



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

Introdução ao Processo de Desenvolvimento de Software

Ygor Amaral <ygor.amaral@ufrpe.br>

Disciplina: Processo de Desenvolvimento de Software

Curso: Sistemas de Informação (2016.1)

Pergunta...



-
- O que é um projeto?

Pergunta...



- O que é um projeto?
 - Esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo
 - A sua natureza temporária indica um início e um término definidos

Pergunta...

- O que é um gerenciamento de projetos?

Pergunta...

- O que é um gerenciamento de projetos?
 - Aplicação de:
 - Conhecimentos
 - Ferramentas
 - Técnicas...

...as atividades do projeto a fim de atender aos seus requisitos

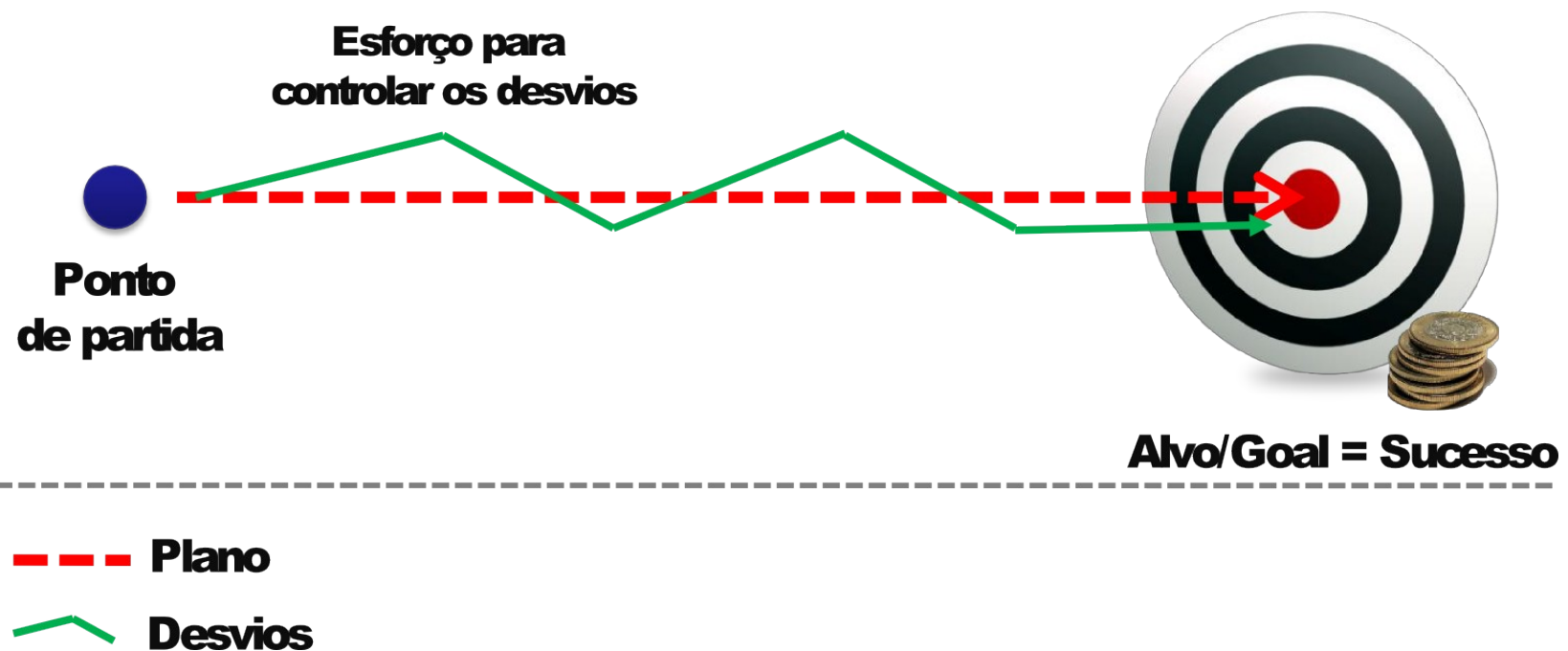
Modelos clássicos de desenvolvimento de software



- Rigor no planejamento
 - Busca-se previsibilidade e controle sobre as entregas a serem realizadas ao fim do projeto
- Busca-se a execução através de um processo prescritivo
 - Bastante planejado, detalhado....
 - A principal função desses modelos é colocar em ordem o caos do desenvolvimento de software

Determinando o objetivo

- Fixa-se o plano e todo esforço busca controlar os possíveis desvios...



Porém...



-
- O alvo é fixo?

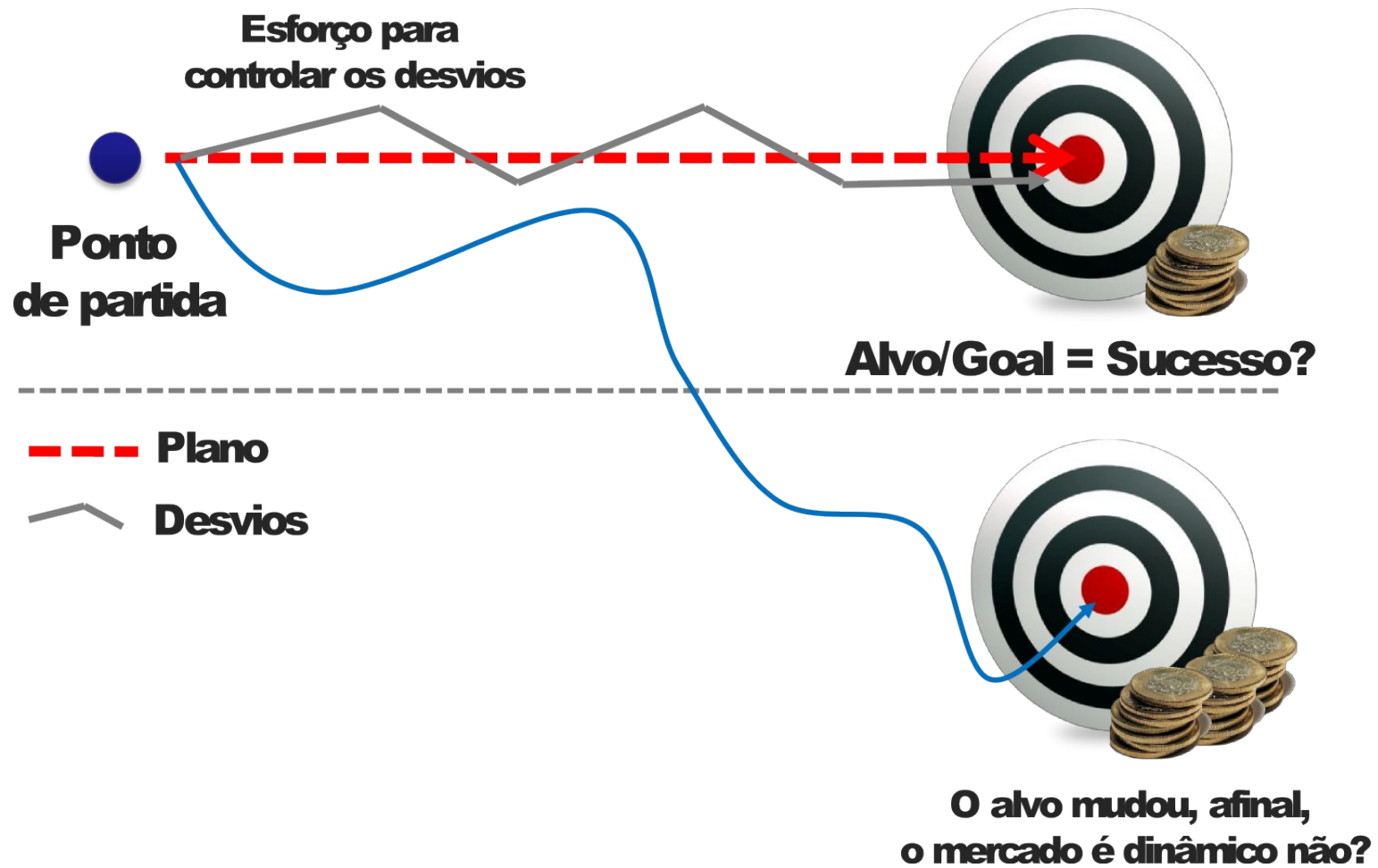
Porém...

