
Gerenciamento Ágil de Projeto

por

Adriana Silveira de Souza



ESTRATÉGIA
Tecnologia da Informação

Adriana Silveira de Souza

- Professora da Pontifícia Universidade Católica de Goiás (PUCGO) e da Universidade Federal de Goiás (UFG)
 - Doutoranda em Psicologia na Universidade de Brasília
 - Mestra em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS)
 - Psicóloga pela PUCGO e Psicanalista
 - Master Coach, Coach de Executivo, Carreiras e de Vida
 - Membro do Comitê Gestor do Modelo MPS.BR
 - Consultora e Auditora Líder (*Lead Auditor*) da Norma de Qualidade NBR ISO/IEC 9001:2008
 - Avaliadora Líder do modelo CERTICS
-

Adriana Silveira de Souza

- Implementadora, Avaliadora Líder e Instrutora oficial do modelo MPS.BR
 - Coordenadora da Instituição Implementadora do MPS.BR Estratégia TI
 - Coordenadora da Implantação de processos da Fábrica de Software da UFG
 - Coordenou a avaliação de Produtos de Software na 1ª Chamada Nacional SOFTEX - Núcleo UFRGS
 - Participou de avaliações de Produtos do Prêmio ASSESPRO
 - Implementadora de melhoria de processos em empresas do estado de Goiás e do Distrito Federal
 - Avaliadora de processos de diversas organizações produtoras de software
 - Pesquisadora nas áreas de processos de software e comportamento organizacional
-

Bibliografia

- ✓ P. Abrahamsson, O. Salo, J. Ronkainen. “Agile software development methods, Review and Analysis”
- ✓ <http://www.agilealliance.org>
- ✓ <http://www.agilemodeling.com>
- ✓ <http://forum.agilesoftwaredevelopment.org/>
- ✓ C. Larman and V. R. Basili, “Iterative and Incremental Development: A Brief History”
- ✓ L Williams, A. Cockburn. “Agile Software Development: It’s about Feedback and Change”
- ✓ J. Highsmith. “Agile Software Development Ecosystems”
- ✓ K. Schwaber, M. Beedle. “Agile Software Development with SCRUM”
- ✓ <http://www.extremeprogramming.org>

Bibliografia

- ✓ Rising. L; Janoff, Norman. (2000). The scrum software development process for small teams. In: IEEE Software, pg 26-32 Silva, Lilian S. F.; Oliveira, Sandro B.. (2016) A Process Framework with Agile Practices for Implementation of Project Portfolio. In: 10th International Conference on the Quality of Information and Communications Technology. Pg. 146-149
- ✓ Boehm, Barry; Turner, Richard. (2004). Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods. In: Proceedings of the 26th International conference on Software Engineering (ICSE04).
- ✓ Boehm, Barry; Turner, Richard. (2003). Using Risk to Balance Agile and Plan-Driven Methods. In: IEEE Computer Society. Pg. 57-66.
- ✓ Luna. Alexandre, Kruchten, Philippe; Moura, Hermano. (2013). Game: Governance for Agile Management of Enterprises: A Management Model for Agile Governance. In: IEEE 8th International conference on Global Software Engineering Workshops. Pg 88-90
- ✓ Morien, Roy. (2005) Agile Management and the Toyota Way for software Project Management. In: IEEE 3th International Conference on Industrial Informatics. P.516-522.

Bibliografia

- ✓ Gold, Benjamin; Vassel, Clive; (2015). Using Risk Management to Balance Agile Methods. A study of the Scrum Process. In: 2nd International conference on knowledge-Based Engineering and Innovation. Pg 49-54.
- ✓ Lopez, R. E. (2015). Agile Software Development Applied to the Management of Business Projects. In: Proceedings of the 2015 Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)
- ✓ HU, Zhi-gen. et alli. (2009). Research on Agile Project Management with Scrum Method. In: 2th International conference on Services Science, Management and Engineering. P.26-29
- ✓ Wells, Hany; Dalcher, Darren; Smyth, Hedley. (2015) The adoption of Agile Management Practices in a tradiconal Project Enviroment: an IT/IS Case Study. In: IEEE 48 th Hawaii International conference on System Sciences.P. 4446-4453.
- ✓ Alawairdhi, Mohammed.(2016) Agile Development as a Change Management Approach in Software Projects: Applied Case Study. In: IEEE
- ✓ Vriens, Christ; Barto, René. (2008). 7 Years of Agile Management. In: IEEE Agile 2008 Conference. Pg. 390 – 394.

