Qualidade de Software

3º ADS

Conteúdo programático

- Introdução à qualidade de software.
- Qualidade do produto de software.
- Qualidade do processo de software.
- Técnicas de verificação e validação de software.
- Testes de software.
- Métricas da qualidade de software.
- Normas e modelos da qualidade de software.

Bibliografia

BARTIE, A. GARANTIA DA QUALIDADE DE SOFTWARE: ADQUIRINDO MATURIDADE ORGANIZACIONAL. São Paulo: Elsevier, 2010.

ROCHA, A. R. C. QUALIDADE DE SOFTWARE. Rio de Janeiro: Pearson, 2001.

HIRAMA, K. ENGENHARIA DE SOFTWARE: Qualidade e Produtividade com Tecnologia. São Paulo: Campus, 2011.

ZANIN, A. QUALIDADE DE SOFTWARE. Porto Alegre: Grupo A, 2018.





Bibliotecas

Digitais

Consultar







Reservas



Biblioteca













Estágio Contábeis



Biblioteca Digital Proview RT

Instruções de acesso Proview RT





Avaliação

```
Média 1º Semestre = (P1 + P2 + Trab1 + Trab2) / 2
Média 2º Semestre = (P1 + P2 + Trab1 + Trab2) / 2
Média_F = (Média 1_Semestre + Média 2_Semestre) / 2
```

1 – Introdução ao Gerenciamento do Produto de Software

3º ADS

Qualidade de Software

Para obter um software melhor...

1. Fornecer o <u>produto de Sw correto</u> para o cliente, significa:

- Facilidade de aprender e facilidade em utilizar.
- Se o Sw atende as necessidades do cliente, é fácil de usar e de aprender e ele não desperdiça seu tempo e deixa os cliente felizes com o Sw:
 - O produto é **VÁLIDO**.

2. <u>Fazer o correto</u>, significa:

- O Sw atende aquilo que foi especificado e o projeto satisfaz o conjunto de requisitos:
 - O produto é **VERIFICADO**.

Para obter um software melhor...

3. <u>Gerenciar corretamente</u>:

- Significa que as outras características sejam realizadas.
- A ideia é criar apenas processos e práticas adequadas para organizar o trabalho de todos os envolvidos.
- Se tudo isso for verdadeiro, dizemos que o projeto de Sw é **GERENCIADO**.

Sucesso de um projeto

 Profissionais de gerenciamento de produtos de Sw costumam usar os requisitos de cronograma, orçamento e necessidades de usuário como pontos de referência para definir o sucesso de projetos de Sw.

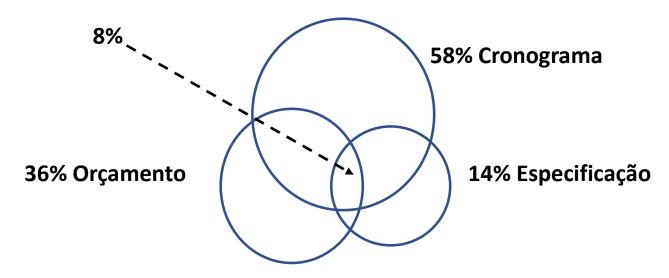
Sucesso de um projeto (cont)

Qual(is) o(s) fator(es) que você pensa ser o(s) mais importante(s) para o sucesso de um projeto de software:

- A. Atender o Cronograma.
- B. Respeitar o Orçamento.
- C. Atender aos Requisitos do usuário.

Sucesso de um projeto (cont)

• Uma pesquisa da AmbySoft* de Scott Ambler (2013) perguntou a 173 profissionais de TI como eles definiam o sucesso de um projeto de Sw:



• Apenas 8% disseram acreditar que o sucesso está na entrega dentro do prazo, dentro do orçamento e da especificação.

- Qual fator você acha que é o mais importante para o sucesso de um projeto?
 - A. Uma boa ideia.
 - B. Um time talentoso de desenvolvedores.
 - C. Um modelo de negócios.
 - D. Recursos financeiros.
 - E. Tempo.