- O alvo é fixo?
 - A indústria de software opera sob “constante mudança”
 - Ambientes com alto grau de inovação!!!!
 - Os processos tradicionais são frequentemente burocráticos e não satisfazem todos os tipos de desenvolvimento de software
 - Surge a necessidade de abordagens que possam facilmente responder as mudanças!!

Porém...

- O alvo é fixo?
 - **Natureza dos projetos de software**
 - Software é intangível
 - Descoberta
 - Investigação
 - Evolução dos requisitos ao longo do desenvolvimento do produto
 - Mudança
 - **Projeto dinâmicos acabam demandando uma abordagem empírica**
 - Projetar, desenvolver e gerenciar simultaneamente

O alvo é fixo?



O que define sucesso em seus projetos?



- É apenas entregar no prazo e dentro do orçamento previsto?

O que define sucesso em seus projetos?

- É apenas entregar no prazo e dentro do orçamento previsto?
- E quanto a satisfação do cliente?

Agilidade (UOL Michaelis)



- Qualidade do que é ágil
- Desembaraço, ligeireza, presteza de movimentos
- Mobilidade, perspicácia, vivacidade

Em nosso contexto, o que é Agilidade?



- Habilidade de criar e responder a mudanças de forma a manter a lucratividade em um turbulento ambiente de negócios
 - O desenvolvimento de software é sim um ambiente turbulento devido a sua dinâmica...

Desenvolvimento de software tradicional

- Mesmo com todos os esforços em tornar o processo de desenvolvimento de software mais efetivo...
 - A prática apresentava resultados ruins

Desenvolvimento de software tradicional

- Excesso de documentação
- Lentidão de resposta as mudanças
- Distanciamento entre o que o cliente quer (ou precisa) em relação ao que é entregue



Manifesto para o desenvolvimento ágil de software



- Criadores do manifesto:
 - Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo

Manifesto para o desenvolvimento ágil de software



- Através deste trabalho, passamos a valorizar:
 - 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
- Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

12 princípios do manifesto



- 1) Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor
- 2) Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento
 - Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas

12 princípios do manifesto



-
- 3) Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos
 - 4) Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto

12 princípios do manifesto



5) Construir projetos ao redor de indivíduos motivados

- Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho

6) O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa “cara a cara”

12 princípios do manifesto



- 7) Software funcional é a medida primária de progresso
- 8) Processos ágeis promovem um ambiente sustentável
- 9) Atenção contínua à excelência técnica e bom design (projeto), aumenta a agilidade

12 princípios do manifesto



10) Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito

11) As melhores arquiteturas, requisitos e designs (projetos) emergem de times auto-organizáveis

12) Em intervalos regulares, a equipe reflete sobre como podem ser mais eficazes, então sintonizam e ajustam seus comportamentos desta maneira

Adoção dos Métodos Ágeis pelo mercado



- Diversas empresas tem adotado alguma metodologia ágil nos seus projetos
- O mais comum, na verdade, é pegar algumas técnicas do processo tradicional e mesclar com métodos ágeis

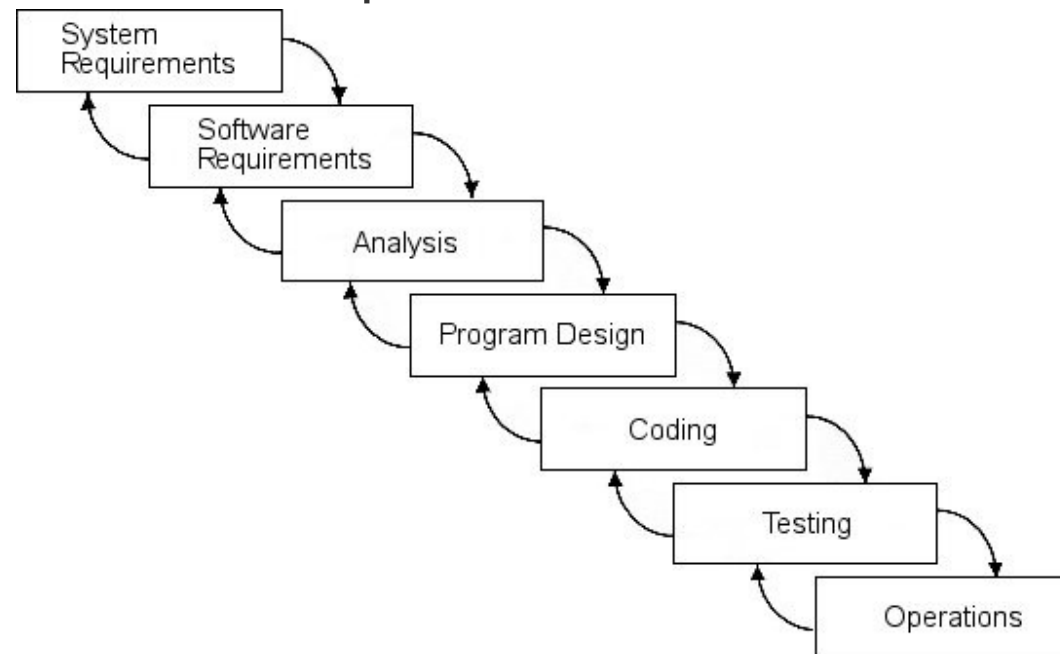
Gerenciamento Ágil de Projetos



- Diante o advento das metodologias ágeis de desenvolvimento de software...
 - Mundo da engenharia de software
- ...o Gerenciamento de Projetos é repensado para esta nova abordagem
 - Mundo da gestão
- Surge assim o chamado “Gerenciamento Ágil de Projetos”
 - Mudança de um gerenciamento prescritivo para adaptativo...

Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - Diversas etapas no projeto
 - Cada etapa leva muito tempo (meses)
 - Levantamento “muito” bem detalhado
 - Contrato de escopo fechado



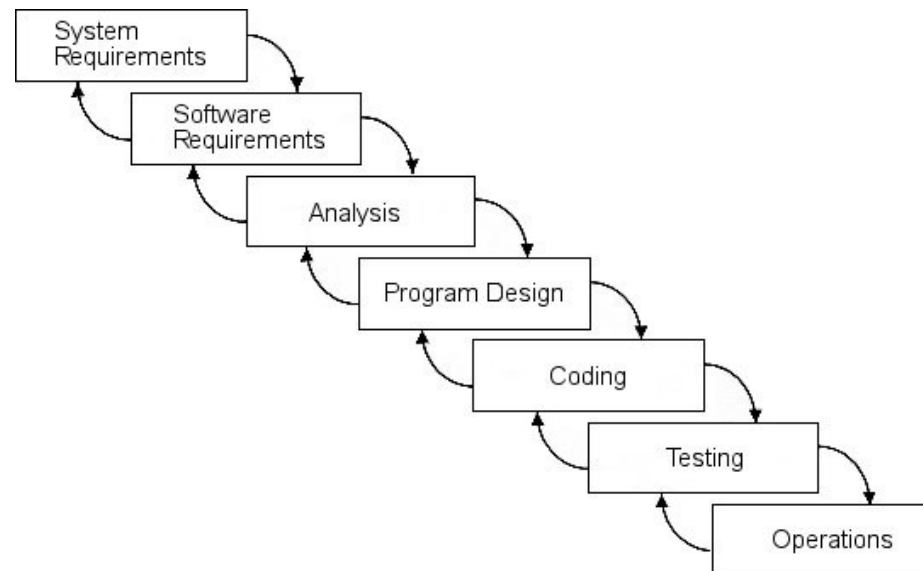
Problemas com métodos tradicionais



- Modelo waterfall (cascata)
 - O cliente era forçado a decidir tudo o que ele “precisa” em um momento muito inicial do projeto
 - Momento em que ele ainda sabe muito pouco sobre o que realmente ele quer
 - “O cliente não sabe o que quer”

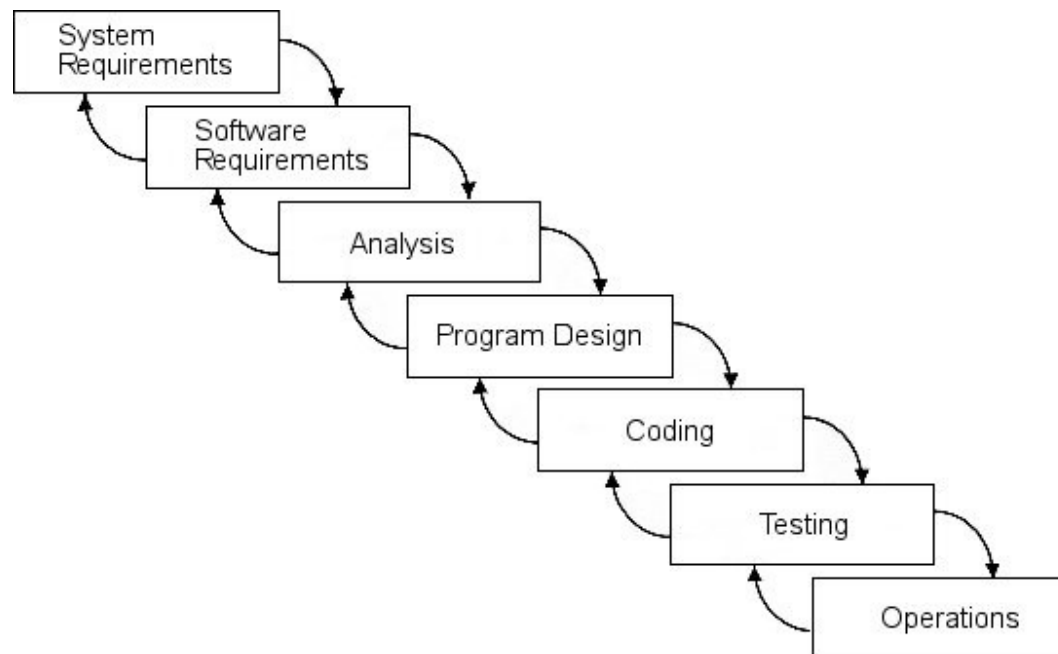
Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - Equipes diferentes faziam etapas diferentes
 - Testes eram realizados apenas no final
 - Muitas vezes os prazos de entrega já estavam apertados por atrasos no desenvolvimento, o que levava a testar muito pouco!



