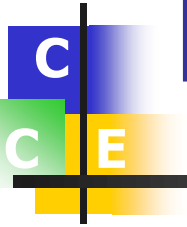


# Tecnologias Atuais de Desenvolvimento de *Software*



## Desenvolvimento Ágil

Prof. Luiz Antônio

lpereira@uninet.com.br



# Desenvolvimento Ágil

## Agenda

---

- Bibliografia Básica
- Motivação
- Fundamentos
- XP



# Desenvolvimento Ágil

## Bibliografia Básica

---

- *Engenharia de Software*; Roger S. Pressman, 5a. Edição, 2002.
- *Extreme Programming Explained. Embrace Change*; Kent Beck, 1999.
- *Planning Extreme Programming*; Kent Beck and Martin Fowler; Addison Weley; 2000.
- *Extreme Programming Explained. Embrace Change (2nd Edition)*; Kent Beck & Cynthia Andres, 2004.
- *Agile Project Management with Scrum*; Ken Schwaber; Microsoft Press, 2004.
- Google.



# Desenvolvimento Ágil

---

- Agilidade refere-se à(\*)
  - Velocidade das operações que ocorrem nas organizações;
  - Velocidade nas respostas aos clientes;
  - Tempo reduzido de ciclo.

(\*)<http://ccs.mit.edu/21c/iokey.html>

# Desenvolvimento Ágil

## Motivação

- Contexto típico de desenvolvimento de S/W:
  - Negócio não é bem conhecido e/ou é dinâmico (necessidade constante de manutenção evolutiva e adaptativa);
  - Documentação externa permanece desatualizada em boa parte do ciclo de desenvolvimento;
  - As manutenções são feitas com base no código (*ninguém* consulta/atualiza a documentação externa);
  - Necessidade de disponibilidade rápida de código;

Continua...



# Desenvolvimento Ágil

## Motivação

---

- Contexto típico de desenvolvimento de S/W (cont.):
  - Desenvolvimento centrado no código;
  - Demanda por maiores níveis de qualidade do produto (qualidades externa e interna);
  - Se há processo, ele é rígido e burocratizado;

Continua...

# Desenvolvimento Ágil

## Motivação

- Contexto típico de desenvolvimento de S/W (cont.):
  - Assunções clássicas:
    - Usuários especificam exatamente o que querem;
    - Desenvolvedores entregam o que os usuários especificaram.
  - A situação real típica:
    - Usuários não sabem o que querem;
    - Desenvolvedores não entregam o que prometeram.



# Desenvolvimento Ágil

## Motivação

---

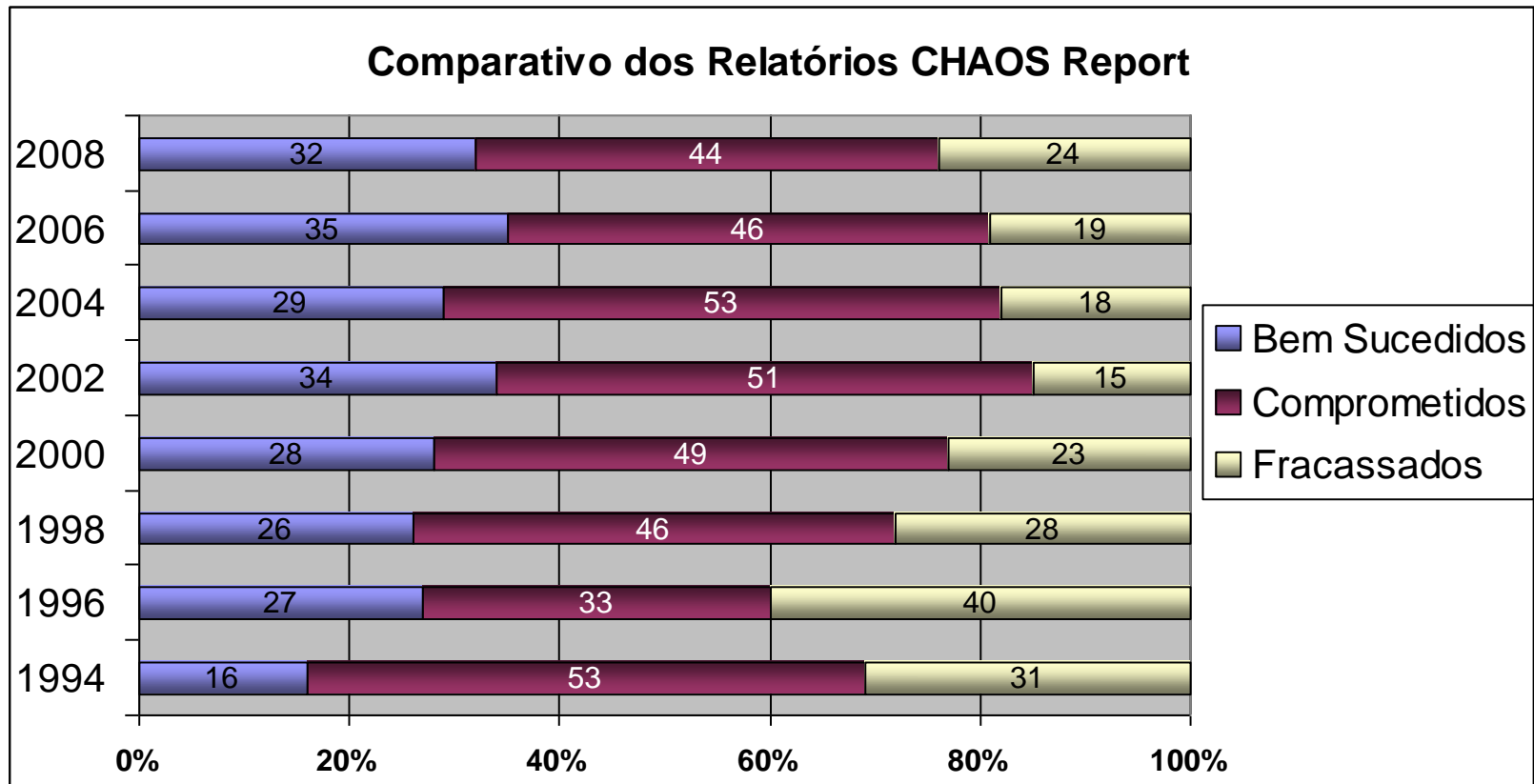
- Algumas constatações importantes:
  - A entrega de *software* em prazos e custos estabelecidos quase nunca é conseguida;
  - Excesso de formalismo nos modelos de processo propostos nos últimos 30 anos;
  - Desenvolvimento de *software* é arriscado e difícil de ser gerenciado.

