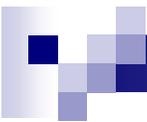


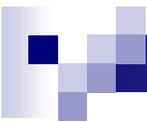
Metodologias Ágeis de Desenvolvimento

Fernando Trinta



Contextualização

- *A Engenharia de software vêm recorrentemente enfrentando o cenário onde ...*
 - *as aplicações são cada vez mais complexas...*
 - *o tempo de desenvolvimento é cada vez menor...*
 - *há necessidade de diminuição de custos ...*
 - *busca constante pelo aumento da qualidade.*



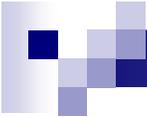
Evolução da Engenharia de Software

- *Modelos e processos de desenvolvimento, refinados a partir de outras engenharias*
 - Cascata*
 - Espiral*
 - Processo Unificado*
- *Software é construído através de um **planejamento** e uma metodologia de gerenciamento*



Porém...

- *Processos tradicionais tornaram-se “pesados” para a engenharia de software*
 - Muita burocracia*
 - Muita documentação*
 - Pouca flexibilidade a mudanças no projeto*
 - Não contemplam o cenário atual*
 - Conflito de interesses*

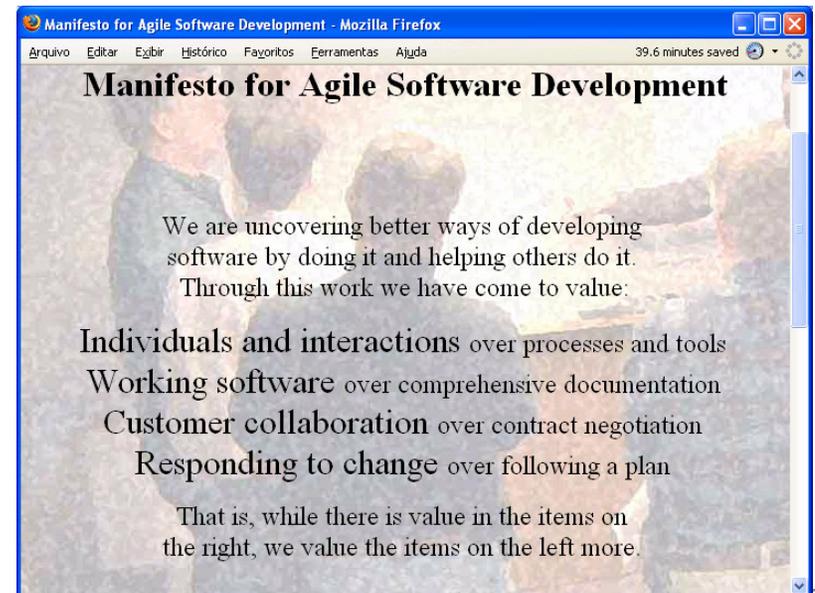


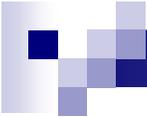
Problemas

- *Software não é algo previsível ou imune a mudanças*
- *Desenvolver software é como desenvolver novos produtos*
 - *Desenvolver é como criar uma receita. Produzir é seguir os passos de uma receita.*
- *O desenvolvimento é um processo de aprendizado, que envolve tentativas e erros.*
 - *Como a manufatura previsível não pode ser comparada ao software, dificilmente as práticas e valores enraizados nesse paradigma trazem algum benefício*
- *Desenvolvimento de software depende muito mais das pessoas e da comunicação.*

Manifesto Ágil

- *17 gurus da engenharia de software*
- *Declaração de 4 valores básicos para o desenvolvimento*
- *<http://www.agilemanifesto.org>*
- *<http://agilealliance.org>*





Manifesto Ágil

Indivíduos e interações são mais importantes que processos e ferramentas.

Software funcionando é mais importante do que documentação completa e detalhada.

Colaboração com o cliente é mais importante do que negociação de contratos.