- O sucesso de um projeto também pode ser medido:
 - Pelo número de *bugs* encontrados após a entrega do produto.
 - Pelo suporte necessário após o lançamento.
 - Pela receita gerada.
 - Pela satisfação do cliente.

• Bill Gross, um grande investidor americano, demonstrou que o fator mais importante para o sucesso de um projeto é o **TEMPO**.



... E qual é a resposta para a pergunta?

Qual(is) o(s) fator(es) que você pensa ser o(s) mais importante(s) para o sucesso de um projeto de software:

- A. Atender o Cronograma.
- B. Respeitar o Orçamento.
- C. Atender aos Requisitos do usuário.



- E quem gerencia muito bem o tempo durante o desenvolvimento de um projeto são os **Métodos Ágeis**.
- Eles colocam o foco no ritmo do desenvolvimento garantindo estabilidade e gerenciamento eficiente.
- Isso impede que os programadores fiquem sobrecarregados com o projeto.



Manifesto Ágil



Introdução

 O Manifesto Ágil foi criado por 17 gurus da engenharia de software que se reuniram num Ski Resort para encontrar maneiras de gerenciar melhor o processo de software.



Introdução

- Eles estabeleceram:
 - 4 sentenças de valores que são mais importantes no desenvolvimento de software e,
 - Para expandir os valores fundamentais eles criaram **12 princípios** de apoio.

Valores Fundamentais

- 1. Os indivíduos e suas interações acima de procedimentos e ferramentas.
- 2. O funcionamento do software acima de documentação abrangente.
- 3. A colaboração com o cliente acima da negociação e contrato.
- 4. A capacidade de resposta a mudanças acima de um plano préestabelecido.

1. Os indivíduos e suas interações acima de procedimentos e ferramentas.

- Este primeiro valor enfatiza o trabalho em equipe e a comunicação.
- Equipes de pessoas constroem sistemas de software, não ferramentas. E para isso eles precisam trabalhar juntos efetivamente através de interações produtivas.

Fonte: https://medium.com/cwi-software/https-medium-com-ftonato-os-quatro-valores-fundamentais-do-manifesto-agile13e555d3c03

2. O funcionamento do software acima de documentação abrangente.

 Enquanto estivermos trabalhando com softwares mais abrangentes, precisaremos de uma documentação robusta. Em tais cenários, precisaremos ler uma centena de páginas de especificações do produto para melhorar o nosso entendimento do software.

- 3. A colaboração com o cliente acima da negociação e contrato.
- Apenas os seus clientes podem dizer o que querem e o seu trabalho é ouvir.
 Equipes de desenvolvimento bem sucedidas trabalham em estreita colaboração com seus clientes e se comunicam com eles com frequência.

- 4. A capacidade de resposta a mudanças acima de um plano préestabelecido.
- Seu cliente pode mudar as ideias sobre o que está sendo construído:
 - pode ser porque as prioridades da empresa mudaram ou
 - novas mudanças regulatórias entram em vigor.
- Algum código poderá ser jogado fora e muito tempo pode ser perdido, mas se você estiver trabalhando com iterações curtas, certamente o tempo perdido é minimizado por conta de eventuais mudanças.

Os doze princípios do Manifesto Ágil

12 Princípios do Manifesto Ágil

• Além dos <u>quatro valores</u> instituídos, firmou-se ainda um conjunto de abordagens de desenvolvimento de softwares que foram apresentados por meio de 12 princípios. A saber:

• PRINCÍPIO 1: valor

• "A maior prioridade está em satisfazer o cliente por meio da entrega adiantada e contínua de software de valor."

• PRINCÍPIO 2: flexibilidade

• Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas, aceitando alterações de requisitos mesmo no fim do desenvolvimento.

• PRINCÍPIO 3: frequência

• "Entregar o software em funcionamento com frequência, seja na escala de semanas ou meses, dando preferência a períodos mais curtos."