Continua...



# Desenvolvimento Ágil

## Motivação



Fonte: Relatório CHAOS Summary 2009

Continua...



# Desenvolvimento Ágil

## Motivação

---

- Algumas constatações importantes (cont.):
  - Planejamento das etapas é fundamental para a execução/construção de algo;
  - Uso de métodos conhecidos e aprovados também é uma medida efetiva;
  - Idem com relação ao controle constante do produto e do processo em cada etapa do projeto.



# Desenvolvimento Ágil

## Motivação

---

- Precisa-se de novas metodologias que, dentre outras coisas:
  - Tratem adequadamente requisitos vagos e “mutantes”;
  - Mantenham a liberdade necessária para que os programadores trabalhem de forma efetiva e prazerosa;
  - Produzam *software* de qualidade;
  - Diminuem os encargos da equipe com documentação;
  - Mantenham as expectativas dos usuários atendidas e em níveis realizáveis;
  - Mantenham os projetos gerenciáveis;
  - Tenham base científica.



# Desenvolvimento Ágil

## Fundamentos

---

- Metodologias ágeis focam em:
  - Redução dos custos de mudanças nos requisitos;
  - Entrega do sistema em pequenas liberações:
    - O sistema **não** fica pronto **só** no final do prazo...
    - ... O sistema está “pronto” ao longo do prazo.
- Metodologias ágeis propõem desenvolvimento de *software* de forma mais rápida, porém com qualidade.



# Desenvolvimento Ágil

## Fundamentos

---

- Metodologias Ágeis:
  - Movimento iniciado por programadores experientes e consultores em desenvolvimento de *software*;
  - Questionam e se opõe a uma série de mitos/práticas adotadas em abordagens tradicionais de Engenharia de *Software* e Gerência de Projetos.



# Desenvolvimento Ágil

## Fundamentos

---

- Manifesto Ágil (\*):
  - Assinado por 17 desenvolvedores (Kent Beck *et al*) em Utah – EUA - em fevereiro de 2001.
  - Descreve a essência de um conjunto de abordagens para desenvolvimento de *software* criadas ao longo da última década.

(\*)<http://agilemanifesto.org>



# Desenvolvimento Ágil

## Fundamentos

---

- Manifesto Ágil – Assunções:
  - Indivíduos e interações são mais importantes que processos e ferramentas;
  - *Software* funcionando é mais importante que documentação completa e detalhada;
  - Colaboração com o cliente é mais importante que (re)negociação de contratos;
  - Adaptação a mudanças é mais importante que seguir o plano inicial.



# Desenvolvimento Ágil

## Fundamentos

---

- Manifesto Ágil - Objetivos:
  - Satisfazer o cliente, entregando, rapidamente e com frequência, sistemas com algum valor.
    - Entregar versões funcionais em prazos curtos;
    - Estar preparado para mudanças de requisitos;
    - Fazer com que o pessoal de negócios e desenvolvedores trabalhem juntos;
    - Trocar informações através de conversas diretas, *face-to-face*.



# Desenvolvimento Ágil

XP

- XP:
  - (Breve) Introdução
  - O foco nos riscos
  - O que é
  - Custos das modificações e correções de rumo
  - Valores, princípios e práticas da XP



# Desenvolvimento Ágil

## XP – Introdução

---

- XP = *Extreme Programming*;
- Metodologia de desenvolvimento de *software* sendo aperfeiçoada nos últimos anos(\*);
- As práticas adotadas são “as melhores práticas de engenharia de *software* levadas a níveis extremos”;
- Originou-se das experiências de Kent Beck, Ward Cunningham e Ron Jeffries (também signatários do Manifesto Ágil) no projeto C3 (Chrysler, com Smalltalk e GemStone), 1996-1999.