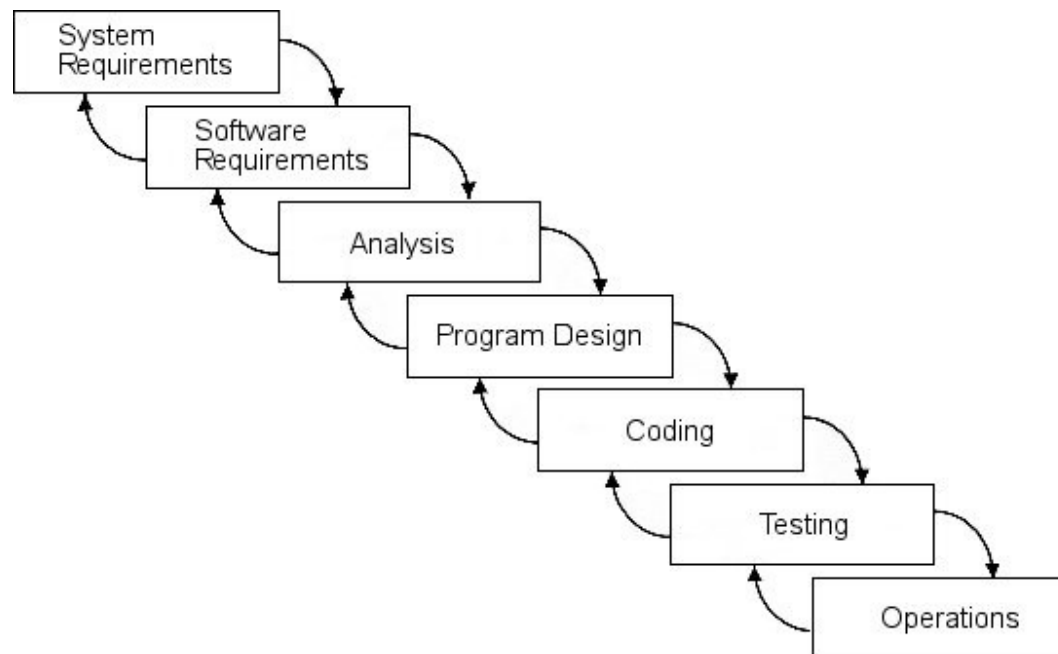
Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - Somente quando uma etapa era completada que se passava para a próxima!
 - Equipes poderiam ficar ociosas, esperando o início de uma nova etapa



Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - O grande problema é que o software só pode ser “visualizado pelo cliente” próximo ao fim do prazo
 - Devido a essa linearidade de etapas
 - “Não era isso que eu queria”



Problemas com métodos tradicionais



- Modelo waterfall (cascata)
 - No início do projeto, sabe-se muito pouco
 - Poucos experimentos e descobertas realizadas
 - Sabe-se pouco sobre o contexto do problema
 - É justamente nesse momento que se toma as “piores” decisões de planejamento do projeto
 - Várias decisões importantes são tomadas nesse momento inicial do projeto

Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - Levantar todos os requisitos de antemão!
 - Tentar fazer todo o *design* de antemão!
 - *Design* != desenho

Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - Tudo isso levava a requisitos pouco claros!
 - Tudo era identificado de antemão...
 - Medo de ir para a próxima fase!
 - Será que planejamos tudo de forma bem feita?
 - Será que vai dar certo?!

Problemas com métodos tradicionais



- Modelo waterfall (cascata)
 - Resistência a mudanças
 - Clientes não sabem o que querem
 - Decisões importantes (e erradas) feitas no início do projeto, geram grandes problemas
 - Mudanças se tornam mais e mais caras
 - O que foi decidido no início deveria que ser mantido até o final

Problemas com métodos tradicionais

- Modelo waterfall (cascata)
 - Desperdício
 - 52% dos requisitos planejados, eram implementados
 - 64% dos requisitos implementados, eram raramente utilizados

Fonte: Standish Report

Quais os
principais
problemas em
projetos?

Principais problemas em projetos...



- Falta de planejamento
- Mudança constante de requisitos
- Escopo mal definido
- Falta de participação do cliente
- Comunicação falha

Tradicional x Ágil

- O Desenvolvimento Ágil de Projetos é uma nova **abordagem** bastante utilizada
- Porque preciso de uma nova abordagem?
 - **Projetos** diferentes **precisam de** métodos diferentes

Tradicional x Ágil

- Práticas ágeis NÃO são apropriadas para todos os cenários
- Em especial, a agilidade é melhor aproveitada nas seguintes situações:
 - Projetos cujo esforço é intelectual;
 - Escopo altamente sujeito a mudanças;
 - Restrições agressivas de tempo.

Industrial x Intellectual

- O trabalho intelectual tem características diferentes do trabalho industrial:

Características do trabalho industrial	Características do trabalho intelectual
O trabalho é visível	O trabalho é invisível
O trabalho é estável	O trabalho muda constantemente
Foco em operação e manutenção	Foco em mudança e inovação
Comando e controle	Autonomia
Normas rígidas	Inovação contínua
Minimizar o custo de trabalhadores em tarefas	Trata os trabalhadores como ativos, não custos



UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

UNIDADE
ACADÊMICA DE
SERRA TALHADA

Entrega orientada por Valor

Value-driven delivery

Entrega orientada por Valor



- É a base da “agilidade de software”
- Esse conceito rege todo o trabalho do projeto do ponto de vista de planejamento e priorização
 - É o valor de uma determinada parte do escopo para o negócio

Entrega orientada por Valor



- Nosso principal objetivo é fazer com que este valor seja entregue o mais rápido possível
- Está intimamente ligado com o valor do manifesto ágil “Software em funcionamento mais do que documentação abrangente”

Perdas

- São consideradas perdas, qualquer tipo de esforço desnecessário
 - Que não agregue valor ao negócio da organização
- Empresas ágeis tomam muito cuidado para não desperdiçar o seu tempo

Planejamento adaptativo

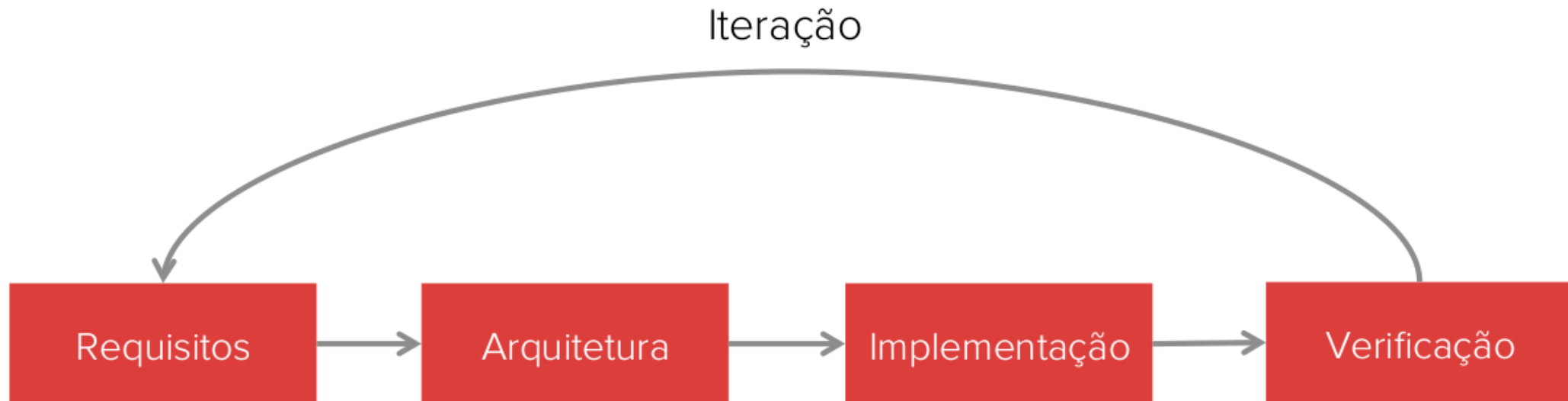
- Como o próprio nome já diz...
 - Além de realizado constantemente durante o projeto – é moldado para o cenário atual
- É aceitar que o plano inicial irá mudar, e se prepare para isso
- É a aplicação de lições aprendidas do projeto, no próprio projeto em andamento