Alguns detalhes importantes

✓ Cronograma

Dia	Semana	Carga Horária
13/04/2018	Sexta-feira	18:50 h – 22:00 h
27/04/2018	Sexta-feira	18:50 h – 22:00 h
11/05/2018	Sexta-feira	18:50 h – 22:00 h
25/05/2018	Sexta-feira	18:50 h – 22:00 h
Total		20 h

Alguns detalhes importantes

✓ Ementa

Manifesto ágil; métodos e frameworks ágeis; iniciação de projetos ágeis; planejamento de projetos ágeis; execução de projetos ágeis; monitoramento e controle de projetos ágeis; e melhoria contínua.

Avaliação

- Participação na Aula
- Exercícios práticos individuais ou em grupos aplicados ao longo das aulas
- Seminários em grupos

Programa

- ✓ Problemática de construir software
- ✓ Disciplina, Agilidade e Perplexidade
- ✓ Abordagem Preditiva X Abordagem Ágil
- ✓ Processo de Gerência de Projetos (MPS.BR)
- ✓ Metodologias Ágeis (Manifesto Ágil)
- ✓ Scrum Conceitos básicos
- ✓ Processo Scrum

Objetivo Principal

- ✓ Fazer uma reflexão sobre:
 - Efetuar um questionamento sobre o quê é efetivamente Ágil.
 - O gerenciamento Ágil significa que não tem documentação
 - Como construir uma gerência ágil

Definição de Software

- ✓ **Stricto Sensu**

- ✓ Software é um conjunto de instruções que governam as ações de uma máquina programável

- ✓ **Lato Sensu**

- ✓ Software é o conjunto de programas de computador, procedimentos e possível documentação e dados associados (ABNT NBR ISO/IEC 25000:2008)

Crise do Software

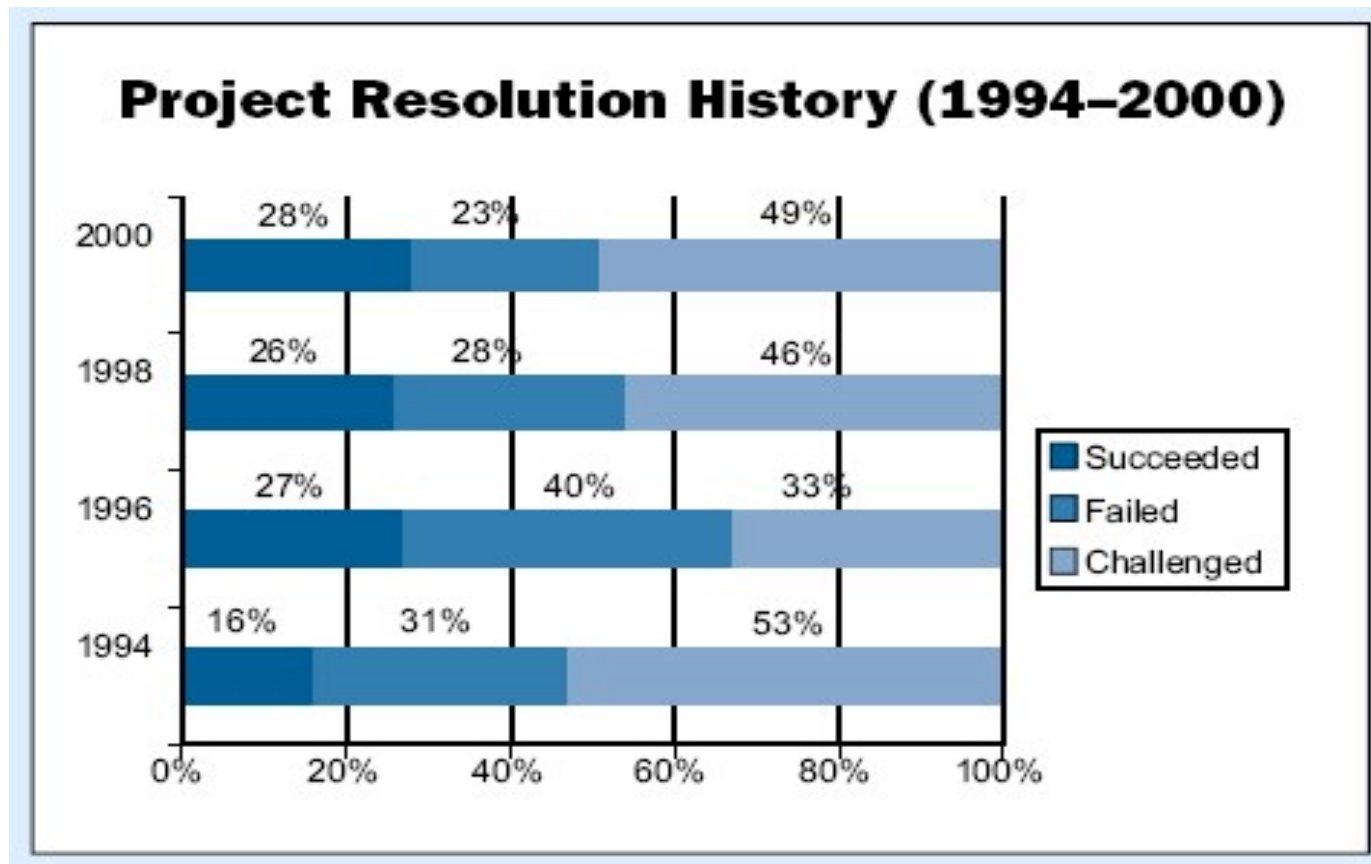
- ✓ Descreve problemas que ocorrem no ambiente de desenvolvimento de software causando atrasos na sua entrega, estouro de orçamento, produtos que não atendam os requisitos do cliente, dificuldades de usar e manter.

