



UFOP



decom

departamento
de computação

Aula 1 - Processo de Software

Prof. Johnatan Oliveira

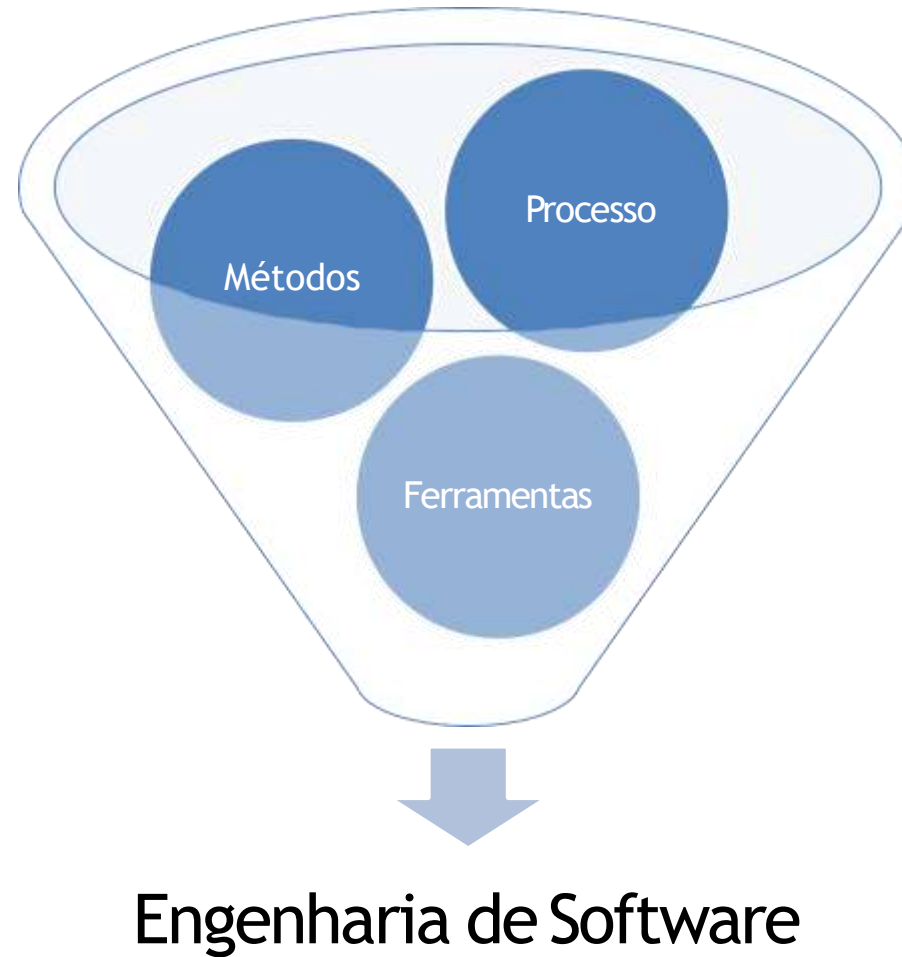
*Material adaptado dos professores: Leonardo Murta, Marco Túlio Valente,
Eduardo Figueiredo*

2021

Histórico (era pós-ES)

- 2000s:
 - Métodos ágeis
 - Desenvolvimento dirigido por modelos
 - Linhas de produto
 - Experimentação
- Atualmente
 - DevOps
 - Continuous*
 - Software Analytics
- – ...

Elementos da ES



Elementos da ES

- Processo
 - Define os passos gerais para o desenvolvimento e manutenção do software
 - Serve como uma estrutura de encadeamento de métodos e ferramentas
- Métodos
 - São os “how to’s” de como fazer um passo específico do processo
- Ferramentas
 - Automatizam o processo e os métodos

Elementos da ES

- Cuidado com o “desenvolvimento guiado por ferramentas”
 - É importante usar a ferramenta certa para o problema
 - O problema não deve ser adaptado para a ferramenta



“Para quem tem um martelo, tudo parece prego”

Diferença entre método, processo e ferramenta

O que é
processo,
método ou
ferramenta?

1. Coloque em uma panela funda o leite condensado, a margarina e o chocolate em pó.
2. Cozinhe [no fogão] em fogo médio e mexa sem parar com uma colher de pau.
3. Cozinhe até que o brigadeiro comece a desgrudar da panela.
4. Deixe esfriar bem, então unte as mãos com margarina, faça as bolinhas e envolva-as em chocolate granulado.



