

# XP EXTREME PROGRAMMING

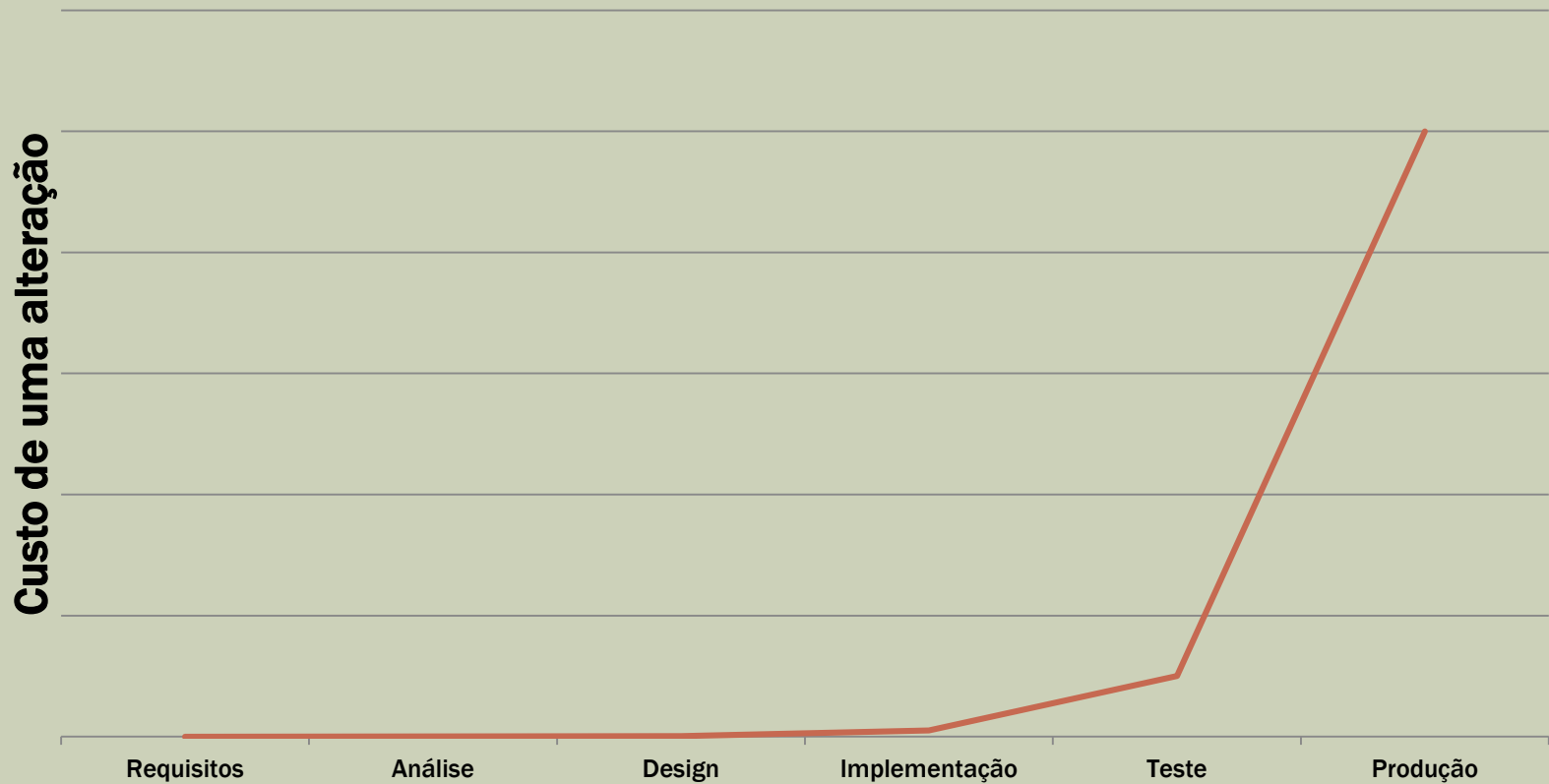
AG0106 - Gestão  
de Processos de  
Desenvolvimento  
de Software