Adaptação a mudanças é mais importante do que seguir o plano inicial

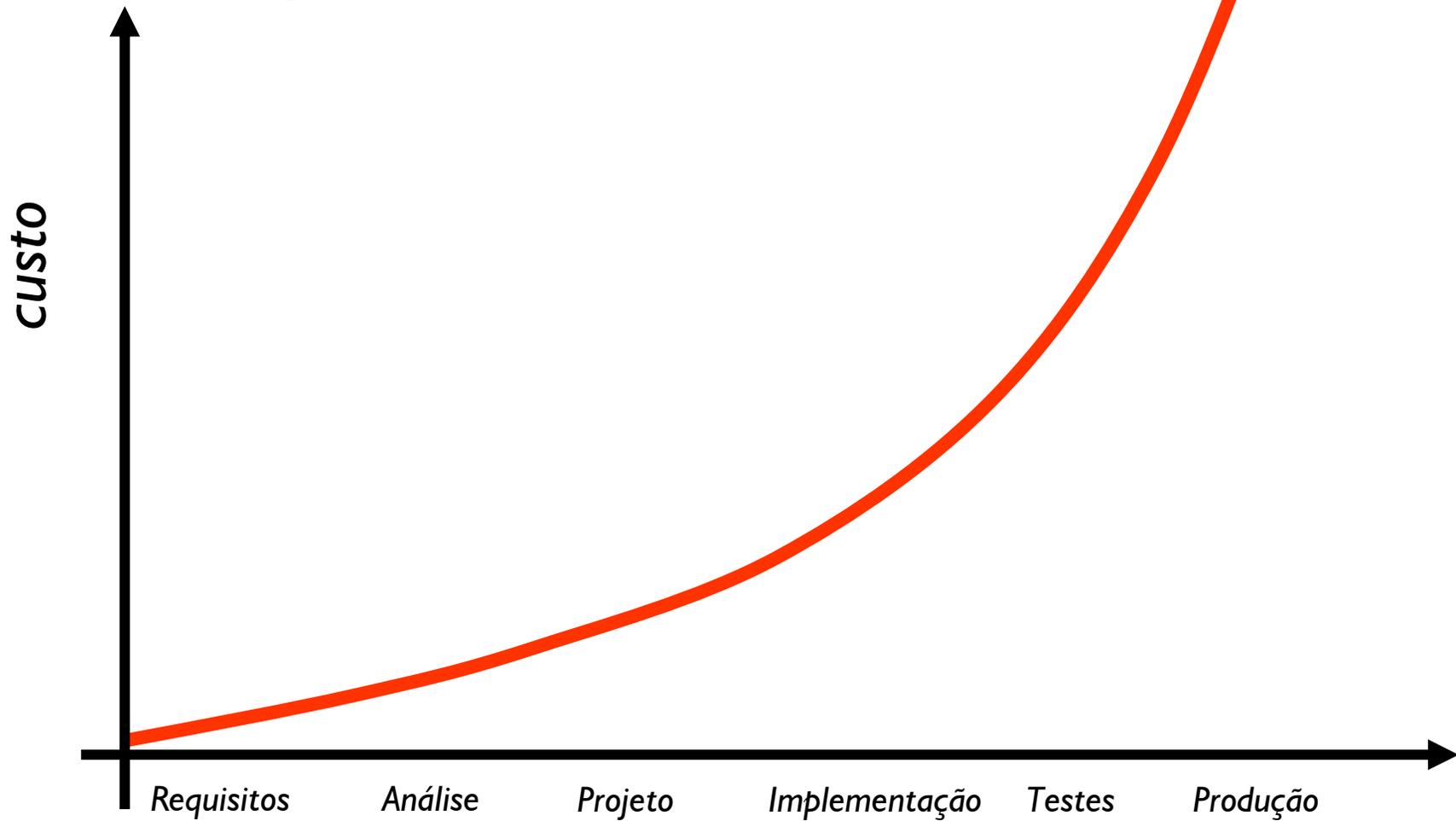


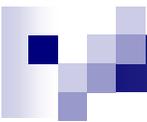
Mudanças e metodologias pesadas

- *Mudanças são caras*

Mudanças e metodologias pesadas

- *Mudanças são caras*





Mudanças e metodologias pesadas

- *Mudanças são caras*
- *“Ok, então vamos evitá-las”*
- *Como?!*
 - Muito planejamento*
 - Ser fiel a este planejamento*
 - Precaução em relação a mudanças*
 - Documentar bastante*
- *Mas...*