Planejamento adaptativo



- É a diminuição da incerteza e falta de previsibilidade de projetos que são sujeitos a mudanças
- É a garantia de que o projeto continua “resolvendo o problema” proposto no início
 - Ou seja, que valores continuam sendo entregues
 - O planejamento não se torna ultrapassado ou obsoleto (por ser adaptativo)

Ciclo de vida iterativo e incremental



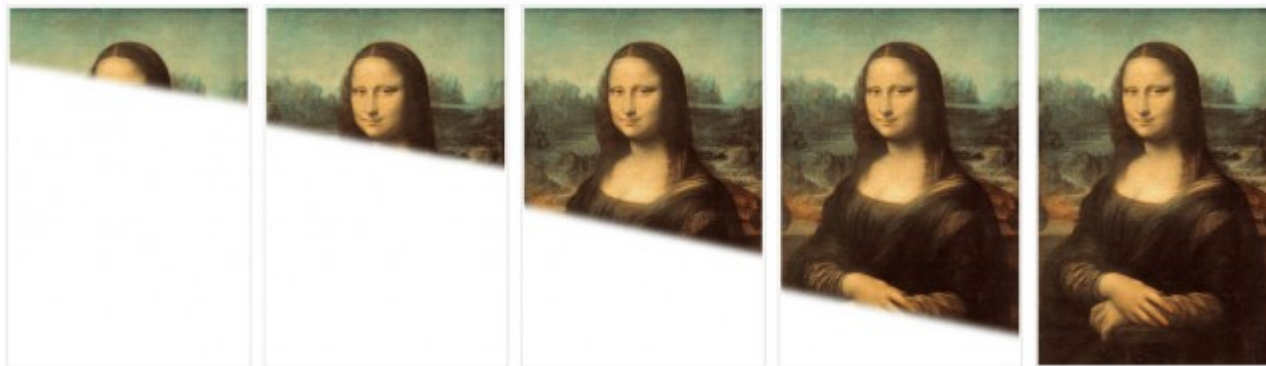
Ciclo de vida iterativo e incremental

WATERFALL



Garante que uma fase não comece antes que a anterior termine

AGILE



Desenvolvimento iterativo incremental, começando pelos requisitos de alta prioridade

Ciclo de vida iterativo e incremental

Traditional Project Management Lifecycle

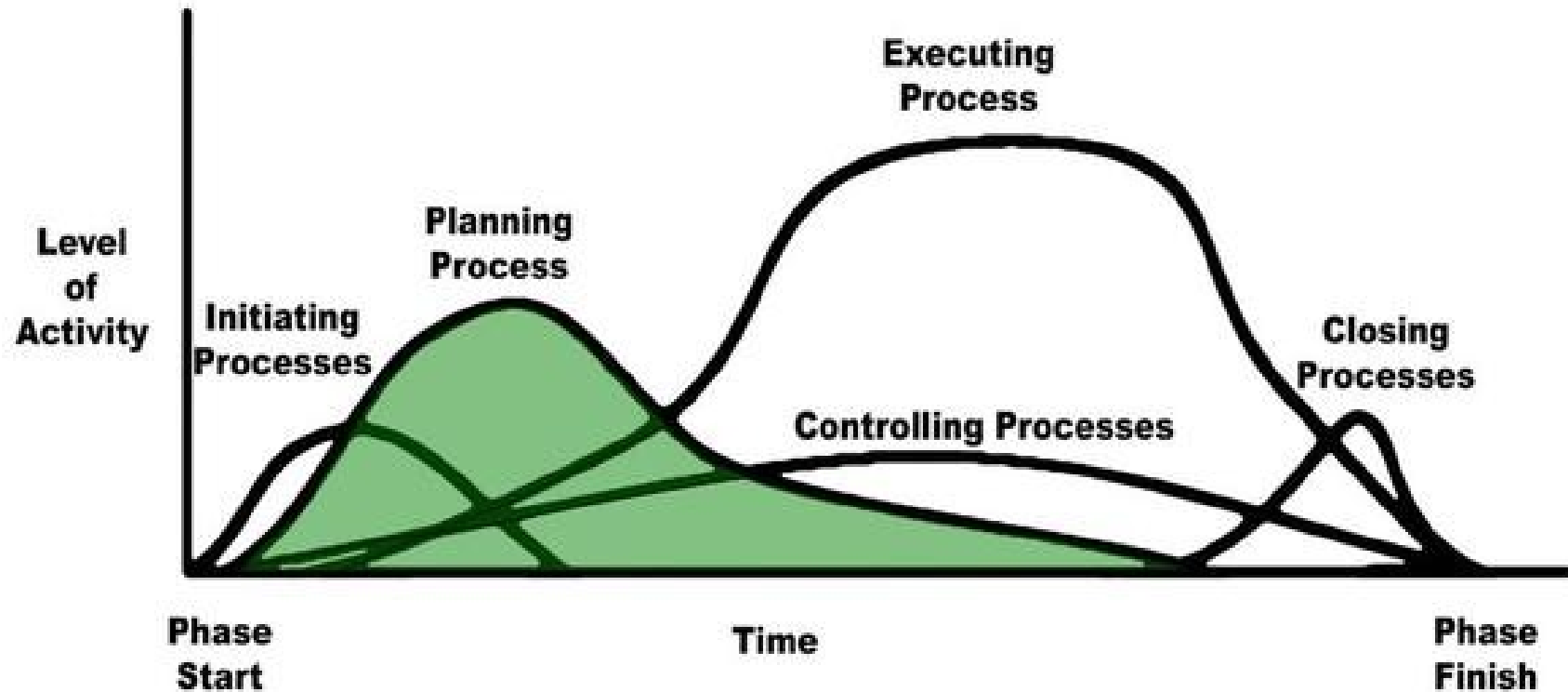


Figure 1 – Traditional Planning Focus
(Adapted the from PMBOK Guide)