Crise do Software

- em 1994, Gibbs publica na Scientific American:
- ✓ para cada 6 grandes projetos de software que são operacionalizados, 2 são cancelados
- ✓ em média, os projetos de desenvolvimento de software estouram seus cronogramas em 50%
- ✓ $\frac{3}{4}$ dos sistemas de grande porte não fornecem a funcionalidade requerida [6]

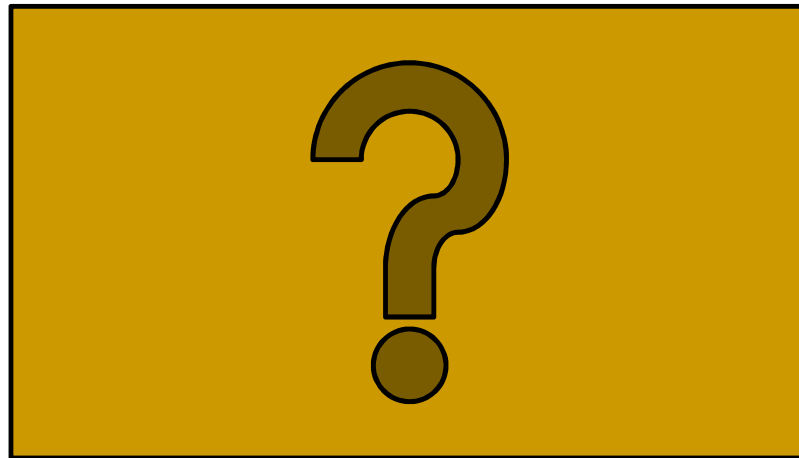
Crise do Software

- Os relatórios do Standish Group também mostram dados relacionados a estes tipos de projetos:



Dados sobre projetos de software. Fonte: Extreme CHAOS (2001) – Standish Group

21 anos depois...



Crise do Software

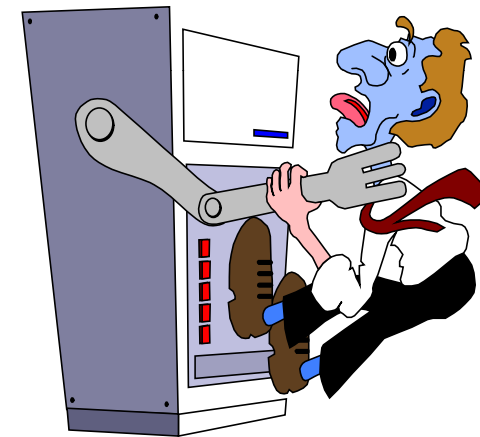
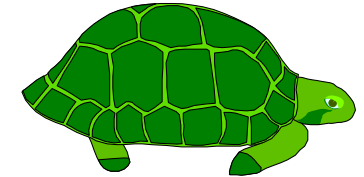
MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Situações Usuais