Mudanças são inevitáveis

■ *Requisitos mudam ...*

- Conhecimento sobre a aplicação se dá a medida da construção do mesmo*

■ *Organizações mudam ...*

- Tomadas de decisões estratégicas*

■ *Pessoas mudam ...*

- Novos clientes, gerentes, desenvolvedores*

■ *Tecnologias mudam ...*

Mudanças são inevitáveis

- *Requisitos mudam ...*

- Conhecimento sobre a aplicação se dá a*
m

- *Orgo*

- T*

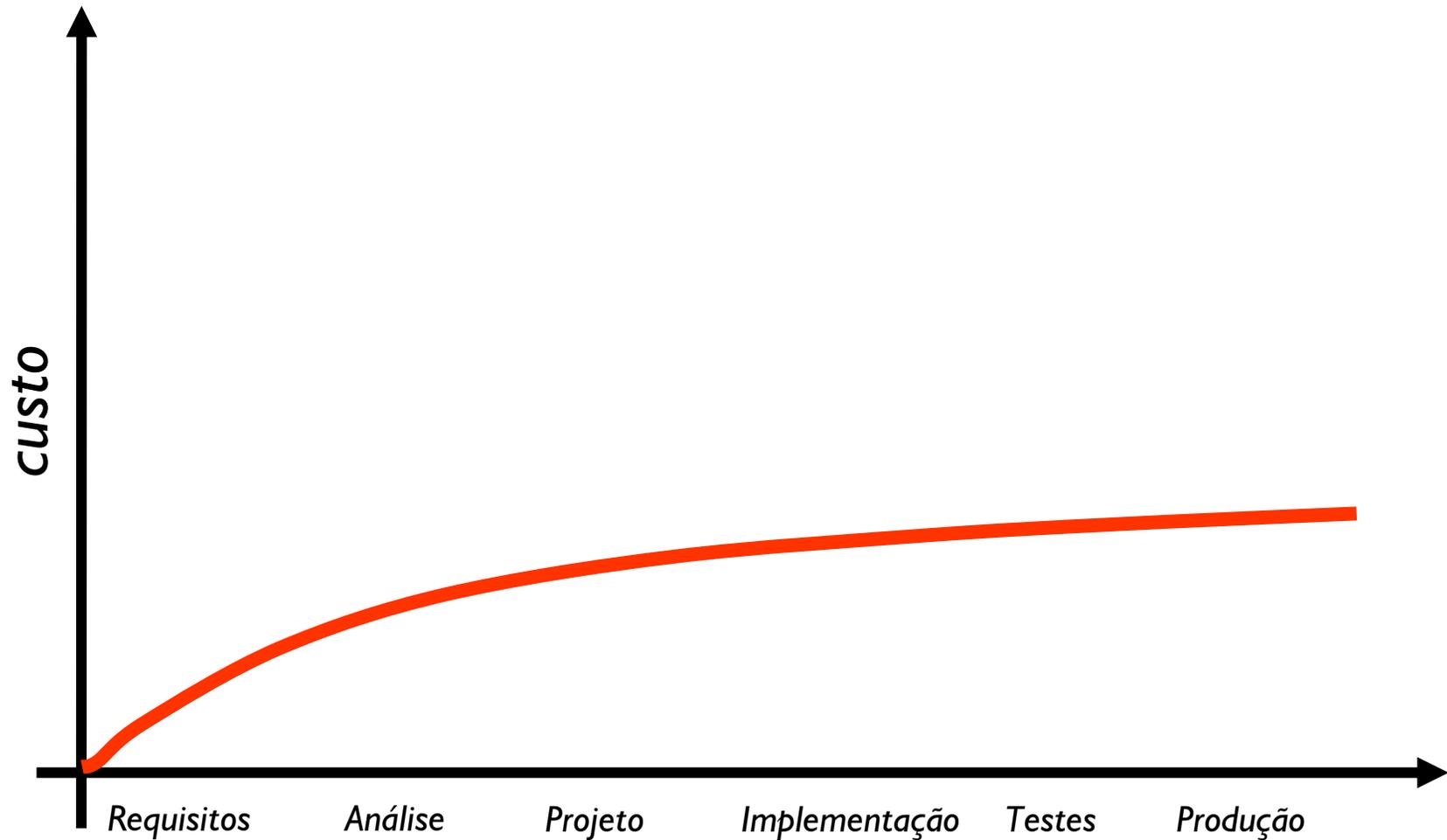
- *Pess*

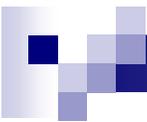
- Novos clientes, gerentes, desenvolvedores*

- *Tecnologias mudam ...*

**Ok, então vamos
“abraçar” mudanças**

Mudanças em Metodologias Ágeis





Metodologias Agéis

- *Coleção de práticas, guiadas por valores e princípios, que podem ser aplicadas por profissionais de TI no seu cotidiano*
- *Princípios*
 - *Comunicação*
 - *Processo Interativo*
 - *Simplicidade*
- *Mudança Cultural*
 - *mudança de atitudes de empresas e profissionais*



Metodologias Agéis

- *Ter um comportamento mais **adaptativo**, que **preditivo***
- *É algo que funciona na **prática***
- *Foco nas **pessoas**, não no processo*
- *Ganho interesse crescente das empresas, com casos de sucesso sendo relatados*

Abordagem Tradicional X Abordagem Ágil

| Abordagem Tradicional | Abordagem Ágil |
|---|---|
| Preditiva: detalhar o que ainda não é bem conhecida; | Adaptativa: conhecer o problema e resolver o crítico primeiro; |
| Rígida: seguir especificação predefinida a qualquer custo; | Flexível: adaptar-se a requisitos atuais, que podem mudar; |
| Burocrática: controlar sempre, para alcançar objetivo planejado; | Simplista: fazer algo simples de imediato, e alterar no futuro se for necessário; |
| Orientado a processos: segui-los possibilita garantir a qualidade; | Orientado a pessoas: motivadas, comprometidas e produtivas; |
| Documentação: é a garantia de confiança | Comunicação: é a garantia de confiança; |
| Sucesso: é entregar o planejado | Sucesso: é entregar o desejado; |
| Gerência: sinônimo de “comando-controlado”, voltada para o trabalho em massa com ênfase no papel do gerente, com planejamento e disciplina fortes. | Gerência: sinônimo de “liderança-orientação”, voltada para o trabalho do conhecimento, com ênfase na criatividade, flexibilidade e atenção às pessoas. |

Analogia...