(\* ) Correspondendo às edições 1 e 2 do livro “*Extreme Programming Explained: Embrace Change*”



# Desenvolvimento Ágil

## XP – O foco nos riscos

---

- Riscos são vistos como o “problema básico”
  - Prazo de entrega não é cumprido (projetos se tornam potencialmente “canceláveis”);
  - *Software* é posto em produção mas, ao deteriorar, os custos de manutenção são altos;
  - *Software* é posto em produção com erros e os custos de manutenção corretiva são altos;
  - Necessidades reais não atendidas (requisitos foram ignorados ou mal entendidos);
  - Rotatividade da equipe (por desinteresse) X produtividade.



# Desenvolvimento Ágil

## XP – O foco nos riscos

---

- Como XP trata os riscos
  - Prazo de entrega não cumprido:
    - Número maior de entregas (entregas parciais);
    - Ciclos de desenvolvimento menores ⇒ gerência do projeto mais fácil;
    - Funcionalidades de maior prioridade para o usuário são implementadas antes;
    - Funcionalidades de maior prioridade, implementadas antes, “cativam” os usuários pois satisfazem mais imediatamente suas principais necessidades;
    - Se houver diminuição do escopo para redução de prazo, as funcionalidades eliminadas tenderão ser as menos importantes (pois são as que ficam para o final).

Continua...



# Desenvolvimento Ágil

## XP – O foco nos riscos

---

- Como XP trata os riscos (cont.)
  - *Software* deteriorado e altos custos de manutenção:
    - Modificações feitas com maior frequência;
    - Testes aplicados a cada modificação;
    - “A XP sempre mantém o sistema em excelente condição” (Beck);
  - Muitos erros no *software* em produção e altos custos de manutenção corretiva:
    - Testes são executados a cada alteração;
    - Testes envolvem a participação dos usuários (testando as funcionalidades)

Continua...



# Desenvolvimento Ágil

## XP – O foco nos riscos

---

- Como XP trata os riscos (cont.)
  - Necessidades reais não atendidas:
    - Usuário é convidado a fazer parte do time;
    - Especificação do projeto é continuamente refinada e, com isso, permite-se um entendimento mais uniforme das necessidades do usuário por parte de todos os integrantes da equipe.

Continua...



# Desenvolvimento Ágil

## XP – O foco nos riscos

---

- Como XP trata os riscos (cont.)
  - Rotatividade da equipe (por desinteresse) X produtividade:
    - Programadores estimam suas próprias produtividades que são comparadas com o que é observado. O *feedback* melhora a capacidade de estimativa e diminui a frustração;
    - Encorajamento da integração entre membros da equipe;
    - Incorporação de um modelo de tratamento da rotatividade baseado na ajuda aos mais novos, assistência mútua e aceitação de níveis cada vez maiores de responsabilidade.



# Desenvolvimento Ágil

## XP – O que é

---

- XP é (\*):
  - Uma tentativa de conciliação entre humanidade e produtividade;
  - Um mecanismo de mudança social;
  - Um caminho para a melhora;
  - Um estilo de desenvolvimento;
  - Uma disciplina de desenvolvimento de *software*.

(\*) Programação Extrema Explicada, Ed. 1 – Kent Beck





# Desenvolvimento Ágil

## XP – Custo das modificações

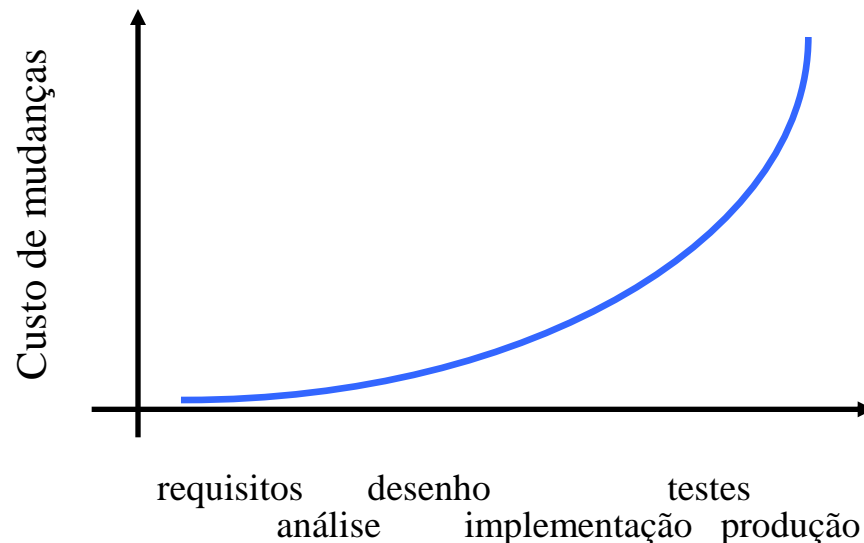
---

- O custo das modificações
  - Premissa *clássica*: Em processos tradicionais, os requisitos devem ser conhecidos *a priori*, já que...