# DESENVOLVIMENTO TRADICIONAL

- **Sequencial:** Análise, Design, Implementação, Teste, Implantação e Manutenção
- **Características:**
  - **Determinismo** – O final do processo de desenvolvimento deve sempre gerar o resultado conhecido, previsto
  - **Especialização** – Dividir o processo de desenvolvimento para que as tarefas sejam suficientemente simples e gerem resultados determinísticos
  - **Foco na execução** – O determinismo e a especialização permitiriam que os desenvolvedores se concentrassem apenas na execução. Bastaria seguir a especificação para chegar ao resultado desejado

# DESENVOLVIMENTO TRADICIONAL

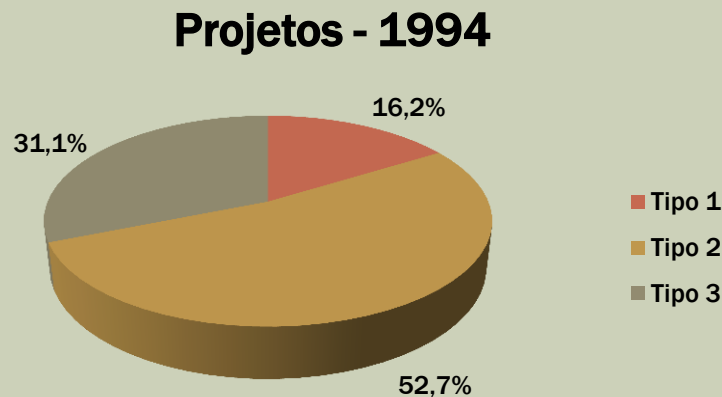
## ■ Custo de uma alteração



# DESENVOLVIMENTO TRADICIONAL

## ■ Tipos de projetos:

- **Tipo 1** (bem-sucedido) – Finalizado no prazo, dentro do orçamento e contendo todas as funcionalidades especificadas
- **Tipo 2** (desafiado) – Finalizado e entrega um software operacional, mas ultrapassa o orçamento e o prazo, além de disponibilizar menos funcionalidades que o especificado
- **Tipo 3** (fracassado) – Cancelado em algum momento durante o ciclo de desenvolvimento