Sistema Toyota de Produção

| Produção artesanal | Produção em massa | Manufatura enxuta |
|---|--|--|
| Ciclo de vida dos produtos longo, medido em muitos anos | Ciclo de vida dos produtos longo, medido em anos | Ciclo de vida dos produtos curto, medido em meses |
| Produção fixa, inflexível | Produção fixa, inflexível | Produção flexível, orientada para o cliente |
| Peças não intercambiáveis | Peças intercambiáveis | Peças intercambiáveis |
| Demanda estável, conhecida, crescente | Demanda estável, conhecida, crescente | Demanda errática, instável, que é confrontada com a capacidade de produção |
| Ciclos de produção longos | Ciclos de produção longos | Ciclos de produção curtos |
| Fabricação para estoque | Fabricação para estoque | Fabricação sob pedido do cliente |
| Decisões de gerenciamento em curto prazo | Decisões de gerenciamento em curto prazo | Decisões enxutas de gerenciamento visando obter, simultaneamente, vantagem competitiva e satisfação do cliente |
| Pequena variedade, baixo volume | Pequena variedade, alto volume | Qualquer variedade e volume, como requerido pelo cliente |
| Empresa opera isoladamente | Empresa opera isoladamente | Formação de alianças enxutas para aperfeiçoar a capacitação da empresa |
| Ênfase na qualidade do produtos | Ênfase no corte de custos | Ênfase simultânea em custo, entrega, qualidade, flexibilidade, projeto e capacitação |
| Trabalhadores muito qualificados | Trabalhadores desqualificados | Trabalhadores multi-qualificados, altamente treinados |
| Relação fornecedor-comprador fraca | Relação fornecedor-comprador fraca, conflituosa | Parcerias enxutas entre comprador e fornecedor |

Fonte: Adaptado de Lepikson (1998); com base em Womack *et al.* (1992); Hammer e Champy (1994) e Pine II (1994).

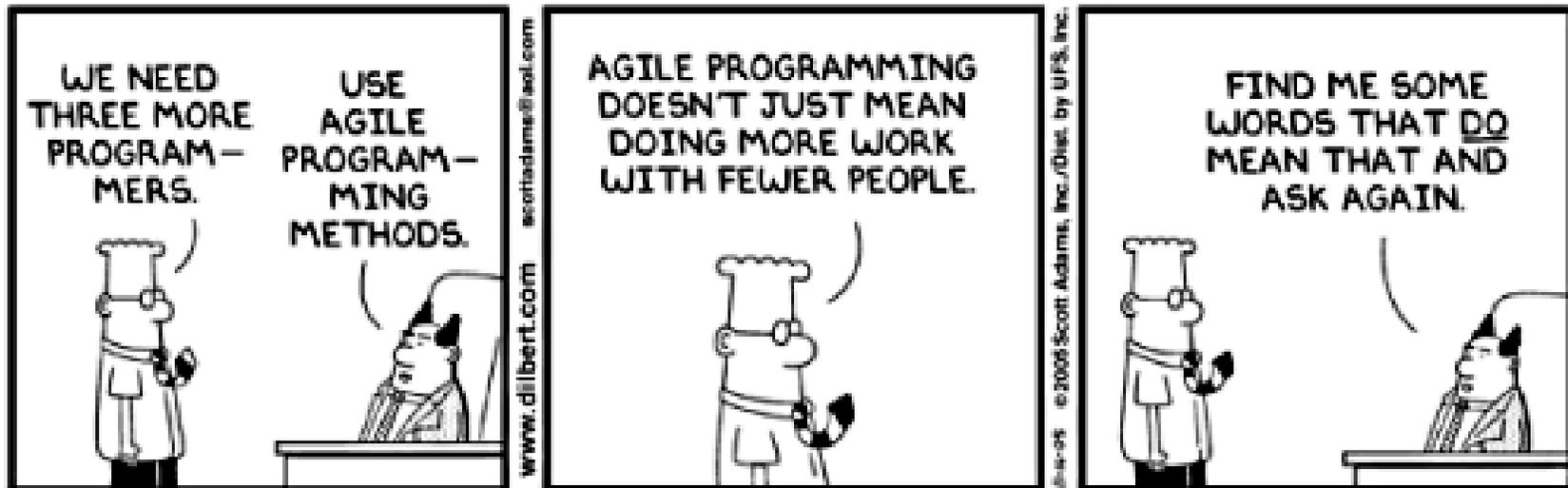
Toyota passa a GM e isso pode ser bom para o Estado