- Baixa produtividade
- Testes não planejados, não realizados
- Frustração do usuário
- Reclamações, alta taxa de manutenção de sistemas
- Atraso na entrega de sistemas
- Requisitos mal compreendidos
- Perda de dados
- Situações não previstas
- Conflitos entre fornecedor e cliente
- Orçamento insuficiente



Caso Real: Aeroporto de Denver

- Centro de conexões nos EUA
 - Com área de aterrissagem para três jatos ao mesmo tempo
 - Software para sistema de controle de bagagem
 - 29 km de esteira para transportar bagagens de 20 cias aéreas
 - Controle feito por 100 computadores ligados em rede, 5000 sensores, 400 receptores de rádio, 56 leitoras de código de barra
- Tudo orquestrado para enviar as bagagens aos seus destinos de forma correta, segura e rápida, por meio de caminhos inteligentes

Caso Real: Aeroporto de Denver

- Planejamento

- Data de inauguração: Outubro/1993
- Valor do contrato do sistema: US\$ 234 milhões
- Custo do atraso? US\$ 1 milhão/dia

- Realização

- Inauguração: Fevereiro/1995 (com um sistema provisório)
- Custo do sistema provisório: US\$ 63 milhões

Outros Exemplos

- FBI
 - Início em 2001: Base de dados com suspeitos de terrorismo
 - Em Janeiro de 2005: Gastos de US\$ 170 milhões: “not even close to having a working system”
- Ford Motors
 - Início em 2000: Projeto Everest, para substituição de sistema legado
 - Em Agosto de 2004: US\$ 200 milhões acima do orçamento; cancelado (versão beta mais lenta do que o legado)

Outros Exemplos

- McDonald's
 - Início em 1999: Projeto Innovate, com orçamento previsto de US\$ 1 bilhão
 - Cancelado em 2002, após gastos de US\$ 170 milhões
- Voo 965 da American Airlines, de Miami para Cali
 - Projeto do sistema feito para melhorar a interação homem-computador → causou um erro e a tripulação não percebeu
 - Este erro implicou em uma mudança de rota
 - Atingiram uma montanha
 - 159 pessoas de 163 morreram

Outro Caso: Foguete Ariane V

- No dia 6 de junho de 1996, o primeiro voo do Ariane V terminou em falha
- Apesar dos extensivos testes e revisões com milhares de correções, não se detectou o erro
- A função de alinhamento do sistema de referência inercial, que só tem função enquanto o foguete não se destacou do chão, permaneceu operativo e causou a falha.
- Esse programa foi aproveitado do Ariane IV, mas não foi percebido que, com as diferenças de condições, havia a possibilidade de “estouro de campo”

[https://www.youtube.com/watch?
v=PK_yguLapgA](https://www.youtube.com/watch?v=PK_yguLapgA)

Falhas em Projetos de Software

- Por quê projetos falham?
 - Prazos não realistas
 - Mudanças de requisitos
 - Esforço honestamente subestimado
 - Riscos não contingenciados (previsíveis ou não)
 - Dificuldades técnicas
 - Falhas na comunicação da equipe
 - Falta de gerência no projeto

Falhas em Projetos de Software

- Por quê projetos falham?
 - Prazos não realistas
 - Mudanças de requisitos
 - Esforço honestamente subestimado
 - Riscos não contingenciados (previsíveis ou não)
 - Dificuldades técnicas
 - Falhas na comunicação da equipe
 - Falta de gerência no projeto

Falhas em Projetos de Software

- Software é considerado, por muitos executivos, uma das áreas com maiores problemas em organizações de grande porte
- Questões críticas para os executivos sêniores
 - Projetos de software não são estimados nem planejados com precisão aceitável
 - O relato de status geralmente é errado e pode induzir a erros
 - A qualidade e confiabilidade de software são geralmente inaceitavelmente baixas

Os engenheiros de software são otimistas!!!!!!

Falhas em Projetos de Software

- Reclamações contra os executivos sêniores
 - Executivos geralmente rejeitam estimativas precisas e cautelosas
 - Executivos aplicam uma pressão nociva em relação aos cronogramas, que geralmente prejudica a qualidade
 - Executivos adicionam novos requisitos significativos durante o desenvolvimento

Características do Software

- Características essenciais do software
 - Invisibilidade do software
 - Complexidade do software
 - Conformidade do software
 - Modificabilidade do software
- Não é construído por manufatura no sentido clássico
- A maioria é feita sob medida em vez de ser montada a partir de componentes
- Não se desgasta, mas se deteriora

Problema inerente ao desenvolvimento de sw

- Descrição do Cliente:
 - Quero um banheiro com dois mictórios.