Diferença entre método, processo e ferramenta

1. **Coloque** em uma **panela** funda o leite condensado, a margarina e o chocolate em pó.
2. **Cozinhe** [no **fogão**] em fogo médio e **mexa** sem parar com uma **colher de pau**.
3. **Cozinhe** até que o brigadeiro comece a desgrudar da **panela**.
4. Deixe esfriar bem, então **unte** as mãos com margarina, **faça as bolinhas** e **envolva**-as em chocolate granulado.



método



ferramenta

Processo

O Supermercado de ES

- ES fornece um conjunto de métodos para produzir software de qualidade
- Pense como em um supermercado...
 - Em função do problema, se escolhe o processo, os métodos e as ferramentas
- Cuidado
 - Menos do que o necessário pode levar a desordem
 - Mais do que o necessário pode emperrar o projeto



Processos implícitos x explícitos

- Lembrem-se: Processos sempre existem, seja de forma implícita ou explícita!
 - Processos **implícitos** são difíceis de serem seguidos, em especial por novatos
 - Processos **explícitos** estabelecem as regras de forma clara



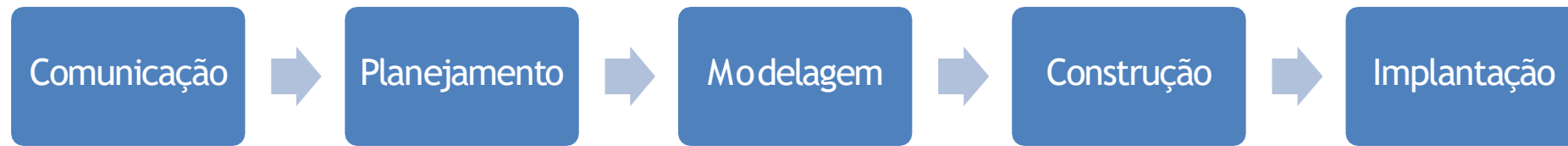
Processo de qualidade

- Última palavra para medir a qualidade de um processo:
Satisfação do Cliente
- Outros indicadores importantes
 - Qualidade dos produtos gerados
 - Custo real do projeto
 - Duração real do projeto