• Princípio 4: união

• "Tanto pessoas relacionadas a negócios como desenvolvedores devem trabalhar em conjunto, diariamente, durante todo o curso do projeto".

- Princípio 5: motivação
 - "Para construir projetos ao redor de indivíduos motivados, é preciso dar a eles o ambiente e o suporte necessários, confiando que farão seu trabalho".
- Princípio 6: comunicação
 - "O método mais eficiente de transmitir informações tanto externas como internas para um time de desenvolvimento é por meio de uma conversa cara a cara."

- Princípio 7: funcionalidade
 - "Um software funcional é a medida primária de progresso".
- Princípio 8: sustentabilidade
 - "Processos ágeis promovem um ambiente sustentável, com patrocinadores, desenvolvedores e usuários sendo capazes de manter passos constantes."

- Princípio 9: revisão
 - "A contínua atenção à excelência técnica e ao bom design aumenta a agilidade."
- Princípio 10: simplicidade
 - "Simplicidade é a arte de maximizar a quantidade de trabalho que não precisou ser feito".

- Princípio 11: organização
 - "As melhores arquiteturas, os requisitos e os designs emergem de times auto organizáveis."
- Princípio 12: autoavaliação
 - "Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo."

Processo de Software

 Como os valores e os princípios dos Métodos Ágeis podem ser aplicados de um modo prático num projeto de desenvolvimento de Sw?

- Um processo é uma boa base para aplicar esses princípios.
- Um **processo** organiza o trabalho das pessoas em fases ou etapas distintas para desenvolver um produto de software.
- Algumas fases que vêm à mente quando você pensa em desenvolvimento de software:
 - Planejamento do projeto
 - Escrita do código
 - Teste do software
 - Manutenção do produto.
- Estes são apenas alguns exemplos. Existem muitas maneiras de estruturar um processo de desenvolvimento.



Se você fosse criar um **processo** para fazer uma pizza. Quais fases você escolheria para este **processo**?

A. Massa, Molho, Queijo, Calabresa.

B. Planejar, Preparar, Montar e Assar.

C. Discar, Encomendar, Comer, Lavar louça.

D. Fazer, Assar, Comer, Comer as Sobras.

Se você fosse criar um **processo** para fazer uma pizza. Quais fases você escolheria para este **processo**?

- A. Massa, Molho, Queijo, Calabresa.
- B. Planejar, Preparar, Montar e Assar.
- C. Discar, Encomendar, Comer, Lavar louça.
- D. Fazer, Assar, Comer, Comer as Sobras.

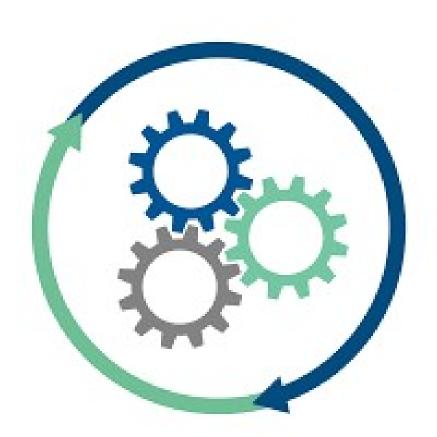
B é a resposta correta porque possui fases de alto nível para planejar, preparar, montar e cozinhar.

Embora existam muitos **processos** diferentes para organizar o trabalho em um produto durante sua vida útil, eles geralmente compartilham a noção de alto nível de uma fase.

47

- Exemplos de fases são:
 - Especificação
 - Projeto e implementação
 - Verificação e Validação.
- A Especificação é onde a ideia do produto é concebida:
 - Você saberá que a fase de especificação foi alcançada quando puder definir o que o software fará.
- **Projeto e Implementação** é descobrir a maneira ideal de estruturar o software que permite que o design e a codificação sejam efetivados.
- Finalmente, **Verificação e Validação** é onde você testa o software em busca de erros e garante que o sistema está entregando o que o cliente precisa.