Ciclo de vida iterativo e incremental

Agile Project Management Lifecycle

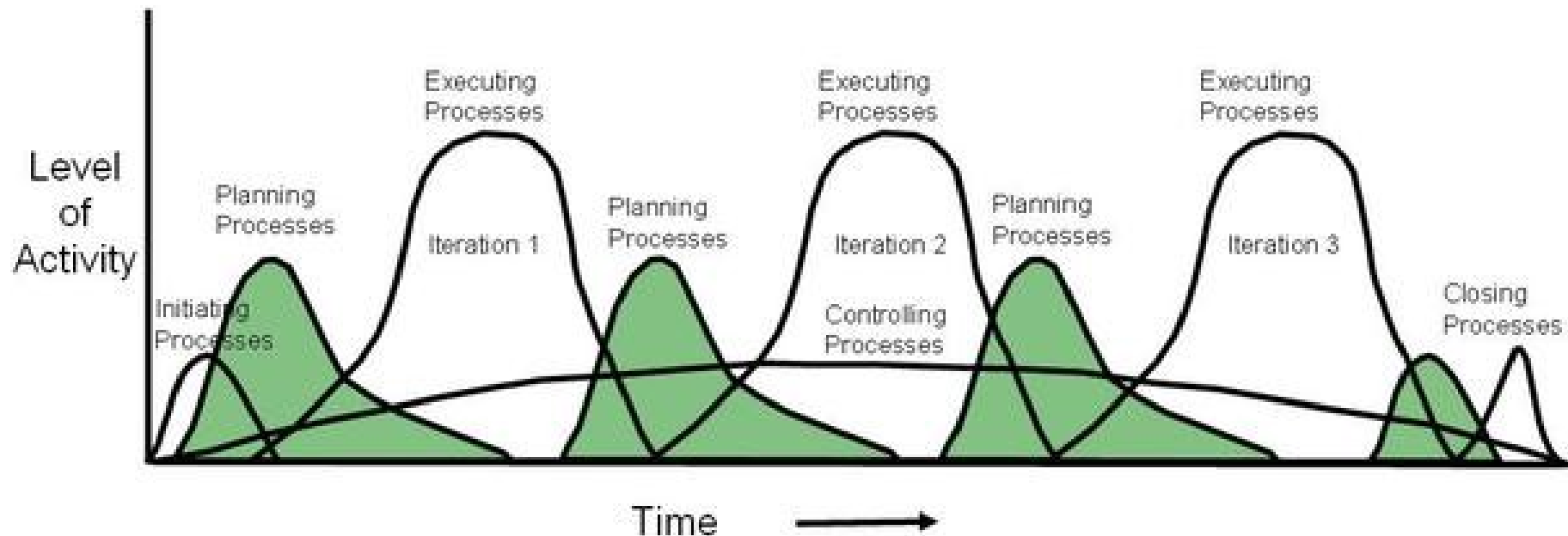


Figure 2 – Agile Planning Focus

Timeboxing

- Fazer o melhor trabalho possível dentro do tempo disponível
 - Caso o desenvolvimento leve mais tempo que o disponível, é preferível reduzir o escopo do que adicionar recursos
- Serve para tornar os resultados de uma iteração visíveis para os envolvidos
 - Avaliar a iteração... *feedback*...

Timeboxing

- Se um trabalho planejado fica de fora por falta de tempo, ele é incluído no próximo timebox



Ferramentas ágeis

- Existem diversos métodos ágeis
 - Não existe uma única maneira de “se fazer ágil”
 - As principais abordagens:
 - Scrum
 - XP
 - Kanban
 - A essência é a mesma, o que muda é a forma

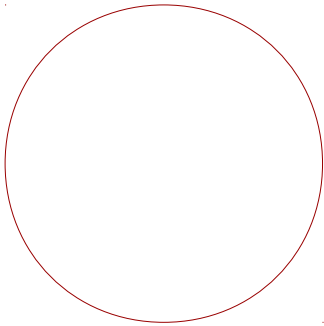
Veja os métodos ágeis como ferramentas!

- Discussões:
 - Scrum é melhor que XP...
 - XP é melhor que Scrum...
 - Kanban é o melhor...
- Na realidade são métodos diferentes!
 - São ferramentas diferentes
 - Você usa uma determinada ferramenta dependendo da sua necessidade

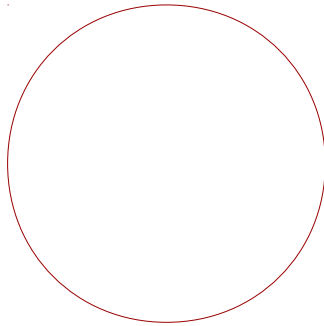


Prescritivo x Adaptativo

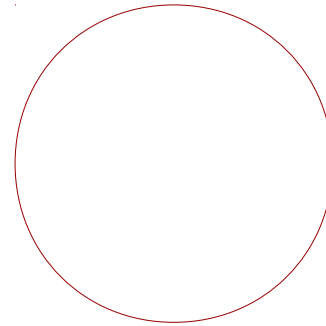
< Prescritivo



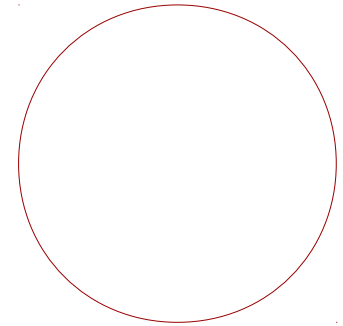
RUP



XP



Scrum



Kanban

Adaptativo >

Prescritivo x Adaptativo

- Quanto mais adaptativo for o processo das empresas, menos ela se parecerão
 - Cada empresa irá trabalhar de uma forma (adaptada)...
 - Vão adaptar bastante o seu dia a dia...
- Empresas que usam RUP (prescritivo) vão ter muitas coisas em comum no seu processo

Prescritivo x Adaptativo



- Quanto mais adaptativo o método for, mais ele pode ser aderente a realidade do seu contexto
 - **Porém mais difícil!**
 - Você vai ter que encontrar as suas próprias formas de trabalhar...

Qualquer ferramenta pode ser mal utilizada...

- Muitas vezes, o insucesso do projeto não é mérito da ferramenta utilizada
 - “Você pode usar uma serra elétrica como um machado para cortar uma árvore...”
 - Se você utilizar uma serra elétrica da forma correta, irá obter resultados melhores





**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

Scrum

Método... ferramenta...

Scrum

- É um processo ágil que permite manter o foco na entrega do maior valor de negócio, no menor tempo possível
- Isto permite a rápida e continua inspeção do software em produção (em intervalos de duas a quatro semanas)
 - Na verdade se discute intervalos de apenas 1 semana também!