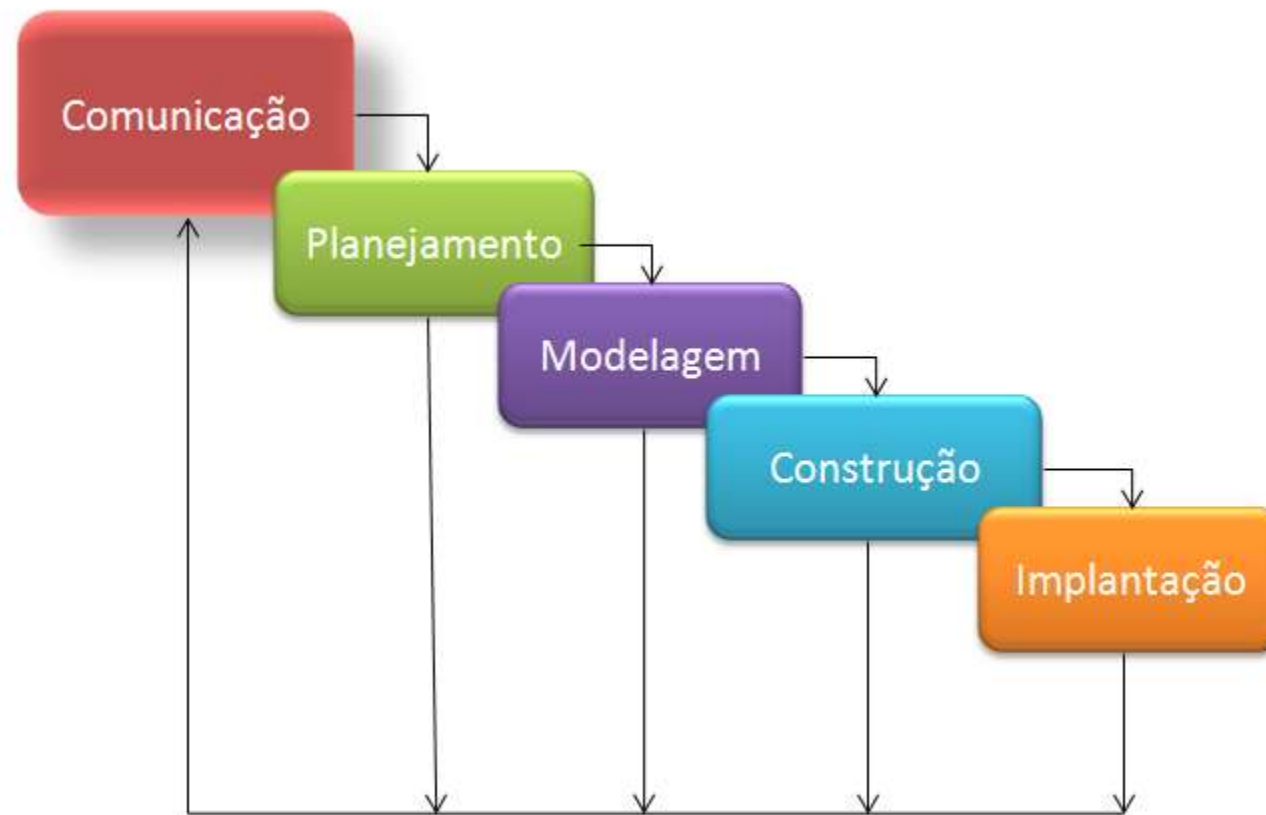
Modelos de ciclo de vida

- Existem alguns processos pré-fabricados
 - Esses processos são conhecidos como modelos de ciclo de vida
 - Esses processos apresentam características predefinidas
- Devem ser adaptados para o contexto real de uso
 - Características do projeto
 - Características da equipe
 - Características do cliente