- Os processos são necessários para organizar seu desenvolvimento e certificar-se de que você está concluindo as coisas em uma ordem lógica.
- Eles também garantem que as etapas não sejam perdidas ou negligenciadas.
- O desenvolvimento de software é bastante assustador quando você não sabe por onde começar.
- Os processos também fornecem clareza sobre onde você deve iniciar seu projeto.



- Um processo de software bem definido estabelece regras e responsabilidades para todos da equipe de desenvolvimento.
- A adoção de um processo de software pode parecer um passo natural para o desenvolvimento de seu produto. Pode até parecer intuitivo que um processo surja naturalmente. Mas seguir um modelo de processo definido permite que todos saibam o que esperar.

Requisitos de Software

- Requisitos são um conjunto de descrições específicas das necessidades do seu cliente.
- O cliente é a pessoa ou grupo de pessoas que deseja o produto de software.
- Requisitos combinados com processos são a espinha dorsal de qualquer projeto de software bem-sucedido.
- Então, o que os torna tão importantes?
- Veja um exemplo de uma lista de requisitos para um aplicativo de rede social. O usuário poderia:
 - armazenar perfis de usuário em seus dados,
 - enviar mensagens para outros usuários,
 - ver perfis de outros usuários,
 - criar atualizações de status
 - e ver as atualizações de status de outros usuários.

- Requisitos geram os recursos de um produto de software.
- Então, o que acontece se você receber uma solicitação aparentemente simples:
 - Como os usuários podem falar uns com os outros através do aplicativo?
- Como isso se traduz em um requisito?
- Os usuários **podem**:
 - enviar notas de voz um para o outro?
 - mensagens de texto?
 - ondas cerebrais?

- E quanto a essa palavra, **pode**? Os usuários só podem conversar entre si em determinados momentos do dia? Esse tipo de ambiguidade pode levar a uma enorme confusão para os desenvolvedores, especialmente se a maioria dos requisitos dos usuários não for clara.
- Você precisará obter mais detalhes para realmente entender as coisas.
- Evitar confusão é extremamente importante no desenvolvimento de software. Isso começa sabendo como **elicitar** e expressar adequadamente os requisitos.
- Ao gastar o tempo necessário para refinar os requisitos, você também pode detectar possíveis erros em seu produto antes mesmo de ele ser construído.
- Ao esclarecer as ideias, o desenvolvimento torna-se focado e eficiente.

Planejamento de Software



"Planejar é trazer o futuro para o presente, para que você possa fazer algo a respeito agora". (Alan Lakein)

- Um plano destaca o que pode ser feito agora e nos conduz para um objetivo futuro.
- O planejamento envolve o uso de processos e requisitos para começar a organizar tarefas e agendamentos. A criação de tarefas e cronogramas requer identificar quem deve fazer o trabalho e estimar quanto tempo esse trabalho levará.
- Requisitos mal escritos tornam as estimativas de tempo imprecisas e difíceis de serem alcançadas. Quanto mais específicos forem os requisitos, mais claro será avaliar quanto tempo o trabalho levará para ser concluído.

- Imagine que você subestime o trabalho e prometa ao cliente que o projeto será concluído em um dia específico.
- O resultado mais provável é que você não entregará como prometido.
- Estimativas e cronogramas são parte importante de boas práticas de planejamento. Ter boas estimativas reduz o risco do projeto. Reduzir o risco e planejar com antecedência faz parte do gerenciamento de riscos.



O que acontece quando sua programação é baseada em estimativas de tempo ruins?

Todo o cronograma do seu projeto será inatingível.



Imagine que você superestima o trabalho que apresenta ao cliente com estimativas de tempo inchadas e cronogramas imprecisos.

Seu projeto inflado diz ao cliente que você não entende o projeto. Você, certamente, perderá o cliente.

• Em branco

O gerenciamento de riscos é o maior benefício do planejamento. Lembre-se

Planejar é trazer o futuro para o presente, para que você possa fazer algo a respeito agora". (Alan Lakein)

Se você puder prever riscos futuros e fazer algo sobre eles no presente para mitigá-los, então seu projeto terá uma chance maior de sucesso.