# Desenvolvimento Ágil

## XP – Custo das modificações

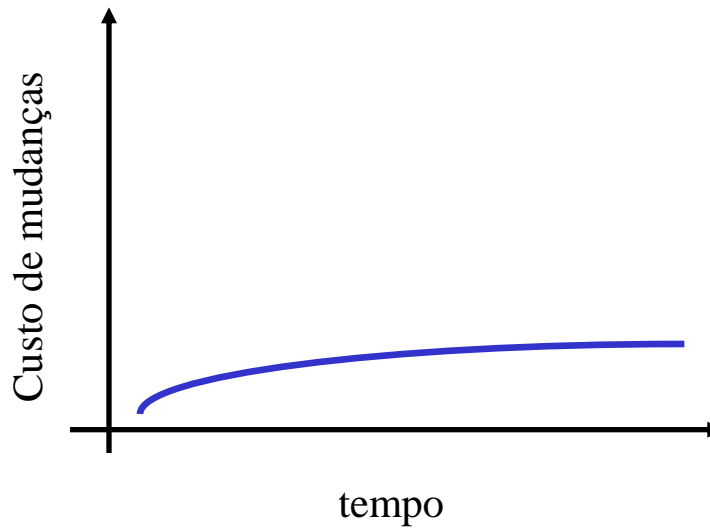
- ... segundo o entendimento tradicional, mudanças de requisitos têm um impacto no custo cada vez maior quanto mais tarde, no ciclo de desenvolvimento, as mesmas ocorrerem.



# Desenvolvimento Ágil

## XP – Custo das modificações

- O objetivo principal da XP é diminuir os custos de mudanças.





# Desenvolvimento Ágil

## XP – Custo das modificações

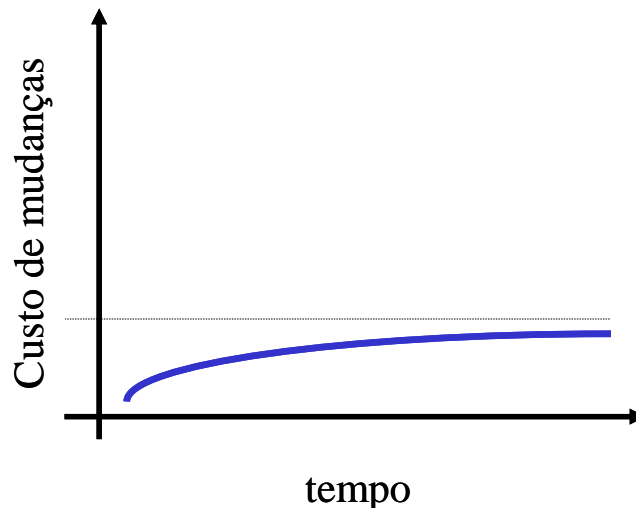
---

- Um custo para modificações que aumenta vagarosamente com o tempo é a premissa básica do XP.
  - A contrapartida (uma curva íngreme de custo X tempo) torna XP impossível!

# Desenvolvimento Ágil

## XP – Custo das modificações

- Com tecnologias e práticas apropriadas, é possível obter-se uma curva de custo de mudanças X tempo do projeto tendendo a um custo constante (máximo).





# Desenvolvimento Ágil

## XP – Custo das modificações

---

- E quais são essas tecnologias e práticas?
  - Objetos são a tecnologia chave
    - Encapsulamento provê manutenibilidade;
    - Modificações de comportamento, sem alteração de código existente, podem ser (mais facilmente) implementadas através de mudanças nas trocas de mensagens entre os objetos.
  - Projeto simples, sem elementos extras (antecipação de necessidades, flexibilidades não solicitadas, etc.);
  - Testes automatizados aplicados logo após as implementações das modificações;
  - Experiência prática desenvolvida na aplicação de modificações em projetos, provendo auto-confiança na hora em que elas são necessárias.



# Desenvolvimento Ágil

## XP – Custo das modificações

---

- Nesse contexto, as atitudes dos desenvolvedores ágeis são:
  - Implementam *agora* somente o que precisam *agora*;
  - Tomam as grandes decisões o mais tarde possível (quem sabe não será preciso tomá-las, de fato?);
  - Não implementam flexibilidade desnecessária num dado momento (não antecipam necessidades – quem sabe se serão mesmo necessárias?).



# Desenvolvimento Ágil

## XP – Correções de rumo

---

- Constatações:
  - O que muda em um *software*?
    - Os requisitos;
    - A tecnologia;
    - O projeto;
    - O time;
    - ...
  - Mudanças são inevitáveis. É importante saber lidar com elas;
  - Pequenas (às vezes grandes) e constantes mudanças são (quase inevitavelmente) necessárias – a metáfora de se dirigir em uma estrada reta – é o paradigma da XP.





# Desenvolvimento Ágil

XP – Valores, princípios e práticas

---

- XP se baseia (atualmente) em
  - 5 valores;
  - 14 princípios e
  - 24 (13+11) práticas.



# Desenvolvimento Ágil

## XP – Os valores

---

- Os cinco valores da XP são:
  - Comunicação
  - Simplicidade
  - *Feedback*
  - Coragem
  - Respeito



# Desenvolvimento Ágil

## XP – Os valores

---

- Valor 1: Comunicação
  - Sucesso/insucesso dos projetos está associado à boa/má comunicação entre membros da equipe e entre eles e usuários;
  - A melhor forma de comunicação é face-a-face;
  - “Treinador” percebe falhas na comunicação e atua na correção.



# Desenvolvimento Ágil

## XP – Os valores

---

- Valor 2: Simplicidade
  - A preocupação com a (real) necessidade de *features* do sistema deve ser enfatizada;
  - Fazer a coisa mais simples que possa funcionar e ser útil;
  - Aposta no “faça o simples e modifique amanhã, se necessário” e não ao “faça o complicado hoje, mesmo correndo o risco de ser não tão útil amanhã”.