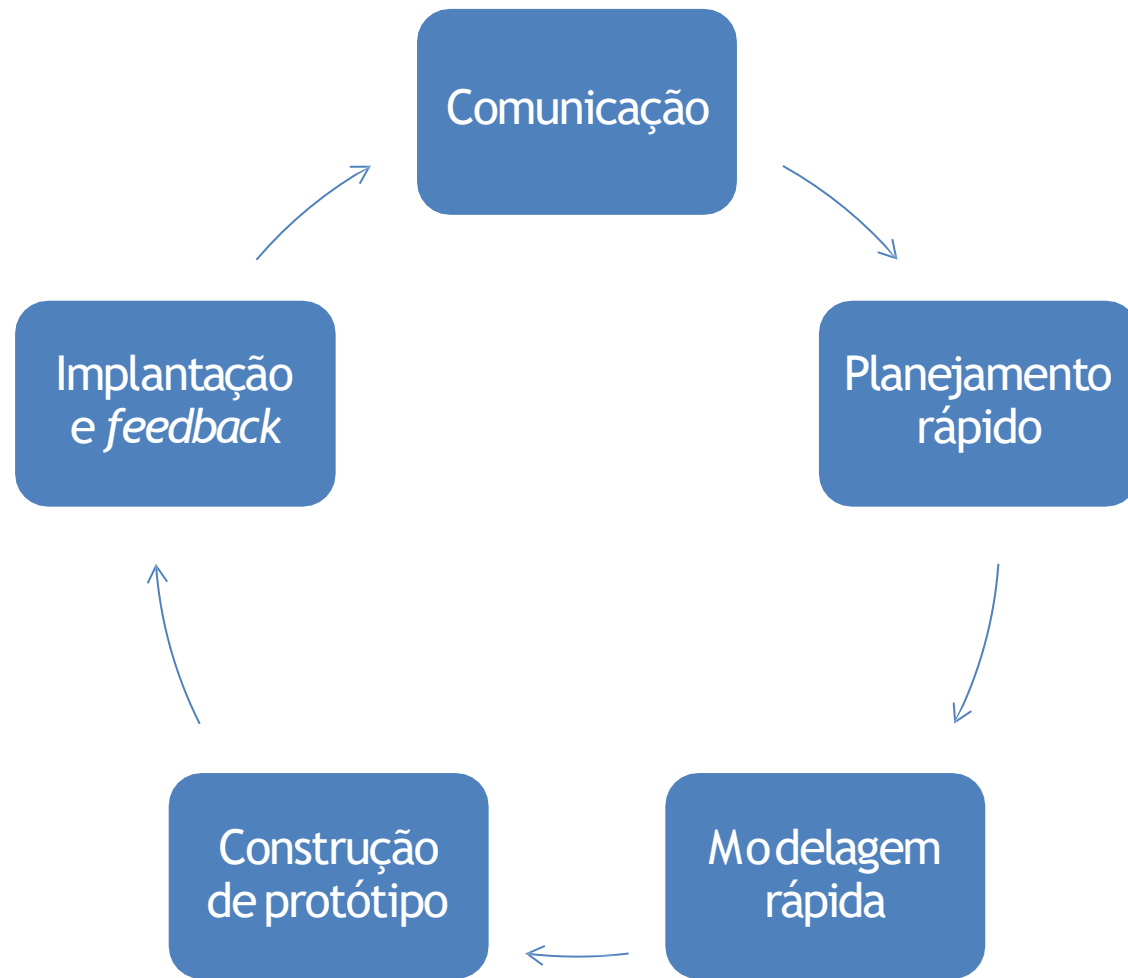
Ciclo de vida Cascata



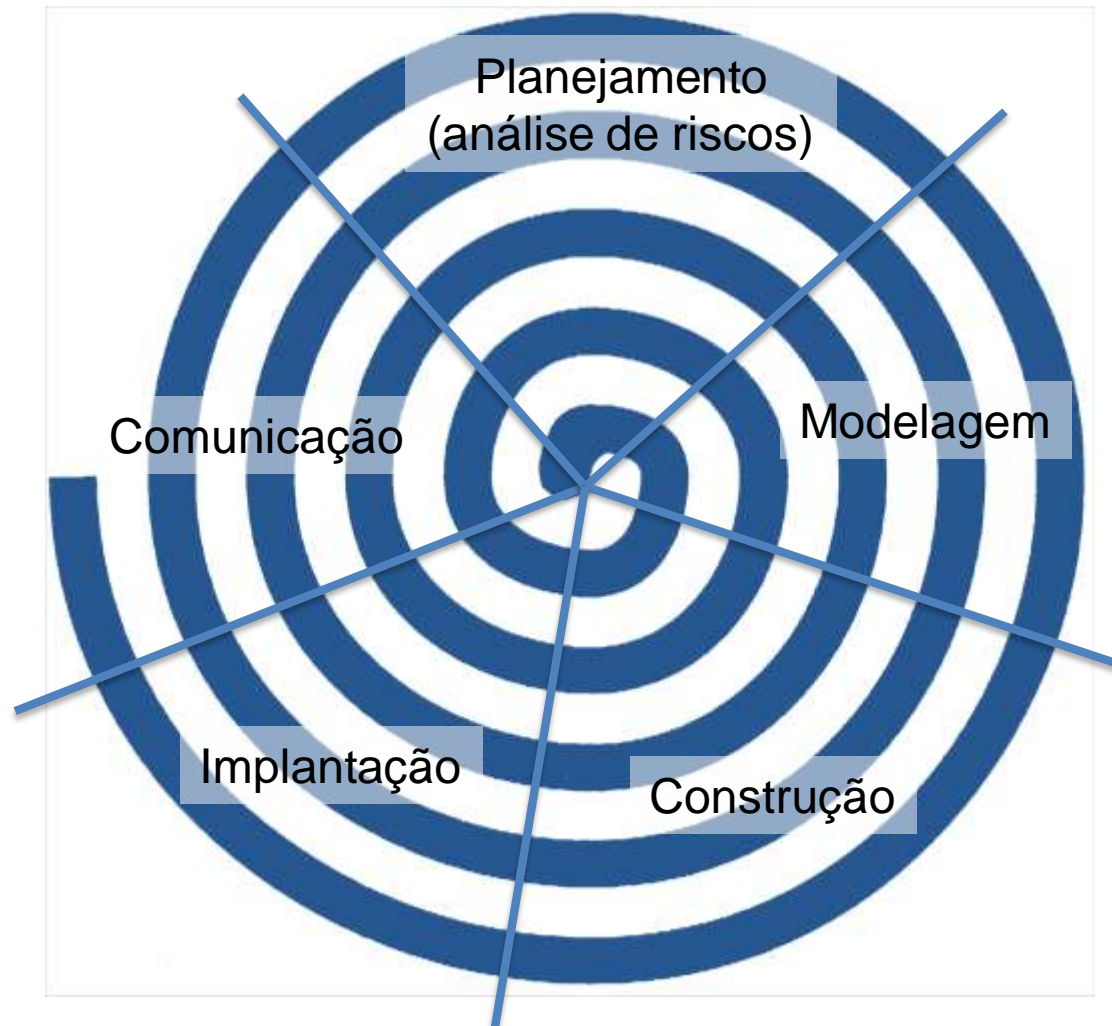