# DESENVOLVIMENTO TRADICIONAL

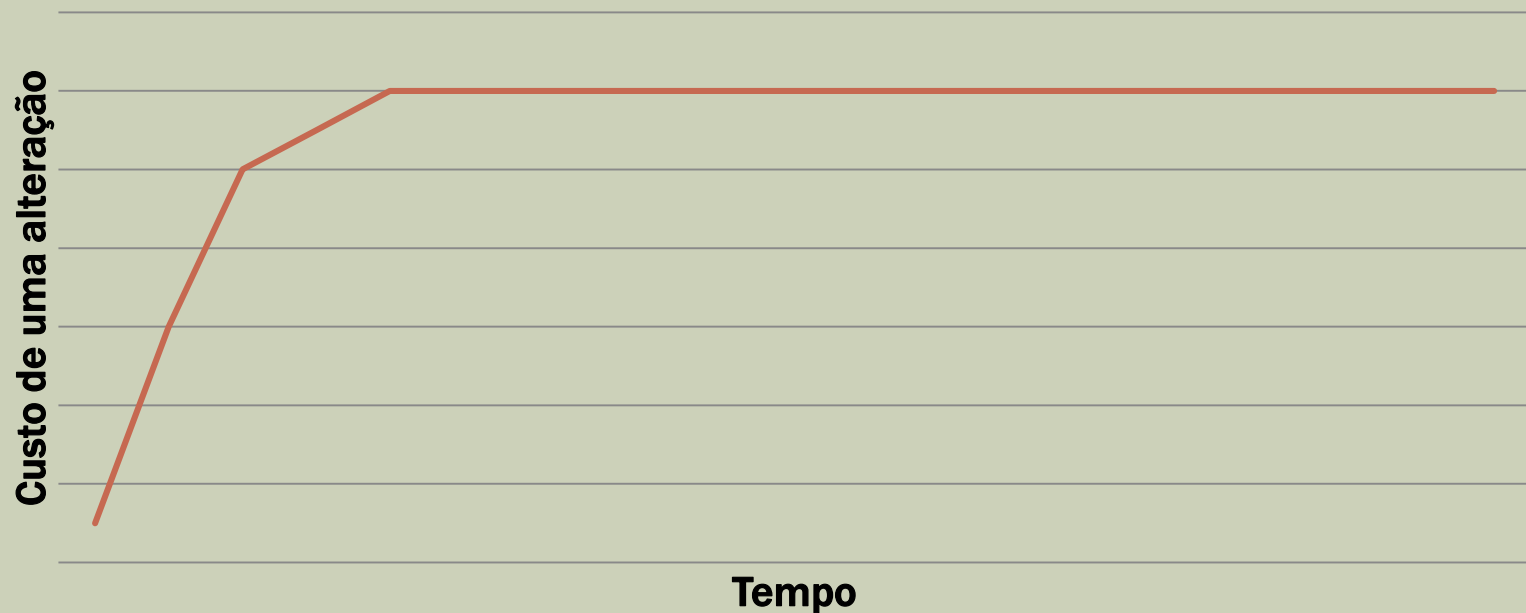
- **Trabalhador manual** – Exerce atividades que dependem basicamente de suas habilidades manuais e não se baseiam no uso intensivo de conhecimento
  - Fácil de automatizar
  - Determinístico
  - Repetitivo
- **Trabalhador do conhecimento** – Produz com base no uso intensivo do conhecimento e criatividade
  - Cometer erros é uma parte natural e saudável do trabalho
- **O desenvolvimento de software necessita de trabalhadores do conhecimento, entretanto as premissas do desenvolvimento tradicional somente são válidas para trabalhadores manuais**

# MANIFESTO ÁGIL

- **Indivíduos e interação entre eles** mais que processos e ferramentas
- **Software em funcionamento** mais que documentação abrangente
- **Colaboração com o cliente** mais que negociação de contratos
- **Responder a mudanças** mais que seguir um plano

# DESENVOLVIMENTO ÁGIL

- Custo de uma alteração – Como o processo é iterativo, o custo para uma alteração se mantém constante



# XP

- Kent Beck – 1996

- Valores:

- **Feedback** – O cliente aprende com o software que utiliza e fornece feedback para a equipe de desenvolvimento, direcionando o desenvolvimento e garantindo que a atenção seja focada naquilo que gera mais valor
- **Comunicação** - Todos os envolvidos no projeto devem poder se comunicar face a face ou da forma mais rica possível. A comunicação com o cliente permite que todos os detalhes do produto sejam tratados com a atenção e a agilidade que merecem
- **Simplicidade** – Apenas aquilo que é suficiente para atender a cada necessidade do cliente deve ser implementado. Evitar futurologia!
- **Coragem** – A equipe deve ser corajosa e acreditar que, utilizando as práticas e valores do XP, será capaz de fazer o software evoluir com segurança e agilidade



# XP - PRÁTICAS



# XP - PRÁTICAS

- **Cliente presente** – É essencial que o cliente participe ativamente do processo de desenvolvimento. Ligada aos valores: feedback, comunicação e simplicidade
- **Jogo do planejamento** – Ocorre no início de cada release e iteração. É uma reunião onde o cliente avalia e prioriza as funcionalidades a serem desenvolvidas. Funcionalidades são descritas em pequenos cartões e denominadas **estórias**. As estórias são estimadas usando **pontos**
  - **Release** – Período longo, onde se desenvolve um módulo do sistema, que gera um valor bem definido para o cliente. Um release pode incluir várias iterações
  - **Iteração** – Período de poucas semanas, onde se desenvolve um conjunto de funcionalidades
  - **Ponto** – Uma unidade de esforço, com significado definido no projeto