O que o fornecedor entregou:



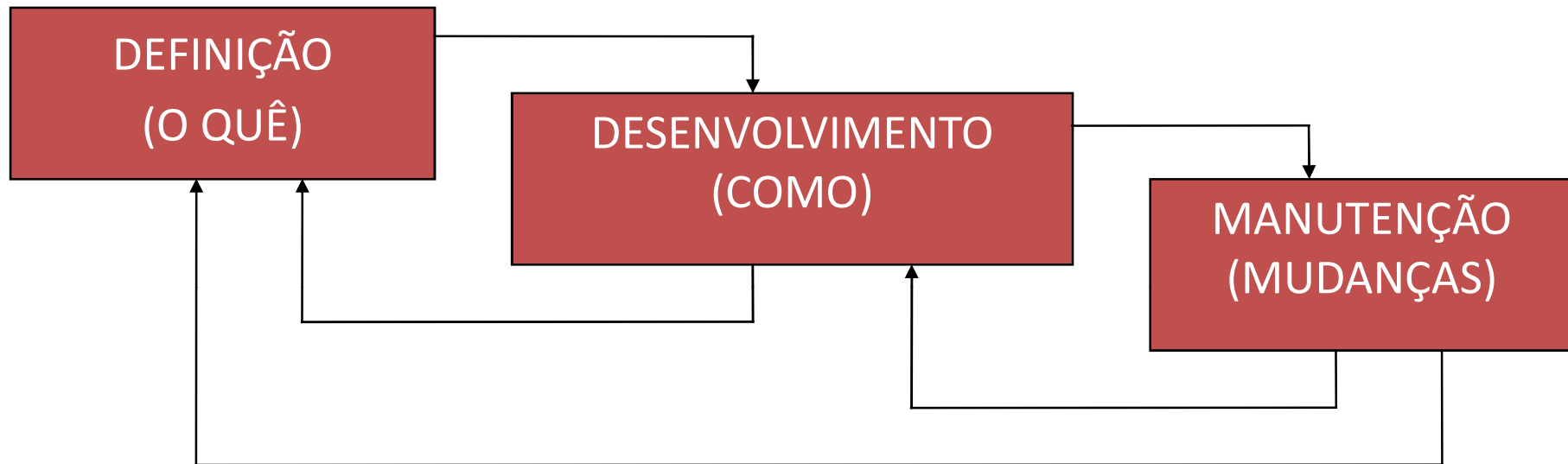
Foi o mesmo fornecedor que entregou esses:



Como podemos resolver esse problema?



