observado, toda contingência exige compromisso

Planejando Riscos e Contingências

- Como pode ser observado, toda contingência exige compromisso
- Entretanto, sem planejamento de risco e contingências o testador é forçado a tomar decisões apressadas

Planejando Riscos e Contingências

- Como pode ser observado, toda contingência exige compromisso
- Entretanto, sem planejamento de risco e contingências o testador é forçado a tomar decisões apressadas
- Alguns planos de projeto apresentam uma seção denominada **Suposições**

Planejando Riscos e Contingências

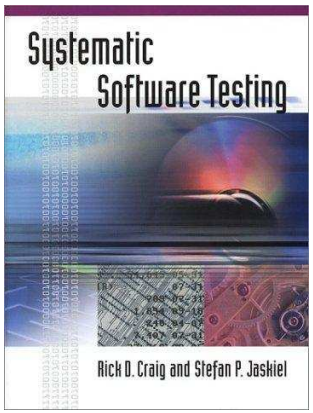
- Como pode ser observado, toda contingência exige compromisso
- Entretanto, sem planejamento de risco e contingências o testador é forçado a tomar decisões apressadas
- Alguns planos de projeto apresentam uma seção denominada **Suposições**
- Em muitos casos, tais suposições acabam se tornando planejamento de risco caso não se mantenham

Conclusão

- Análise de Risco, Planejamento de Risco e Contingência trabalham juntos
- O processo de análise de risco auxilia a identificar os riscos do software e como priorizar o esforço de teste de modo a reduzir esses riscos
- O planejamento de risco auxilia no "E se..." e em planejar as contingências

Leitura Recomendada

Mais informações podem ser obtidas no Capítulo 2 de:



Systematic Software Testing
Rich D. Craig & Stefan P. Jaskiel
Artech House, Norwood, MA, 2002.

Referências

Craig, R. D.; Jaskiel, S. P. *Systematic software testing*. Artech House Publishers, 2002.