# Desenvolvimento Ágil

## XP – Os valores

---

### ■ Valor 3: *Feedback*

- Respostas do sistema quanto ao seu estado atual são importantes;
- Testes são as perguntas; resultados dos testes são as respostas (*feedback* de alterações efetuadas);
- Clientes definem necessidades e programadores as implementam, dando o *feedback* da implementação aos clientes que dão o *feedback* do resultado observado aos programadores...
- *Feedback* é fundamental na busca de uma solução de qualidade.



# Desenvolvimento Ágil

## XP – Os valores

---

- Valor 4: Coragem
  - Para alterar o que precisa ser alterado visando a um bem maior para o projeto;
  - Para refazer o que precisa ser feito;
  - Para voltar atrás e buscar melhores alternativas.



# Desenvolvimento Ágil

## XP – Os valores

---

- Valor 5: Respeito
  - Membros de um time devem se respeitar;
  - Membros de um time devem respeitar o trabalho de cada um;
  - Membros de um time devem respeitar o usuário.



# Desenvolvimento Ágil

## XP – Princípios

---

- Valores (vagos, subjetivos) precisam ser transformados em *princípios* (guias) bem definidos para que possam ser usados.
- Princípios são a *ponte* entre os valores e as práticas.





# Desenvolvimento Ágil

## XP – Princípios

---

### ■ Humanidade

- *Software* é desenvolvido por pessoas  $\Rightarrow$  fatores humanos influenciam a qualidade;
- Beck afirma que as necessidades das pessoas são, segurança, reconhecimento, acolhimento pelo grupo, crescimento e habilidade de entender e ser entendido pelos outros.

### ■ A Economia

- Produzir *software* é agregar valor ao negócio;
- O *software* deve estar disponível para uso o quanto antes (pois a idéia é agregar valor ao negócio o quanto antes).

Continua...



# Desenvolvimento Ágil

## XP – Princípios

---

- **Benefício mútuo**
  - Tanto as empresas quanto seus colaboradores devem ser beneficiários das soluções e das mudanças.
- **Similaridade**
  - Deve se buscar o uso de soluções similares em contextos diferentes (reúso).
- **Melhora contínua**
  - A história do “está bom, mas pode melhorar”,
  - Vale para a qualidade e para as funcionalidades.

Continua...



# Desenvolvimento Ágil

## XP – Princípios

---

- **Diversidade**
  - Equipe formada por talentos em diferentes áreas;
  - Diferenças devem ser gerenciadas para não causarem conflitos.
- **Reflexão**
  - Análise constante dos sucessos e das falhas (lições aprendidas)...
  - ... entretanto deve-se fazer mais do que pensar.

Continua...



# Desenvolvimento Ágil

## XP – Princípios

---

- Fluxo (de *software*)
  - Todas as atividades visam à produção de software; não é o caso de haver uma seqüência de atividades com a produção do *software* somente ao final.
- Oportunidade
  - Problemas dever ser vistos como oportunidades para melhorias.
  - Não basta corrigir os problemas, mas aprender dando soluções para eles.

Continua...



# Desenvolvimento Ágil

## XP – Princípios

---

- Redundância
  - Para as questões críticas deve-se pesquisar mais de uma solução;
  - Defeitos devem ser perseguidos de várias formas (programação em pares, testes automáticos, envolvimento com usuário real...)
- Falhas
  - Não ter medo de falhar;
  - Falhas ajudam a aprender;
  - Falhas propiciam a busca por uma solução melhor;
  - Melhor tentar e falhar do que não agir.

Continua...



# Desenvolvimento Ágil

## XP – Princípios

---

### ■ Qualidade

- Sempre ao máximo  $\Rightarrow$  não baixar a qualidade para ganhar velocidade;
- Qualidade aumenta a auto-estima do time;
- Qualidade não é perfeccionismo (lembrar que ação é melhor do que contemplação).

### ■ Passos de bebê

- Grandes mudanças, concebidas por um longo tempo, são perigosas;
- Melhor é ir em passos curtos na direção certa;
- Não confundir com “ir devagar”.

Continua...



# Desenvolvimento Ágil

## XP – Princípios

---

- Responsabilidade aceita...
  - ... e não imposta;
  - “Obrigado, mas eu passo...” deve ser algo compreensível.



# Desenvolvimento Ágil

## XP – As práticas

---

- *Práticas* são atitudes e atividades concretas que o programador XP faz no dia-a-dia. Elas garantem a aplicação dos valores.
- Na versão inicial do XP havia 12 práticas (as originais) que continuam válidas até hoje, mas que vêm sendo revistas nos últimos anos.





# Desenvolvimento Ágil

## XP – As práticas correntes

---

- O conjunto atual de práticas está dividido em dois grupos:
  - **13 Práticas primárias**
    - São independentes das práticas atuais de sua equipe.
    - São seguras e cada uma delas pode gerar melhoria imediata.
    - Começar por elas, ir para as decorrentes depois.
  - **11 Práticas decorrentes (ou corolárias)**
    - Dependem da aplicação efetiva das práticas primárias;
    - É difícil aplicá-las sem antes se ter domínio sobre as práticas primárias.



# Desenvolvimento Ágil

## XP – As práticas correntes

---

- Todas as 24 práticas são importantes;
- Devem ser aplicadas em suas plenitudes.