Processo de Desenvolvimento de Software



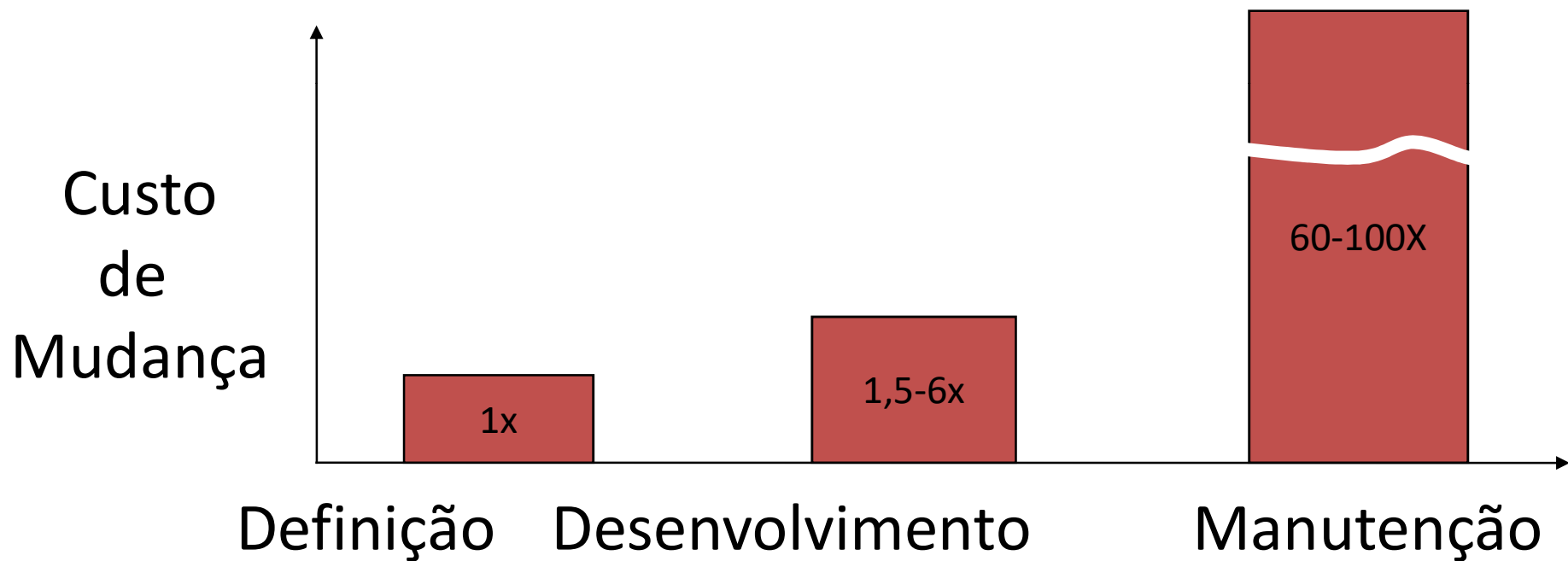
Definição: Análise, Planejamento e Requisitos

Desenvolvimento: Projeto, Codificação, Teste

Manutenção: Correção, Adaptação, Melhoria

Índices de manutenção

Mito: Mudanças são facilmente realizadas por que o software é flexível



Exercício I

Faça agora o planejamento do processo de desenvolvimento do software do novo produto (firmware) que será construído para a empresa NEC.

Discuta com seu grupo e identifique quais são os principais itens que devem compor um planejamento de processo. Procure ser o mais abrangente possível.