Riscos que podem ser gerenciados na fase de Planejamento

1. Desenvolvimento de responsabilidades de equipe:

- Você pode criar planos para o que é necessário para ser feito e quem irá completá-lo.
- Isso reduzirá o mal-entendido sobre quem é responsável por o quê.
- Os esforços coordenados de trabalho contribuem para um desenvolvimento eficiente e ajudam a estabelecer um ritmo constante de desenvolvimento.
- Este é um princípio do Manifesto Ágil.

Riscos que podem ser gerenciados na fase de Planejamento

2. Bugs de lançamento de software:

• O planejamento eficaz pode gerar tempo de teste suficiente. Isso reduzirá os erros no produto liberado e dará tempo para o reparo se um *bug* significativo for encontrado.

3. Prazos e/ou orçamentos ausentes:

• O planejamento pode manter o desenvolvimento nos trilhos, pois reduz recursos desperdiçados. Isso reduz o risco de o projeto atrasar e/ou ultrapassar o orçamento.

• Em branco

Em branco

- O planejamento também pode ajudá-lo a prever problemas que possam ocorrer.
- Você será capaz de criar planos de risco para esses problemas. Exemplos de coisas que poderiam dar errado:
 - E se um desenvolvedor ficar doente?
 - E se alguém desistir?
 - E se o contato do seu cliente mudar?
 - E se a sua tecnologia falhar?
- Como você vai lidar com esses problemas?
- O planejamento permite que você pense em soluções para esses problemas, para que você esteja preparado caso eles aconteçam. O planejamento não é apenas para o começo de um projeto. No desenvolvimento de software, o planejamento é algo que ocorre constantemente durante todo o projeto.

- O software está sempre evoluindo, seus planos precisam ser flexíveis.
 Quando você tem um processo para revisar seu plano, seu projeto se torna mais ágil. Significa que você pode se adaptar facilmente à mudança.
- Qualquer solução preparada em um estado calmo e focado será melhor do que o raciocínio rápido no estresse do momento.

FIM

Gestão de Riscos em Projetos de Software

Plano de Projeto de Software

I. Introdução

- 1. Escopo e propósito do documento
- 2. Objetivos do projeto

II. Organização de projeto

1. Organização da equipe e papéis envolvidos

III. Riscos do Projeto

- 1. Descrição dos riscos
- 2. Estratégias de redução dos riscos

IV. Recursos de software e hardware

- 1. Descrição dos recursos
- 2. Custos relacionados

V. Divisão do trabalho (atividades do projeto, *milestones* e resultados de cada atividade)

VI. Cronograma

VII. Mecanismos de Monitoração VIII. Relatórios



O que é risco?

Definição de Risco

- "Evento ou condição incerta que, se ocorrer, terá um efeito positivo ou negativo sobre pelo menos um objetivo do projeto" (PMBOK, 2004)
- "Risco é a possibilidade de sofrer perdas" (SEI - Software Engineering Institute)



Exemplos de riscos em projetos de software



- Requisitos pouco claros (entrada/saída, fluxo, stakeholders);
- Tecnologias não conhecidas pela equipe;
- Ideias e conceitos novos;
- Novas **pessoas** na equipe;
- Mudanças de situações e prioridades;
- Planejamentos irreais.

Tipos de risco





Riscos de projeto



Riscos técnicos



Riscos de negócio

Riscos de Projeto

- Os riscos de projeto ameaçam o plano do projeto, podendo atrasar o cronograma e aumentar custos.
- Identificam problemas de:
 - Custo, tempo, pessoal (composição do pessoal e organização), recursos, clientes, requisitos...
 - A complexidade, tamanho e estrutura do projeto também são definidos como fatores de risco.