# Desenvolvimento Ágil

## XP – As práticas correntes

---

- **Práticas primárias:**
  - **Todos juntos:** trabalhar em um ambiente grande, para toda a equipe se ver (*War Room*).
  - **Time completo:** incluir na equipe pessoas com as habilidades necessárias para o sucesso do projeto.
  - **Transparência da informação:** manter as informações correntes do projeto em local de fácil visualização (e.g. um mural).
  - **Trabalho energizado:** manter ritmo de trabalho sustentável. Ter vida fora do trabalho (☺).
  - **Programação em pares:** duas pessoas programando juntas, num processo de interação constante.
  - **User Stories:** planejar e controlar o desenvolvimento através de “pedaços” de funcionalidades do negócio, tais como vistas pelos usuários.

Continua...



# Desenvolvimento Ágil

## XP – As práticas correntes

---

- **Práticas primárias (cont.):**
  - **Ciclo semanal** – Planejar o trabalho de cada semana: no início de cada semana é feita uma reunião para definição de que histórias serão implementadas na semana.
  - **Ciclo trimestral** – Planejamento feito em uma escala mais macro através de encontros trimestrais. Garante-se conformidade com os objetivos gerais do projeto.
  - **Colocar folgas** – Inclua alguns itens opcionais no cronograma. Se algo der errado, eles podem ser removidos.
  - **Ten minute-build** – Faça o *build* e rode os testes de 10 em 10 minutos, no máximo.
  - **Integração contínua** – Integre e teste as mudanças a cada duas horas, no máximo.

Continua...



# Desenvolvimento Ágil

## XP – As práticas correntes

---

- **Práticas primárias (cont.):**
  - ***Test first-programming*** – Formular testes e programar para satisfazê-los.
  - ***Design incremental*** – Invista no *design* somente o necessário para resolver os problemas de hoje.



# Desenvolvimento Ágil

## XP – As práticas correntes

---

### ■ Práticas decorrentes

- Envolvimento do cliente real
  - Beneficiários da solução fazem parte do time e participam do planejamento semanal e trimestral.
- Entrega incremental
  - Evitar o “tudo ou nada”. A substituição de um sistema legado deve ser gradual.
- Continuidade da equipe
  - O time deve permanecer o mesmo em vários projetos (por questões de integração, familiarização, etc.)
- *Shrinking teams* (times que encolhem)
  - Na medida em que um time se torna mais capaz e produtivo, mantenha a mesma carga individual de trabalho e libere os membros livres para novos projetos.

Continua...



# Desenvolvimento Ágil

## XP – As práticas correntes

---

### ■ Práticas decorrentes (cont.)

- Análise de causa-raiz
  - Resolver os defeitos e as causas dos defeitos.
- Código compartilhado
  - Qualquer um do time pode alterar qualquer parte do código (*propriedade coletiva* do código).
- Código e testes
  - São os únicos artefatos permanentes do projeto e, portanto, devem ser preservados. Os outros documentos podem ser gerados a partir deles.
- Base de código única
  - Só existe uma única versão oficial do sistema. *Branches* não sobrevivem por muito tempo.

Continua...



# Desenvolvimento Ágil

## XP – As práticas correntes

---

### ■ Práticas decorrentes (cont.)

- Implantação diária
  - A cada noite uma nova versão do *software* deve entrar em produção. Deve-se evitar *gaps* muito grandes entre a versão em desenvolvimento e a versão em produção.
- Contrato de escopo negociado
  - Um contrato de desenvolvimento deveria ter fixados o prazo, os custos e a qualidade. O escopo seria negociado durante o desenvolvimento;
  - Ter-se mini-contratos seria melhor para evitar-se riscos.
- *Pay-per-use*
  - Clientes usualmente pagam por cada *release* ⇒ querem menos *releases* e mais funcionalidades por *release*;
  - Fluxo de desenvolvimento deve coincidir com fluxo de dinheiro.