# XP - PRÁTICAS

- **Stand up meeting** – Uma reunião, onde cada membro da equipe apresenta o que foi realizado, os próximos passos e os impedimentos encontrados. É rápida (alguns minutos) e todos os membros permanecem em pé
- **Programação em par** – Diante de cada computador existem sempre dois desenvolvedores que trabalham juntos para produzir o mesmo código. Permite que o código seja revisando enquanto é construído. Os desenvolvedores se complementam e têm mais oportunidades de gerar soluções inovadoras

# XP - PRÁTICAS

- **Desenvolvimento guiado pelos testes (TDD)** – Os desenvolvedores escrevem testes para cada funcionalidade antes de codificá-las. Fazendo isso, os desenvolvedores:
  - Aprofundam o entendimento das necessidades do sistema (o que aprimora a análise)
  - Se preocupam com as interfaces externas dos métodos e classes antes de codificá-los (o que melhora o design)
  - Sabem até onde codificar cada funcionalidade (o que ajuda a manter o sistema simples)
  - Passam a contar com uma massa de testes que pode ser usada a qualquer momento para validar todo o sistema
- **Refactoring** – É o ato de alterar um código sem afetar a funcionalidade que ele implementa. É usado para tornar o software mais simples de ser manipulado

# XP - PRÁTICAS

- **Código coletivo** – Os desenvolvedores têm acesso a todas as partes do código e podem alterar aquilo que julgarem importantes, sem a necessidade de pedir autorização de outra pessoa
- **Código padronizado** – A equipe deve criar padrões de codificação para tornar o sistema mais homogêneo e permitir que manutenções futuras ocorram de forma mais rápida
- **Design simples** – Em vez de criar generalizações dentro do código, de modo a prepará-lo para possíveis necessidades futuras, a equipe deve sempre optar por um código que seja suficiente para atender às necessidades da funcionalidade que está implementando
  - Os desenvolvedores se baseiam na premissa de que serão capazes de incorporar qualquer necessidade futura quando e se ela surgir

# XP - PRÁTICAS

- **Metáfora** – Tem o poder de transmitir ideias complexas de forma simples, utilizando uma linguagem comum que é estabelecida entre a equipe de desenvolvimento e o cliente
- **Ritmo sustentável** – Para garantir que a equipe tenha sempre o máximo de rendimento e produza software com melhor qualidade possível, o XP recomenda que os desenvolvedores trabalhem apenas oito horas por dia e evitem fazer horas extras, visto que é essencial estar descansado a cada manhã, de modo a utilizar a mente na sua plenitude ao longo do dia
- **Integração contínua** – Uma nova funcionalidade pode afetar outras já implementadas. Os pares devem integrar seus códigos com o restante do sistema várias vezes ao dia, executando todos os testes, para assegurar que a integração ocorreu de forma satisfatória

# XP - PRÁTICAS

- **Releases curtos** – O XP tem como objetivo gerar um fluxo contínuo de valor para o cliente e, por isso, trabalha com releases curtos. A equipe produz um conjunto reduzido de funcionalidades e o coloca em produção rapidamente, de modo que o cliente possa utilizar o software no dia a dia e se beneficiar dele

# XP - EQUIPE

## ■ Papéis:

- **Gerente de projeto** – Responsável por assuntos administrativos e relacionamento com o cliente
- **Coach** – Responsável técnico do projeto. Assegura o bom funcionamento do processo e busca formas de melhorá-lo continuamente
- **Analista de teste** – Auxilia o cliente a escrever os testes de aceitação. Garante a execução contínua dos testes
- **Redator técnico** – Auxilia a equipe a documentar o sistema. Permite que os desenvolvedores se concentrem prioritariamente na implementação do software
- **Desenvolvedor** – Analisa, projeta e codifica o sistema. Em XP, não existem divisões entre analista, projetista, programador, etc



# REFERÊNCIAS

- Teles, Vinícius Manhães. **“Extreme Programming ”**. Segunda edição. Novatec, 2014



**DÚVIDAS?**