Riscos Técnicos

- Os riscos técnicos ocorrem porque um problema é mais difícil de ser resolvido do que se imaginava.
- Ameaçam o pontualidade e a qualidade do software, tornando a implementação impossível.
- Problemas no desenvolvimento do software (implementação, interface, manutenção), novas tecnologias, tecnologia não adequada a solução ...

Riscos de Negócio

- Os riscos do negócio ameaçam a viabilidade do software a ser criado
- Riscos de maior destaque:
 - construir um excelente produto que ninguém realmente quer;
 - perder o apoio da alta administração devido à mudança de foco ou mudança de pessoas (risco administrativo);
 - perder o compromisso orçamentário.



Tipos de Riscos (outras classificações)

- Conhecidos
- Previsíveis
- Imprevisíveis

- de tecnologia
- de pessoal
- da organização
- de ferramentas
- de requisitos
- de estimativa

- O projeto de software está em risco?
- Exemplo:
 - "Como consequência do uso de um novo hardware (uma exigência definida), erros inesperados de integração do sistema podem ocorrer (um risco incerto), o que levaria a estouros dos custos do projeto (efeito sobre o orçamento)".

- Técnicas para identificação de riscos:
 - Uso de checklists;
 - Reuniões e *brainstormings* com gerente e equipes experientes no projeto;
 - Análise de cenários e lições aprendidas em projetos anteriores.

• Exemplo:

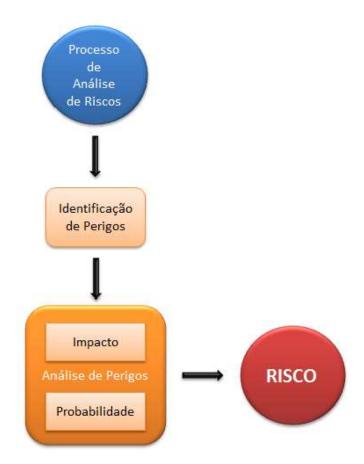
- Checklist para identificação dos riscos de Composição do Pessoal:
 - Há pessoas suficientes à disposição?
 - As pessoas têm a combinação certa de habilidades?
 - O pessoal está comprometido com toda a duração do projeto?
 - Algum membro estará trabalhando parcialmente nesse projeto?
 - O pessoal tem as expectativas certas sobre o trabalho que tem à mão?
 - A equipe recebeu o treinamento necessário?
 - A rotatividade entre os membros do pessoal será baixa o bastante para permitir continuidade?

• Exemplo:

- Checklist para identificação dos riscos de Características do Cliente:
 - Você já realizou outros projetos com o cliente?
 - O cliente tem ideias sólidas dos requisitos?
 - O cliente concorda em "gastar" tempo com você?
 - O cliente está disposto em participar das revisões?
 - O cliente tem expectativas realísticas?

Análise dos Riscos

- Identificar quais riscos são relevantes
- Propriedades dos riscos:
 - Probabilidade
 - Impacto
 - Proximidade.



Análise dos Riscos

- Isto é um risco ou não?
- Qual a probabilidade de ocorrência?
- O quanto sério é este risco?
- Quais são as consequências?







Risk Management, Mitigation and Monitoring - RMMM

Risco	Prob.	Impacto	RMMM
1. Tamanho estimado menor que o real	60%	2	
2. Numero de usuários maior que o planejado	30%	3	
3. Menos reuso que o planejado	70%	2	
4. Usuários finais resistentes ao sistema	40%	3	
5. Pressão para reduzir prazos de entrega	50%	2	
6. Perda de financiamento	40%	1	
7. Usuários irão modificar os requisitos	80%	2	
8. Tecnologia não atingirá as expectativas	30%	1	
9. Falta de treinamento no uso de ferramentas	80%	3	
10. Pessoal sem experiência	30%	2	
11. Rotatividade do pessoal grande	60%	2	

Tabela de Riscos

Probabilidade: avaliada pela equipe em rodadas sucessivas até obter consenso

Impacto: 1 - catastrófico; 2 - crítico; 3 - marginal; 4 - negligenciável

Tabela de Riscos (ordenada)