# Desenvolvimento Ágil

## XP – As práticas correntes

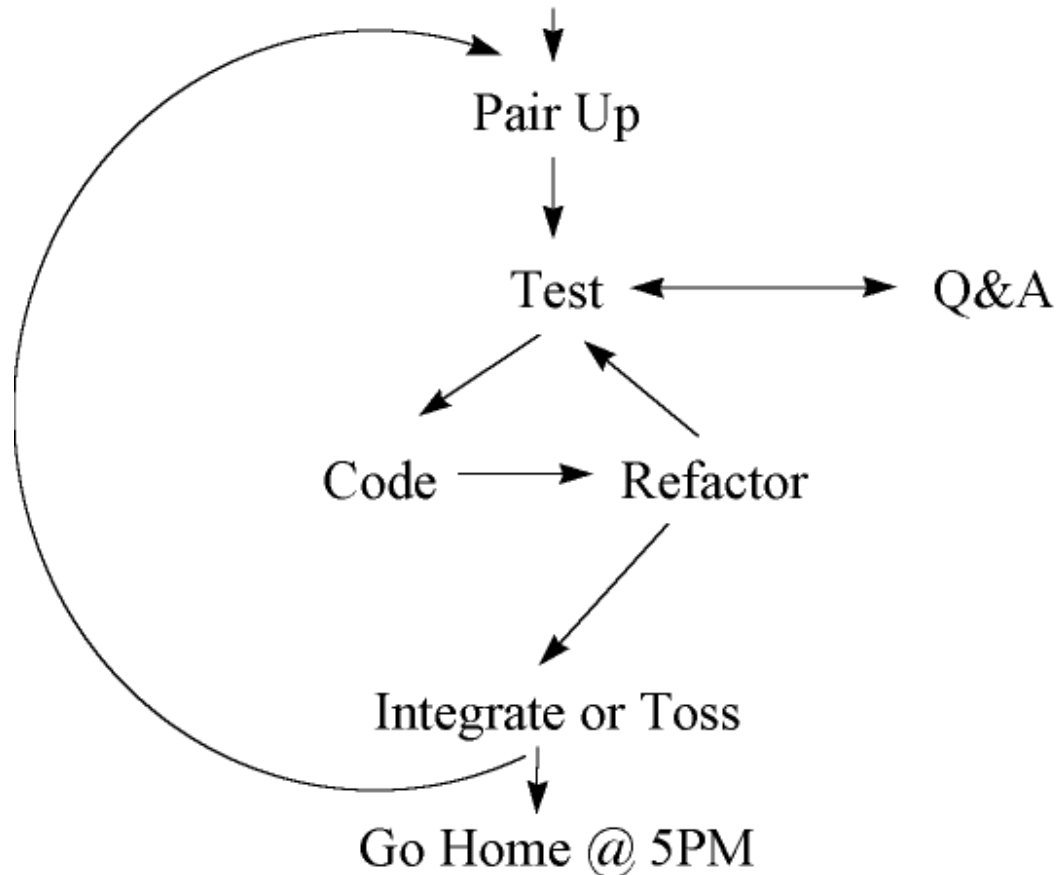
---

- Deixaram de ser mencionadas como práticas correntes no “novo XP”:
  - Adoção de padrões de codificação;
  - Metáfora.

# Desenvolvimento Ágil

## XP – O dia-a-dia de um programador XP

Standup Meeting @ 9AM





# Desenvolvimento Ágil

## XP – O dia-a-dia de um programador XP

---

- Reunião de início do dia
  - Não é prática oficial da XP;
  - Ajudam a definir o foco do dia;
  - Ajudam a lembrar da semana de 40h.
- Definição dos pares
  - Quem digita atua taticamente;
  - Quem acompanha atua estrategicamente;
  - Troca de papéis é feita periodicamente.
- Teste
  - Escreva testes para pequenas porções de código de cada vez;
  - Certificar-se de que testes falham antes da codificação;
  - Testar tudo o que pode dar errado.

Continua...



# Desenvolvimento Ágil

## XP – O dia-a-dia de um programador XP

---

- **Codificação**
  - Fazer o mais simples que pode funcionar;
  - Implementar apenas o suficiente para fazer o teste funcionar;
  - Siga os padrões de codificação.
- **Refatoração**
  - Melhore o código sempre que possível;
  - Elimine repetição de lógica;
  - Rode os testes.
- **Perguntas e Respostas**
  - Usuário disponível para respostas;
  - Decisões tomadas pelos usuários rapidamente.



# Desenvolvimento Ágil

## XP – O dia-a-dia de um programador XP

---

- Integrar ou jogar fora
  - Passar o código para a máquina de integração;
  - Testar e modificar até funcionar;
  - Se não funcionar, desfazer alterações e deixar para o dia seguinte.
- Redefinir o par
  - Se ainda há tempo, trocar de par (pelo menos) ou de papel.
- Ir para casa às 17:00h
  - Ajuda a reforçar a prática de 40h/s;
  - Não deixar nada “engatilhado” na integração.



# Desenvolvimento Ágil

## XP – Ciclo de desenvolvimento

---

- Ciclo de desenvolvimento compreende:
  - Definir o que o cliente quer;
  - Ter a certeza do que o cliente quer;
  - Estimar o esforço para construir a solução;
  - Construir a solução;
  - Implantar a solução.
- Ciclo é parecido com os de outros modelos de processos, porém bem mais rápido, envolvendo menos documentação.



# Desenvolvimento Ágil

## XP – Ciclo de desenvolvimento: Passos-chave

---

- Exploração:
  - Quando ocorre a visão do sistema;
  - Quando são escritas as primeiras *user stories* (em alto nível);
  - Possivelmente são feitas prototipações (*spikes*) para validar técnicas, etc;
  - Dura poucos dias/semanas;
  - Produz um plano de alto nível e uma estimativa de orçamento e prazo.



# Desenvolvimento Ágil

XP – Ciclo de desenvolvimento: Passos-chave

---

- Planejamento:
  - Fase curta onde clientes e desenvolvedores concordam com as funcionalidades a serem implementadas num primeiro *release* (essa escolha deve levar em consideração as necessidades de negócio e os aspectos técnicos).





# Desenvolvimento Ágil

## XP – Ciclo de desenvolvimento: Passos-chave

---

- Iterações para o *release*:
  - A liberação é transformada em iterações com durações típicas de 1 a 4 semanas;
  - A iteração inicial foca nos componentes da arquitetura que provêem a infra-estrutura para o sistema;
  - Cada liberação produz algo de valor para o usuário.
  - Usuários definem o que será produzido em cada iteração (escolhendo as histórias a serem implementadas).