Tempo: 30 minutos

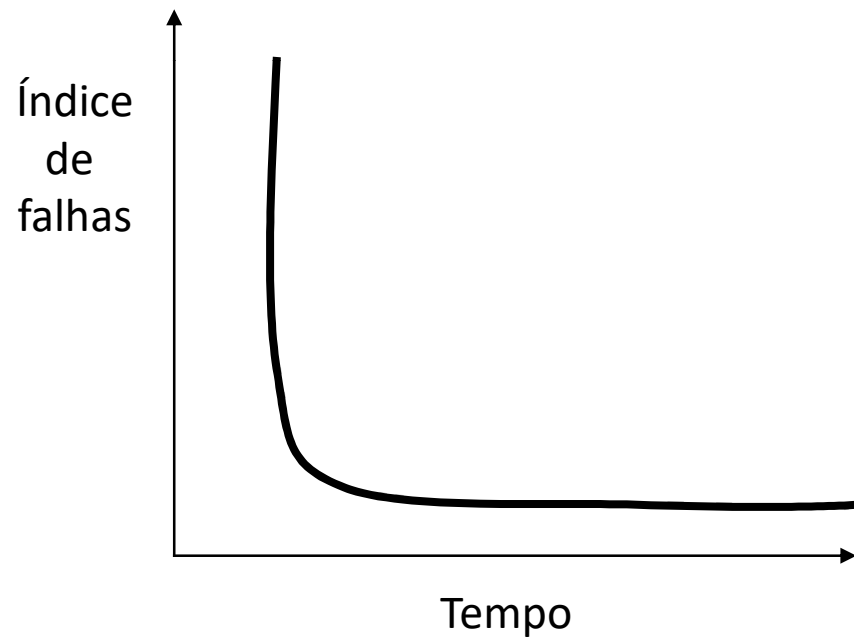
Grupo: 4 pessoas

Entregar por escrito

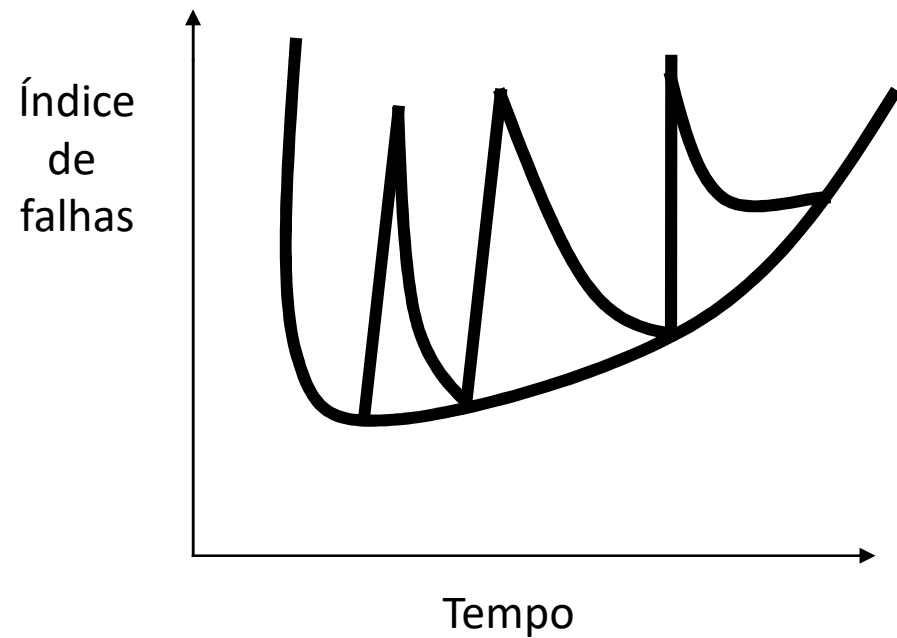
Vida útil do software

Desgaste X Obsolescência

Situação Ideal



Situação Real



Mitos do software

- Mitos de Gerenciamento. Advém de gerentes sobre pressão de orçamento e tempo.
 - Mitos do cliente. Advém de falsas expectativas e insatisfação com o desenvolvedor
 - Mitos do Profissional de desenvolvimento. Advém de se considerar o sw como uma forma de arte. Será que o sw é uma arte ou uma engenharia?
-

Mito do Gerenciamento

- Mito: Já temos um manual repleto de padrões e procedimentos para a construção de sw. Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?
 - Realidade: O manual de padrões pode muito bem existir, mas será que ele é usado? Os profissionais de sw têm conhecimento de sua existência? Ele reflete a moderna prática de desenvolvimento de sw? É completo? Em muitos casos, a resposta a todas estas perguntas é “não”.
-

Mito do Gerenciamento

- Mito: Meu pessoal tem ferramentas de desenvolvimento de sw de última geração; afinal de contas lhes compramos os mais novos computadores.
 - Realidade: É preciso muito mais do que o último modelo de computador para se fazer um desenvolvimento de sw de alta qualidade. As ferramentas CASE são mais importantes do que o hw para se conseguir boa qualidade e produtividade; contudo, a maioria dos desenvolvedores de sw não as usa ainda
-

Mito do Gerenciamento–Horda Mongol

- Mito: Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso.
 - Realidade: O desenvolvimento de sw não é um processo mecânico igual à manufatura. Acrescentar pessoas em um projeto de sw atrasado torna-o ainda mais atrasado. Gasta-se tempo educando os recém-chegados, o que reduz o tempo de desenvolvimento produtivo. Pessoas podem ser acrescentadas, mas somente de uma forma planejada e bem coordenada.
-

Mito do Cliente

- Mito: Uma declaração geral dos objetivos é suficiente para se começar a escrever programas –detalhes são mais tarde.
 - Realidade: Uma definição inicial ruim é a principal causa de fracasso dos esforços de desenvolvimento de sw. Uma descrição formal e detalhada do domínio da informação, função, e do comportamento é fundamental. Essas características podem ser determinadas somente depois de cuidadosa comunicação entre o cliente e o desenvolvedor.
-

Mito do Cliente

- Mito: Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o sw é flexível.
 - Realidade: É verdade que os requisitos de sw se modificam, mas o impacto da mudança varia de acordo com o tempo em que ela é introduzida.
-

Mito do Profissional de Desenvolvimento

- Mito: Assim que escrevemos o programa e o colocarmos em funcionamento, nosso trabalho estará completo.
 - Realidade: Alguém disse certa vez que “quanto mais cedo se começa a ‘escrever o código’, mais tempo demora para que se consiga terminá-lo”. Os dados da indústria indicam que entre 50 e 70% de todo o esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.
-