Desde 1931, quando a General Motors passou à frente da Ford na fabricação e venda de automóveis nos EUA e no mundo, havia uma frase que sintetizava a importância da montadora como símbolo do poderio industrial dos Estados Unidos: o que é bom para a GM é bom para os EUA. Pois 76 anos depois, os japoneses, que foram os grandes derrotados pelos estadunidenses na II Guerra Mundial, destronaram a General Motors. Katsuaki Watanabe, presidente da Toyota, anunciou que a sua empresa vendeu mais automóveis e caminhões em todo o mundo, no primeiro trimestre de 2007, 2,35 milhões de carros e veículos pesados. Suplantada a GM, encerrou-se o mais longo domínio de uma empresa num setor industrial em todos os tempos. Que a Toyota acabaria chegando à liderança era questão de meses, no máximo um ano, previam, em 2006, os especialistas do setor automobilístico, pois a indústria dos Estados Unidos vem declinando, com a chegada dos carros pequenos e econômicos. Mas, os clientes da superpotência gostam de carros grandes e não necessariamente econômicos, apesar do barril de petróleo a US\$ 65. Os japoneses da Toyota fizeram carros médios/grandes e, o melhor de tudo,

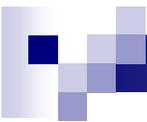
modelos menos consumidores de gasolina. Excesso de benefícios trabalhistas, descuido com a qualidade, veículos grandes consumidores de combustível, a lentidão em admitir o avanço da concorrência e a mudança na preferência por carros pequenos foram minando a liderança da GM desde duas décadas, um gigante industrial que nos anos 60 dominava o mundo e parecia que jamais seria ultrapassado. Foi quando a Toyota literalmente surgiu das cinzas e dos escombros bélicos do Japão para criar, nos anos 70 e 80, uma indústria padrão de automóveis bons, baratos e confiáveis. Embora continue liderando no cobiçado mercado chinês, a GM tem imensas dificuldades de vendas nos Estados Unidos e na Europa, além de nunca ter conseguido se firmar no próprio Japão. No Brasil, a Toyota tem 3,3% do mercado de automóveis e comerciais leves, tendo vendido 15.645 unidades no primeiro trimestre deste ano. A General Motors detém 21,2% do mercado, com 99.684 unidades comercializadas. A Toyota quer atingir 10% do mercado nacional em 2010. Para tanto, implantará uma fábrica de veículos compactos, sendo hoje a quinta montadora no ranking

brasileiro. O presidente Katsuaki Watanabe disse que o Brasil deverá abrigar uma linha de montagem de um carro que chamou de "ultra-baixo custo", com preço externo na faixa dos US\$ 6,6 mil, algo em torno de R\$ 14 mil. O local mais cotado para a instalação é o Rio Grande do Sul. Desde 1998, a Toyota tem fábrica em Indaiatuba/SP, que produz o renomado sedan Corolla e a perua Fielder. Agora em maio, a Toyota lança o Corolla 1.8, com motor flexível, usando álcool ou gasolina. Será a última montadora do Brasil a adotar a tecnologia, que está disponível desde 2003. Existe um curioso adágio na psicanálise, segundo o qual quando a pessoa certa faz a coisa errada, ela é certa. Quando a pessoa errada faz a coisa certa, ela é errada. Pois ao assumir uma atitude de superioridade em relação às demais fábricas, a GM se esqueceu de que deveria ter um planejamento atento e irrepreensível. Para a GM, os japoneses foram as pessoas erradas - sem tradição em automóveis - fazendo a coisa certa, mas que deu muito certo, contrariando o raciocínio citado. Custou caro à General Motors, poderá ser uma bênção para o Rio Grande do Sul, lá em Guaíba, em 2008.

Porém, não confundir...

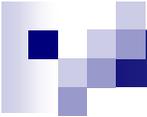


© Scott Adams, Inc./Dist. by UFS, Inc.



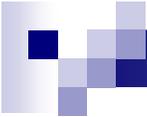
Exemplos de Metodologias Ágeis

- *Programação Extrema (XP)*
- *Scrum*
- *Método Dinâmico de Desenvolvimento de Sistemas (DSDM)*
- *Desenvolvimento Adaptativo de Software*
- *Crystal*
- *Desenvolvimento guiado por Funcionalidades*
- *Programação pragmática*
- *Desenvolvimento guiado por testes*
- *Lean Software Development*



Exemplos de Metodologias Ágeis

- *Programação Extrema (XP)*
- *Scrum*
- *Método Dinâmico de Desenvolvimento de Sistemas (DSDM)*
- *Desenvolvimento Adaptativo de Software*
- *Crystal*
- *Desenvolvimento guiado por Funcionalidades*
- *Programação pragmática*
- *Desenvolvimento guiado por testes*
- *Lean Software Development*



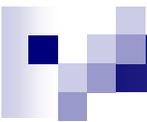
Comum a estas metodologias

- *Times pequenos*
 - *Máximo de 20 pessoas*
- *Pessoal qualificado*
- *Forte Comunicação*
 - *Face a face, de preferência*
- *Iterações curtas*
 - *Semanas e não meses*

Programação Extrema



Copyright: United Feature Syndicate Inc.