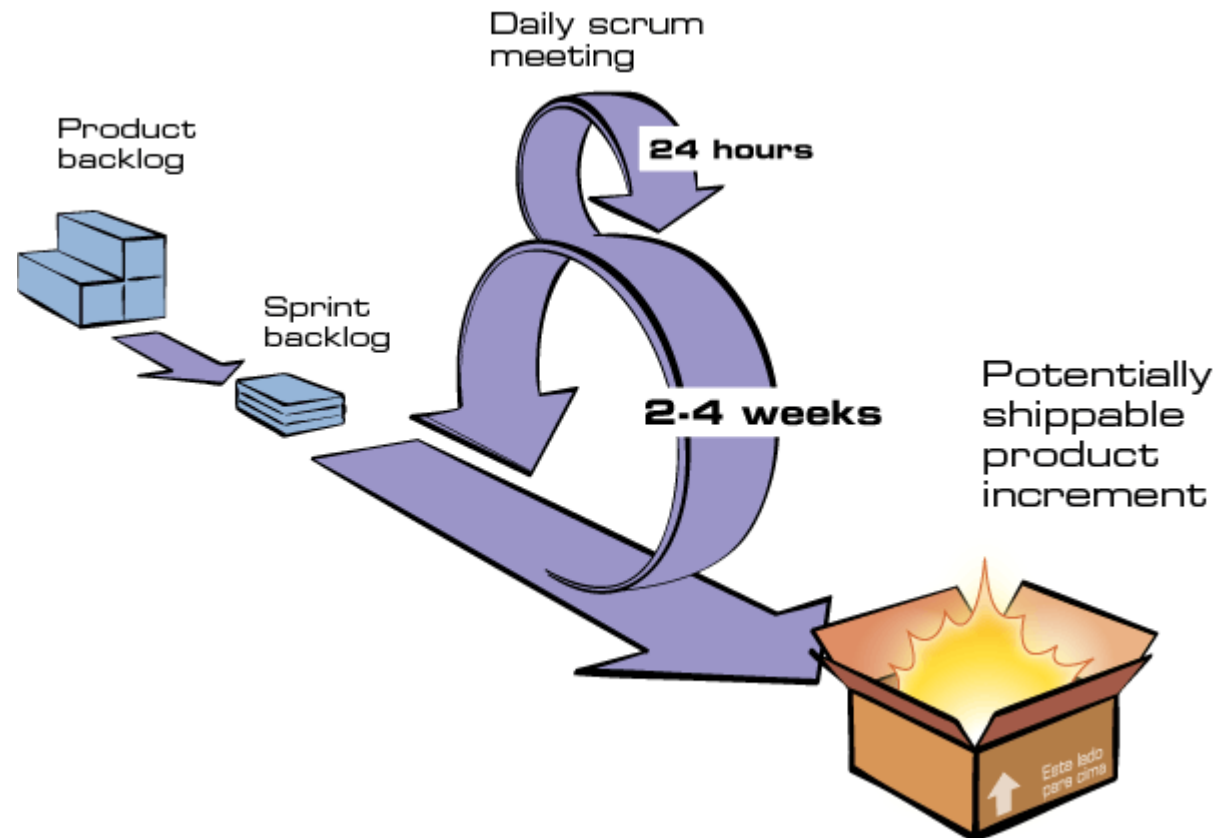
Scrum

- As necessidades do negócio é que determinam as prioridades do desenvolvimento de um sistema
 - As equipes se auto-organizam para definir a melhor maneira de entregar as funcionalidades de maior prioridade

Scrum

- A cada intervalo todos podem ver o real software em produção
 - Decidindo se o mesmo deve ser liberado (validação) ou continuar a ser aprimorado por mais uma "sprint"

Ciclo do Scrum



Sprints

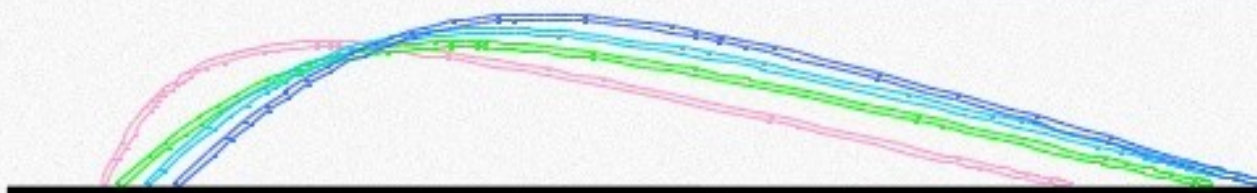
- Projetos Scrum progridem em uma série de “sprints”
- Normalmente ocorrem em intervalos de duas a quatro semanas
 - As vezes 1 semana também!
- O produto é projetado, codificado e testado durante a sprint

Desenvolvimento sequencial versus paralelo



Ao invés de completar uma coisa por vez...

... equipes Scrum fazem um pouco de cada coisa, todo o tempo.



Fonte: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

Scrum framework

Papéis

- Dono do produto
- ScrumMaster
- Equipe

Cerimônia

- Planejamento
- Revisão
- Retrospectiva
- Reunião diária

Artefatos

- Product backlog
- Sprint backlog
- Burndown charts

Papéis

- Product Owner – PO (dono do produto)
 - Define as funcionalidades do produto
 - Decide datas de lançamento
 - Responsável pela rentabilidade
 - Prioriza as funcionalidades de acordo com o valor de mercado
 - Aceita ou rejeita os resultados das sprints

Papéis

- Scrum Master
 - Responsável pela aplicação de valores e práticas do Scrum
 - Remove/diminui obstáculos
 - Garante a plena funcionalidade e produtividade da equipe
 - Garante a colaboração entre os diversos papéis e funções
 - Escudo para interferências externas

Papéis

- Time (Equipe)
 - Além do PO e ScrumMaster...
 - Todos os demais membros do time...
 - Desenvolvedor, *tester*, *design*...