# Desenvolvimento Ágil

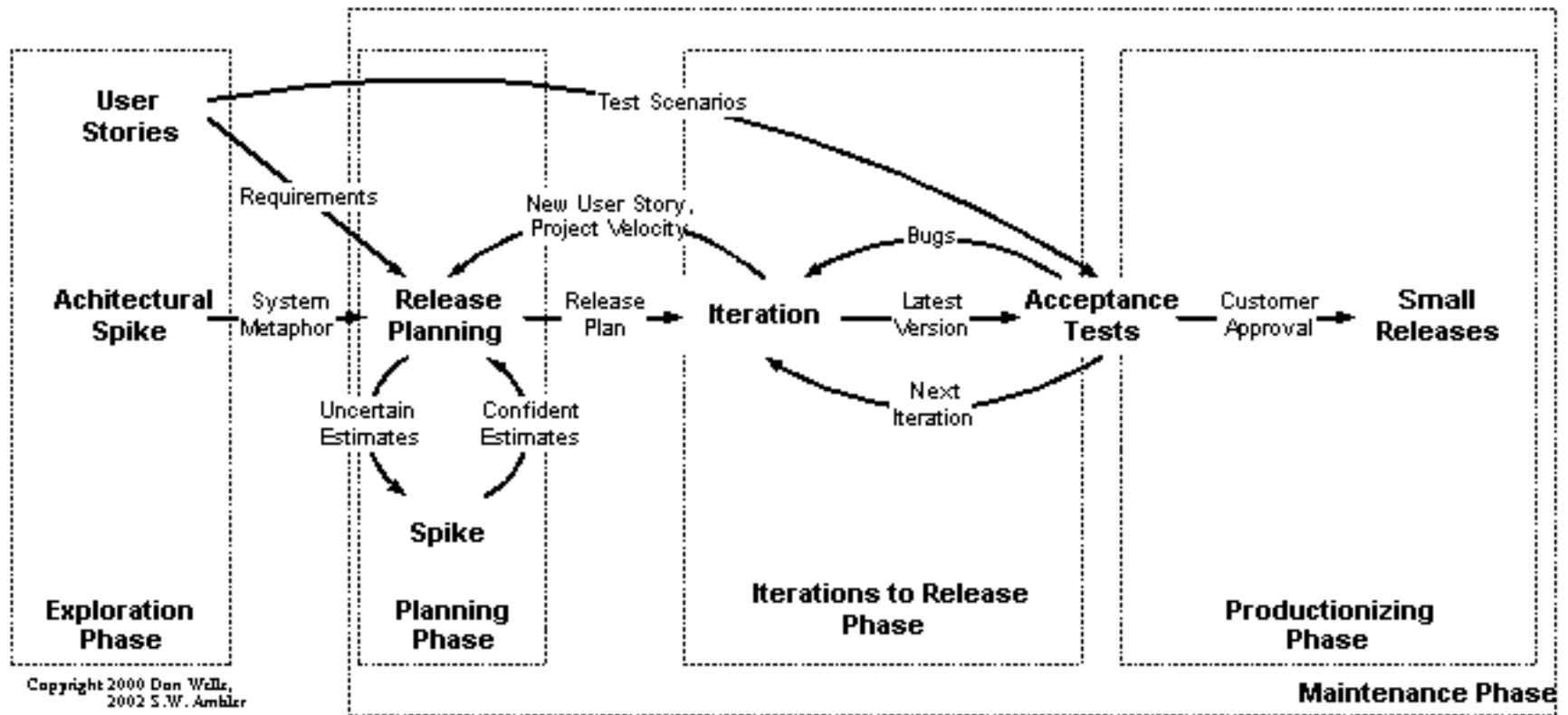
XP – Ciclo de desenvolvimento: Passos-chave

---

- “Producionalização” (*productionizing*):
  - Ao final da de cada iteração o produto é verificado e certificado para implantação;
  - Pode compreender ajustes finos;
  - Objetivo é estabilizar o sistema (não é manutenção evolutiva).

# Desenvolvimento Ágil

## XP – Ciclo de desenvolvimento





# Desenvolvimento Ágil

XP – *User stories*

---

**... E, PARA CONCLUIR**



# Desenvolvimento Ágil

## XP – *User stories*

---

- Uma história é uma pequena descrição de uma funcionalidade que o cliente pretende ver desenvolvida no sistema;
- Histórias compõem os requisitos. São atômicas, no sentido de que não é possível parti-las em conceitos mais pequenos;
- Histórias devem ser escritas pelo cliente ou por um seu representante; nunca pela equipe de desenvolvimento.



# Desenvolvimento Ágil

## XP – *User stories*

---

- Uma *user story* é constituída por:
  - Uma pequena descrição da necessidade, utilizada como lembrete e para as atividades de planejamento;
  - Conversações sobre a história, entre clientes e programadores, de modo a detalhar a história e esclarecer dúvidas;
  - Um conjunto de testes de aceitação.
- As histórias devem apresentar claramente algo de valor para o cliente;
- A história deve ser uma coisa simples, passível de se implementar em pouco tempo, tipicamente de 1 a 5 dias.



# Desenvolvimento Ágil

## XP – *User stories*: exemplo

---

- Registro de cliente na *Virtuloja* (necessário para colocar pedidos):
  - O cliente deve se registrar, indicando o *e-mail* como forma de acesso e escolhendo uma palavra-chave alfa-numérica.
  - O cliente pode fornecer os seguintes dados adicionais (não obrigatórios): endereço residencial, telefone e data de nascimento.
  - Em seguida deve receber um *e-mail* de confirmação do registro.
  - O cliente só fica devidamente registrado depois de responder ao *e-mail* de confirmação.



# Desenvolvimento Ágil

Lembrete

---

Próximo tópico: Scrum + XP