Programação extrema

- *Mais conhecida e aplicada*

- *Projeto C3 (Chrysler) - Kent Beck, Ward Cunningham and Ron Jeffries (1996)*

 - <http://www.xprogramming.org>

- *Valores:*

 - *Comunicação, Simplicidade, Feedback, Respeito e Coragem*

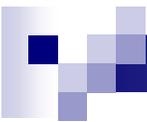
- *12 Práticas:*

 - *Programação em pares. Refatoramento, Projeto Simples, Testes, Propriedade Coletiva, Padrões de Codificação, Integração Contínua, Ritmo Sustentável, Metáforas, Jogo de Planejamento, Pequenos Releases, Testes do usuário e time integrado*



Por que ao extremo?!

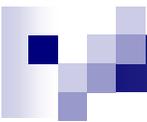
- *Se houvesse tempo, o que desenvolvedores gostariam de realizar?!*
 - Mais testes?*
 - Mais projeto e arquitetura?*
 - Melhorar qualidade?*
 - Aumentar meu conhecimento sobre a aplicação?*



Ok então!!!

■ *Levar todas as boas práticas do desenvolvimento ao Extremo*

- *É importante testar, então vamos fazer testes sempre!*
- *É importante projetar, então vamos fazer disso uma rotina!*
- *É importante integrar as partes, então vamos integrar o máximo possível!*
- *É importante que as iterações sejam curtas, então vamos deixá-las realmente curtas!*