Reuniões SCRUM (cerimônias)

- Reunião diária (*Daily Scrum Meeting*)
- Reunião de Planejamento da Sprint (*Sprint Planning Meeting*)
- Reunião de Revisão da Sprint (Sprint Review)
- Retrospectiva da Sprint (Sprint Retrospective)

Reunião diária

- Como o próprio nome diz... é diário!
 - 15 minutos apenas...
- Todos precisam ficar em pé...
 - A reunião deve acontecer no mesmo local e mesma hora todos os dias
- Não é para solução de problemas
- Ajuda a evitar reuniões adicionais desnecessárias



Reunião diária

- Durante a reunião, cada membro da equipe responde a três perguntas:
 - O que você tem feito desde ontem?
 - O que você está planejando fazer hoje?
 - Você tem algum problema impedindo você de realizar seu objetivo?
- É papel do Scrum Master para facilitar a resolução desses impedimentos
 - Fora do contexto da reunião para que ela possa durar menos de 15 minutos

Planejamento da Sprint



- No início do ciclo de sprint (a cada 7-30 dias), uma *Sprint Planning* é realizada
 - Selecionar o trabalho que precisa ser feito
- Dividida em duas partes:
 - Time com o Product Owner: diálogo para priorizar o Product Backlog
 - Apenas o time: detalhamento das atividades, resultando na Sprint Backlog
 - Tarefas identificadas e estimadas
 - De forma colaborativa, não apenas o Scrum Master

Reunião de Revisão da Sprint (*Review*)



- Time apresenta os resultados obtidos durante a Sprint
- Tipicamente, demonstração de novas funcionalidades ou sua arquitetura
- Informal
- Todo o time participa

Reunião de Revisão da Sprint (*Review*)

- O objetivo desta reunião é apresentar ao Product Owner todos os entregáveis que foram produzidos pelo time
 - Se uma funcionalidade não foi completada, será recusada pelo PO e (re)colocada na próxima sprint...
 - “Validação”
- “Cerimônia de encerramento da Sprint”

Retrospectiva da Sprint

- Analisar o que funcionou ou não funcionou durante a Sprint
- Tipicamente de 15 a 30 minutos
- É realizada após cada Sprint
- Todo o time participa

Artefatos do Scrum

- *Product Backlog*
- *Sprint Backlog*
- *Burndown charts*

Product Backlog

- Os requisitos
- Uma lista de todo o trabalho desejado no projeto
- Cada item tem seu peso de acordo com a vontade dos clientes
- Priorizado pelo PO
- (Re)priorizado a cada Sprint

Sprint Backlog

- É uma lista de itens selecionados do *Product Backlog*
 - Contém tarefas concretas que serão realizadas durante a próxima *Sprint*
 - Atividades são retiradas do *Product Backlog* e desmembradas em tarefas...
- É mantido pela equipe que desenvolve o produto durante a *Sprint*

Gerenciando a *Sprint Backlog*

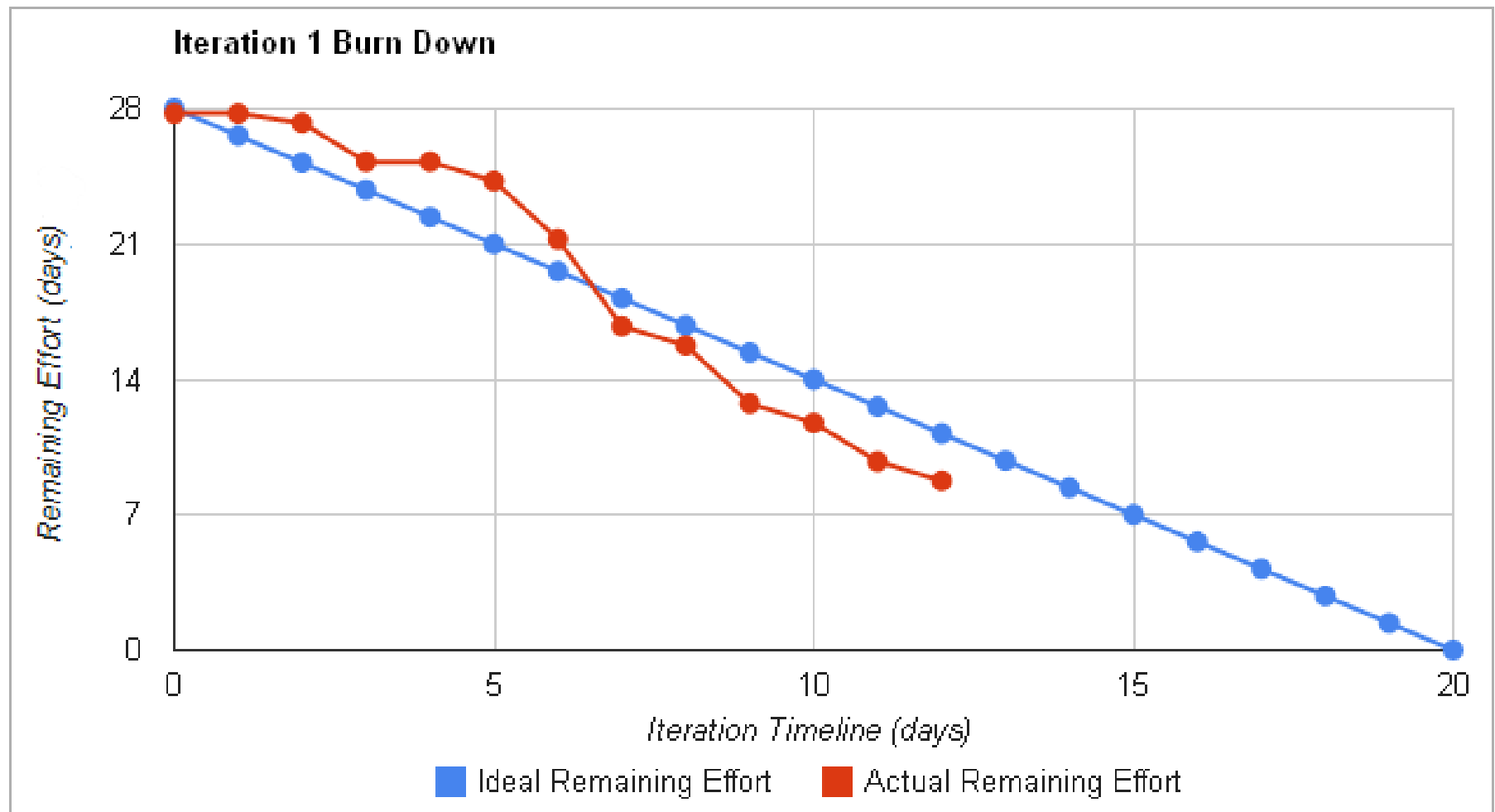


- Cada indivíduo escolhe o trabalho que fará
- Atualização diária da estimativa do trabalho restante
 - *Burndown chart....*
- Qualquer membro pode adicionar, apagar ou mudar tarefas
 - *Atualiza o que precisa ser feito (tarefa) na medida em que o problema se torna melhor compreendido*

Burndown chart

- O burndown é um gráfico simples...
 - Eixos X e Y
 - O eixo Y indica o número de tarefas existentes no Sprint
 - Eixo X os dias que representam o tamanho do Sprint
- Possui duas linhas
 - Andamento ideal do desenvolvimento das atividades
 - Andamento REAL do desenvolvimento das atividades

Burndown chart



Escalabilidade do Scrum

- Times de tamanho pequeno
 - Entre 2 e 7 membros
- Como fazer a escalabilidade?
 - Através de vários times....