Risco	Prob.	Impacto	RMMM
7. Usuários irão modificar os requisitos	80%	3	
9. Falta de treinamento no uso de ferramentas	80%	3	
3. Menos reuso que o planejado	70%	2	
11. Rotatividade do pessoal grande	60%	2	
1. Tamanho estimado menor que o real	60%	2	
5. Pressão para reduzir prazos de entrega	50%	2	
4. Usuários finais resistentes ao sistema	40%	3	
6. Perda de financiamento	40%	1	
2. Numero de usuários maior que o planejado	30%	3	
10. Pessoal sem experiência	30%	2	
8. Tecnologia não atingirá as expectativas	30%	1	

Probabilidade: avaliada pela equipe em rodadas sucessivas até obter consenso

Impacto: 1 - catastrófico; 2 - crítico; 3 - marginal; 4 - negligenciável

Tabela de Riscos (ordenada)

Risco	Prob.	Impacto	RMMM
7. Usuários irão modificar os requisitos	80%	3	2
9. Falta de treinamento no uso de ferramentas	80%	3	
3. Menos reuso que o planejado	70%	2	*
11. Rotatividade do pessoal grande	60%	2	2
1. Tamanho estimado menor que o real	60%	2	
5. Pressão para reduzir prazos de entrega	50%	2	Ť
4. Usuários finais resistentes ao sistema	40%	3	
6. Perda de financiamento	40%	1	
2. Numero de usuários maior que o planejado	30%	3	
10. Pessoal sem experiência	30%	2	
8. Tecnologia não atingirá as expectativas	30%	1	

___ 96

Impacto do risco

- Avaliar as consequências do risco, depende:
 - Natureza do risco
 - Escopo do risco
 - Época
- Exposição geral ao risco (risk exposure RE) NASA:
 - RE = P * C
 - P = probabilidade e
 - C = custo do risco para o projeto
- Permite ajustar a estimativa do custo e para previsão de recursos de pessoal no projeto.



Impacto do risco – Avaliação – Exemplo

Risco do Projeto	Descrição
Identificação do risco	Somente 70% dos componentes poderão ser reutilizados. As funcionalidades restantes serão desenvolvidas de forma personalizada.
Probabilidade	80%
Impacto	Foram planejados 60 componentes reutilizáveis. Se 70% for usado (42 componentes), 18 terão que ser desenvolvidos desde o início. Considerando que cada componente possui 100 LOC (<i>lines of code</i>) e dados iniciais indicam um custo de R\$40/LOC, o custo a mais será: 18 * 100 * 40 = R\$ 72.000
Exposição ao risco	RE = 0,80 * 72.000 = R\$ 57.600 (estimativa de custo a mais com pessoal neste projeto)

RMMM – Mitigação, Monitoramento e Controle de Riscos

- Como evitar o risco?
- Como monitorar o risco?
- Como gerenciar o risco e planejar a contingência?



RMMM – Mitigação, Monitoramento e Controle de Riscos

- Exemplo
 - Como mitigar risco de rotatividade de pessoal?



Plano para mitigar riscos

- Reunião com a equipe para identificar causas de rotatividade elevada:
 - Condições de trabalho ruins, salários baixos, mercado de trabalho competitivo.
- Após o início do projeto, assumir que a rotatividade acontecerá e desenvolver técnicas para garantir a continuidade do projeto.
- Organizar a equipe de tal maneira que informações sobre atividades de desenvolvimento são amplamente conhecidas.
- Definir padrões para o desenvolvimento.
- Executar revisões em pares de todo o trabalho.

Monitoramento dos riscos

 Após o início do projeto, o gerente deve monitorar os fatores que podem indicar se a probabilidade do risco está aumentando ou diminuindo.

• Exemplos:

- Atitudes da equipe em função do aumento das pressões do projeto
- Relações interpessoais
- Problemas com remuneração
- Aumento da oferta de trabalho.