Programação Extrema

■ *Comunicação*

- fazer as soluções chegarem as pessoas*

■ *Coragem*

- compartilhar código, mexer em algo que já esteja funcionando, recomeçar*

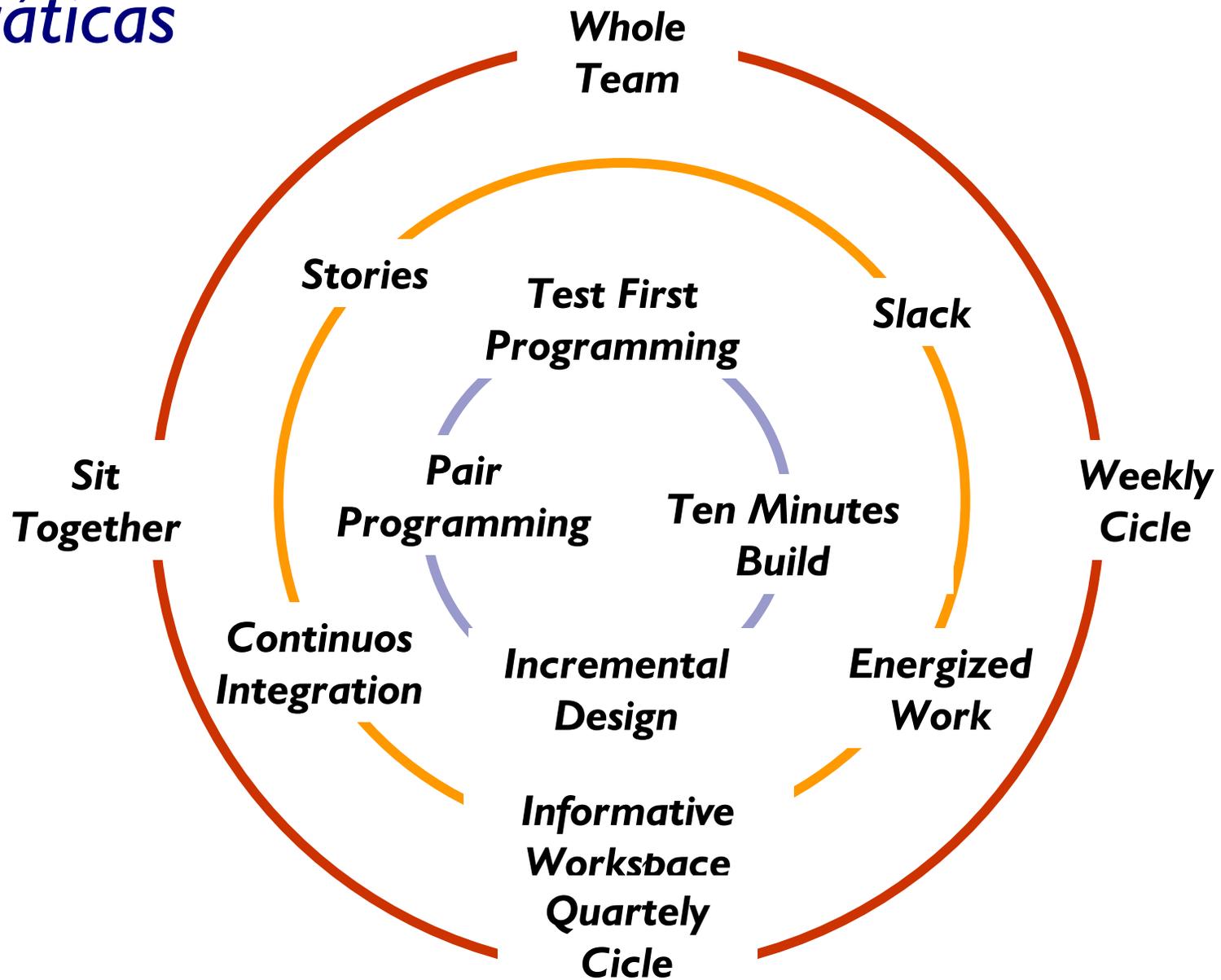
■ *Simplificidade*

- sempre pensar no mais simples*

■ *Respeito*

- à equipe e a si mesmo*

Práticas





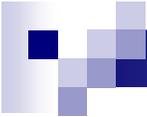
Ponto de vista de fora do processo

- *Ceticismo*
- *É uma zona... Não há disciplina*
- *E a documentação?!*
- *Programação em pares eleva os custos*
- *Testes são inconsistentes*



Ponto de vista de fora do processo

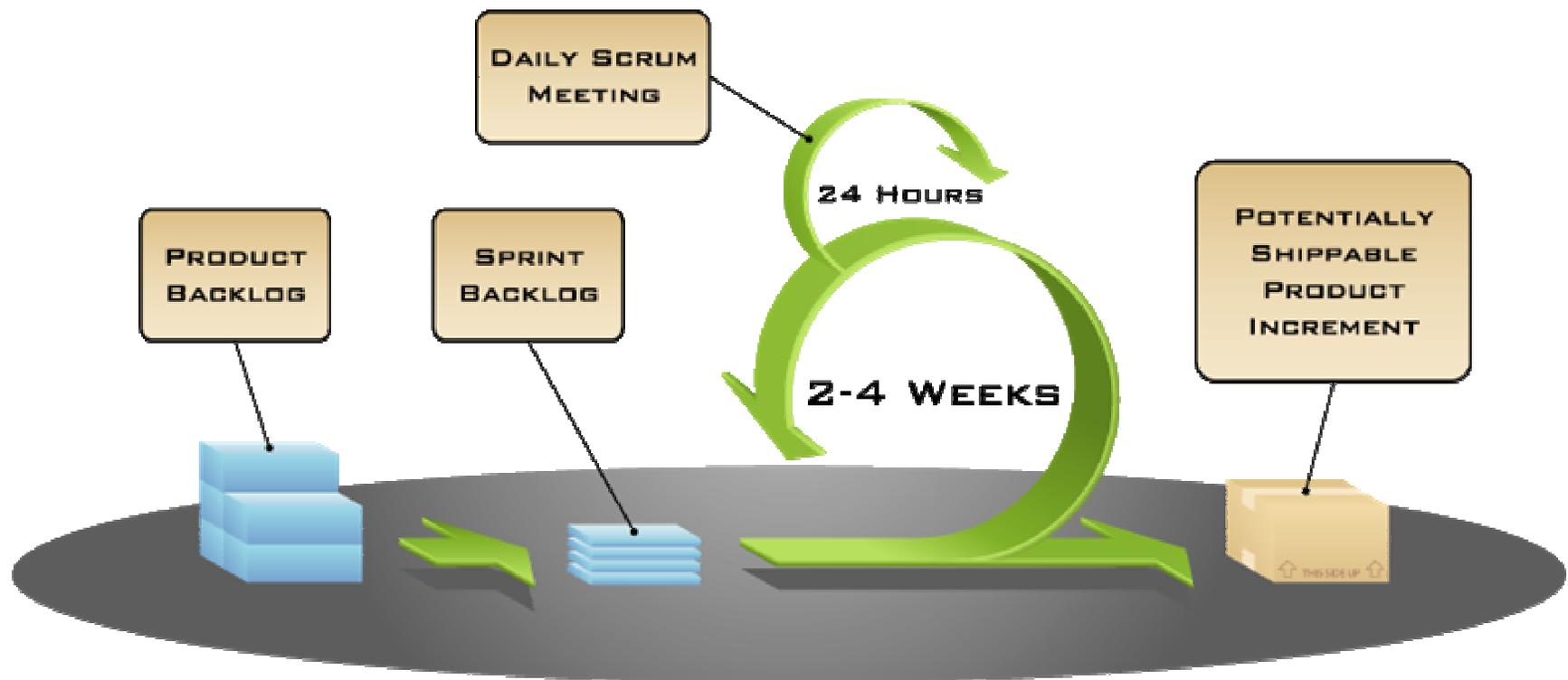
- *Ceticismo*
- *É uma zona... Não há disciplina*
- *E a documentação?!*
- *Programação em pares eleva os custos*
- *Testes são inconsistentes*
- *Puxa!! Né que funciona!!*