Mito do Profissional de Desenvolvimento

- Mito: Enquanto não tiver o programa “funcionando”, eu não terei realmente nenhuma maneira de avaliar sua qualidade.
 - Realidade: Um dos mecanismos mais efetivos de garantia de qualidade de sw pode ser aplicado desde o começo de um projeto – a revisão técnica formal. As revisões de sw são um “filtro da qualidade” que têm sido consideradas mais eficientes do que a realização de testes para a descoberta de defeitos.
-

Mito do Profissional de Desenvolvimento

- Mito: A única coisa a ser entregue em um projeto bem-sucedido é o programa funcionando.
 - Realidade: Um programa funcionando é somente uma parte de uma configuração de sw que inclui vários outros elementos. A documentação forma os alicerces para um desenvolvimento bem-sucedido e fornece um guia para a tarefa de manutenção do sw.
-

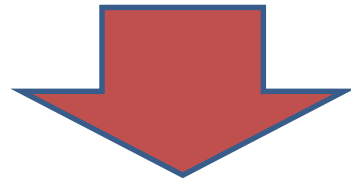
Qual caminho a seguir?



Qual caminho a seguir?

Mercado Globalizado
Altamente competitivo
Redução de Custos
Diminuição de Prazos
Conformidade de Requisitos
Demandas de Negócio
Necessidade de mudanças rápidas

Situação



Qualidade = Agilidade + Disciplina

Disciplina, Agilidade e Perplexidade

- ✓ Disciplina é o fundamento para qualquer empreendimento de sucesso
 - Atletas treinam
 - Músicos praticam
 - Artesões aperfeiçoam as técnicas
 - Engenheiros aplicam processos
- ✓ As pessoas que não possuem disciplina podem ter sucesso ocasional, mas ele será dependente do talento natural.

Disciplina, Agilidade e Perplexidade

- ✓ A força e segurança que vem da disciplina apoiam o empreendimento quando as coisas são difíceis.
- ✓ A disciplina cria memórias bem organizadas, histórias e experiências.
- ✓ A agilidade é a contrapartida da disciplina
- ✓ Onde a disciplina está institucionalizada, a força e agilidade liberam o poder de invenção

Disciplina, Agilidade e Perplexidade

- ✓ A agilidade aplica a memória e história para ajustar a novos ambientes, reagindo e adaptando, levando vantagens de oportunidades inesperadas e atualizam a base para o futuro.
- ✓ Qualquer negócio bem sucedido requer agilidade e disciplina. (sw, artes e esporte)
- ✓ Disciplina sem agilidade é burocracia
- ✓ Agilidade sem disciplina é caos

Disciplina, Agilidade e Perplexidade

- ✓ Perplexidade é o estado de quem demonstra hesitação, diante de uma situação complexa, em que não se sabe qual decisão tomar:
 - ficou chocado diante da perplexidade dos fatos;
 - ficou em estado de perplexidade ao presenciar aquela cena horrível.
 - A perplexidade paralisa!!!!
- ✓ Produzir documentos para se manter a qualidade
- ✓ Risco de apoiar-se menos nos documentos para manter os sistemas e aumentar a produtividade

Controvérsia

- ✓ **Pró-agilista:** "os metodologistas tradicionais são um bando de "pés na lama" que preferem produzir documentação sem falhas em vez de um sistema que funcione e atenda às necessidades do negócio."
- ✓ **Pró-processo:** "os metodologistas de pouco peso (lightweight), quer dizer, os metodologistas ágeis são um bando de hackers pretenciosos que vão acabar tendo uma grande surpresa ao tentarem transformar seus brinquedinhos em sw de porte empresarial."

Premissas sobre a construção de sw

- ✓ É difícil afirmar antecipadamente quais requisitos irão persistir e quais sofrerão alterações. É difícil prever de que maneira as prioridades do cliente sofrerão alterações conforme o projeto avança.
- ✓ Para muitos tipos de sw, o projeto e a construção são atividades que devem ser realizadas em sequência. É difícil prever o quanto de trabalho de projeto será necessário (complexidade) antes que a sua construção seja implementada.
- ✓ Análise, projeto, construção e testes não são previsíveis quanto gostaríamos que fosse (planejamento).

Exercícios: Questões que não querem calar?

- ✓ Como criar projetos capazes de administrar a imprevisibilidade?
- Responda a questão acima em grupos de 4 pessoas