RMMM – Mitigação, Monitoramento e Controle de Riscos

- O controle dos riscos gera custo ao projeto
- Grandes projetos:
 - 30 a 40 riscos
 - Se cada risco demandar de 3 a 7 passos de gestão de risco, isso gera um projeto por si só!
- Solução: adaptação a riscos do Princípio de Pareto
 - Regra 80 20:
 - 80% das falhas de projeto são devido a
 - 20% dos riscos identificados.



Gestão de riscos

- Conjunto de atividades que tem por objetivo minimizar os efeitos dos riscos e maximizar oportunidades;
- Lida com imprevistos, fazendo com que possíveis cenários futuros fiquem dentro de uma faixa aceitável;
- É um processo contínuo ...



Gestão de riscos

- Atividades da Gestão de Projetos:
 - 1. Identificação dos riscos
 - 2. Análise dos riscos (priorização)
 - 3. Planejamento (planos para reduzir a probabilidade do risco)
 - 4. Rastreamento
 - 5. Controle (execução de planos caso necessário)

Formulário de informação do risco

ID do risco PO-024-32	Probabilidade: 80% - Impacto: alto
Descrição	Somente 70% dos componentes de software programados para reutilização serão de fato integrados na aplicação. A funcionalidade restante deverá ser desenvolvida de maneira Personalizada.
Mitigação/ Monitoramento	 verificar os padrões de projeto e como utilizar adequadamente os componentes já desenvolvidos; certificar para que ocorra a padronização de interfaces de comunicação; identificar e descrever claramente as interfaces de comunicação dos componentes a serem desenvolvidos.
Gerenciamento/ Contingência	Foi calculada a exposição ao risco e resultou em \$57.600. Reservar esse valor no custo de contingência do projeto. Desenvolver um cronograma para o desenvolvimento dos 18 componentes personalizados.
Estado atual	Iniciadas as etapas de mitigação (//)

10 maiores riscos (Boehm)

- 1. Imprevistos com pessoal.
- 2. Cronogramas e orçamentos não realísticos.
- 3. Desenvolvimento das funções erradas.
- 4. Desenvolvimento da interface com o usuário errada.
- 5. Requisitos sofisticados, sem necessidade.





10 maiores riscos (Boehm)

- 6. Fluxo contínuo de mudanças nos requisitos.
- 7. Imprevistos em serviços terceirizados.
- 8. Imprevistos em componentes terceirizados.
- 9. Imprevistos de desempenho em tempo real.
- 10. Capacidade de computação excedida.



Exercício – A partir do seu Projeto de TCC

- Identifique os riscos de: projeto, técnico e de negócio (slides 84 a 86)
- Para cada risco defina o impacto e a probabilidade de ocorrência.
 - Probabilidade: avalie, segundo sua experiência, e atribua um valor em % para cada risco identificado: de 0 a 100%
 - Impacto:
 - 1 catastrófico
 - 2 crítico
 - 3 marginal
 - 4 negligenciável
- Elabore uma Tabela RMMM (slides 94 e 95) e apresente os riscos que identificou. Aponte pelo menos 06 (seis) riscos relacionados ao seu projeto.
- Apresente o formulário* para a Identificação do Risco (slide 106).
- *Observação: não precisa constar as linhas "Gerenciamento/Contingência" e "Estado Atual".

Exercício — A partir do seu Projeto de TCC (cont)

- O trabalho deverá apresentar:
 - a) Seu nome
 - b) Título do Projeto de TCC (pode ser título provisório)
 - c) Descrição sucinta do projeto
 - Breve descrição dos objetivos do seu projeto.
 - d) Tabela RMMM com as Probabilidades e Impactos (do risco 1 ao risco 6 pelo menos)
 - e) Tabela RMMM Ordenada
 - f) Para cada risco identificado:
 - Formulário de informação do risco.
 - g) Gere um arquivo com extensão *pdf* da seguinte forma:
 - Altamiro Figueiredo Neto.pdf
 - h) Faça um *checklist* para certificar-se de que está tudo como o solicitado e faça o *upload* no Moodle.

FIM