Scrum

- *Jeff Sutherland, Ken Schwaber (1993)*
 - <http://www.controlchaos.com/>
- *Foco em gerência e rastreamento*
 - *Pode ser combinado com métodos mais prescritivos*
- *É um processo simples*
 - *Processo, artefatos e regras são poucas e fáceis de entender*

Esquema base do Scrum

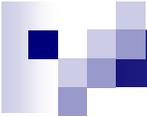


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE



Visão Geral

- **Scrum** divide o desenvolvimento em Sprints de 30 dias.
- Equipes pequenas, de até 7 pessoas, são formadas por projetistas, programadores, engenheiros e gerentes de qualidade.
- Estas equipes trabalham em cima de funcionalidade definidas no início de cada Sprint. A equipe toda é responsável pelo desenvolvimento desta funcionalidade



Papeis Scrum

■ *ScrumMaster:*

- Responsável pelo sucesso da aplicação do Scrum na empresa*

■ *Product Owner:*

- Responsável por apresentar os interesses de todos os stakeholders*

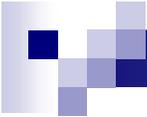
■ *Time:*

- Responsável pelo desenvolvimento e escolha das funcionalidades a serem desenvolvidas*



Ciclo do Scrum

- *A cada iteração (sprint) de 30 dias, onde o os envolvidos escolhem as funcionalidades a serem desenvolvidas*
- *Estas informações alimentam o product backlog e o sprint backlog*
- *Durante os dias, são realizadas reuniões diárias curtas*



Reunião diária

- *ScrumMaster faz as seguintes perguntas para cada membro do time:*
 - O que você fez desde a última reunião diária do Scrum relacionada a este projeto?*
 - O que você irá fazer desde agora até a próxima reunião diária do Scrum relacionada a este projeto?*
 - O que está impedindo você de realizar o seu trabalho o mais efetivamente possível?*
- *Os membros devem responder apenas a estas perguntas para não estender a reunião*



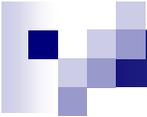
Princípios do Scrum

- *A Gerência do projeto é compartilhada por todos os papéis*
 - *Time se auto gerencia*
- *Todos se comprometem com o BackLog Product*



Benefícios Scrum

- *Maior integração entre os membros da equipe*
- *Rápida solução de problemas*
 - *Promovem o compartilhamento de conhecimento*
- *Progresso medido continuamente*
 - *Minimização de riscos*



Scrum x XP

- **Scrum** é interessante porque fornece um mecanismo de informação de status que é atualizado continuamente, e porque utiliza a divisão de tarefas dentro da equipe de forma explícita.
- **Scrum e XP** são complementares pois Scrum provê práticas ágeis de gerenciamento enquanto XP provê práticas integradas de engenharia de software.



Ok... Então tá tudo resolvido!!!

- *Vamos todos utilizar metodologias ágeis...*

- *Não é bem assim...*



Principais críticas

- *Não há falta de estrutura e documentação necessárias*
- *Requer a adoção de muita mudança cultural*
- *Difícil estabelecer metas de longo prazo*
- *É necessário desenvolvedores sênior*
- *Dificuldades em definir um contrato*



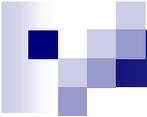
Cenários onde metodologias precisam ser melhor avaliadas

- *Grandes equipes*
 - *difícil comunicação*
- *Equipes distribuídas*
- *Cliente não disponível*



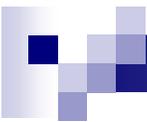
Aplicabilidade

- *Requisitos ainda não conhecidos*
- *Novas tecnologias*
- *Equipes inexperientes*



Conclusões

- *Projetos complexos e urgentes necessitam de uma nova abordagem*
 - *Métodos ágeis surgiram como uma nova alternativa no processo de desenvolvimento de software*
- *Métodos ágeis demandam...*
 - *mudança de cultura*
 - *muita disciplina*
 - *muita competência*
 - *muita flexibilidade*



Conclusões

- *Cada vez mais grandes empresas estão aderindo às práticas e comportamento ágil em busca de qualidade, produtividade e lucratividade.*
- *Tema já comum em conferências workshops da área*
 - *Agile Conference: <http://www.agile.org>*



Pontos para discutir...

- *Depois de tanto esforço da engenharia de software, métodos ágeis podem ser um retrocesso?!*
- *RUP pode ser ágil?*