Ciclo de vida Cascata



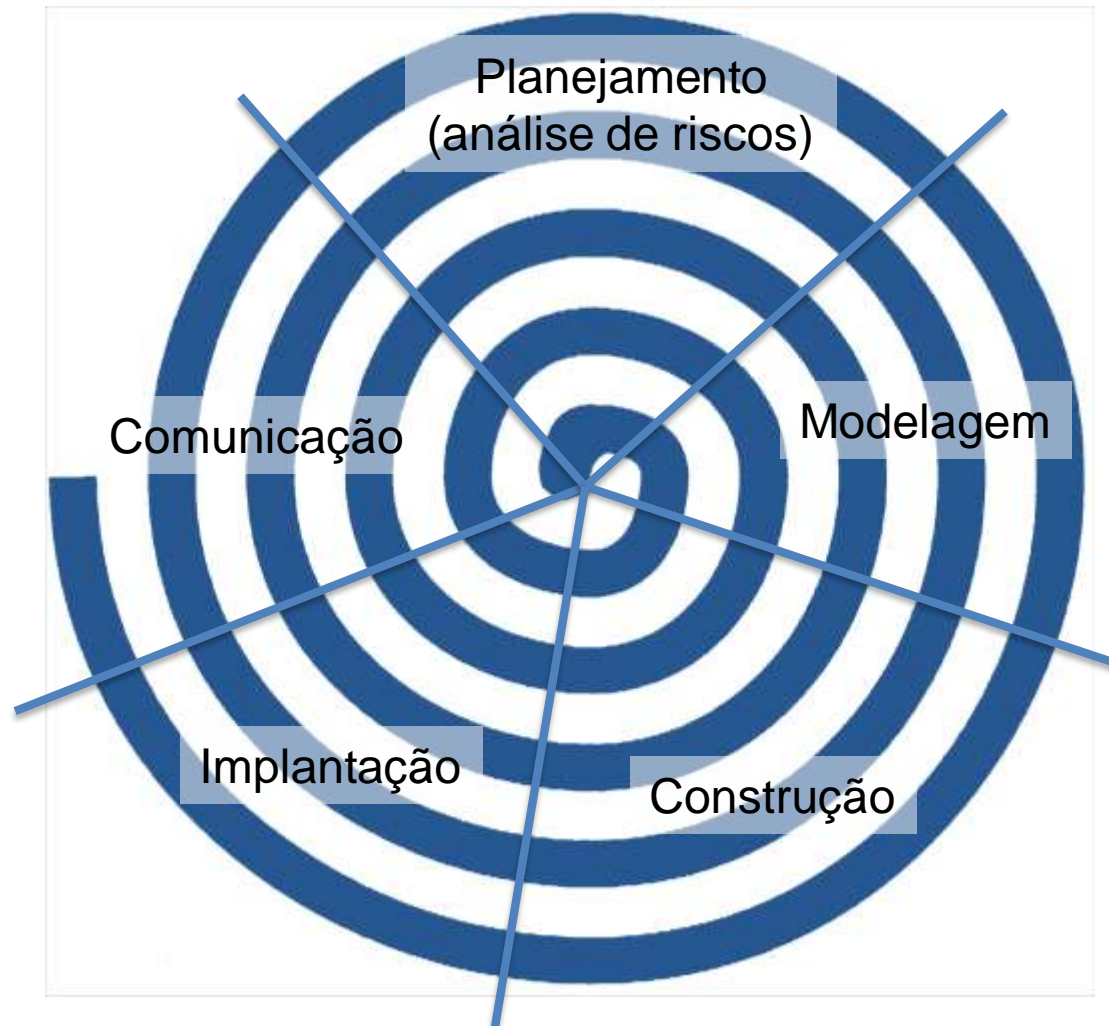
Prototipação



Ciclo de vida Espiral

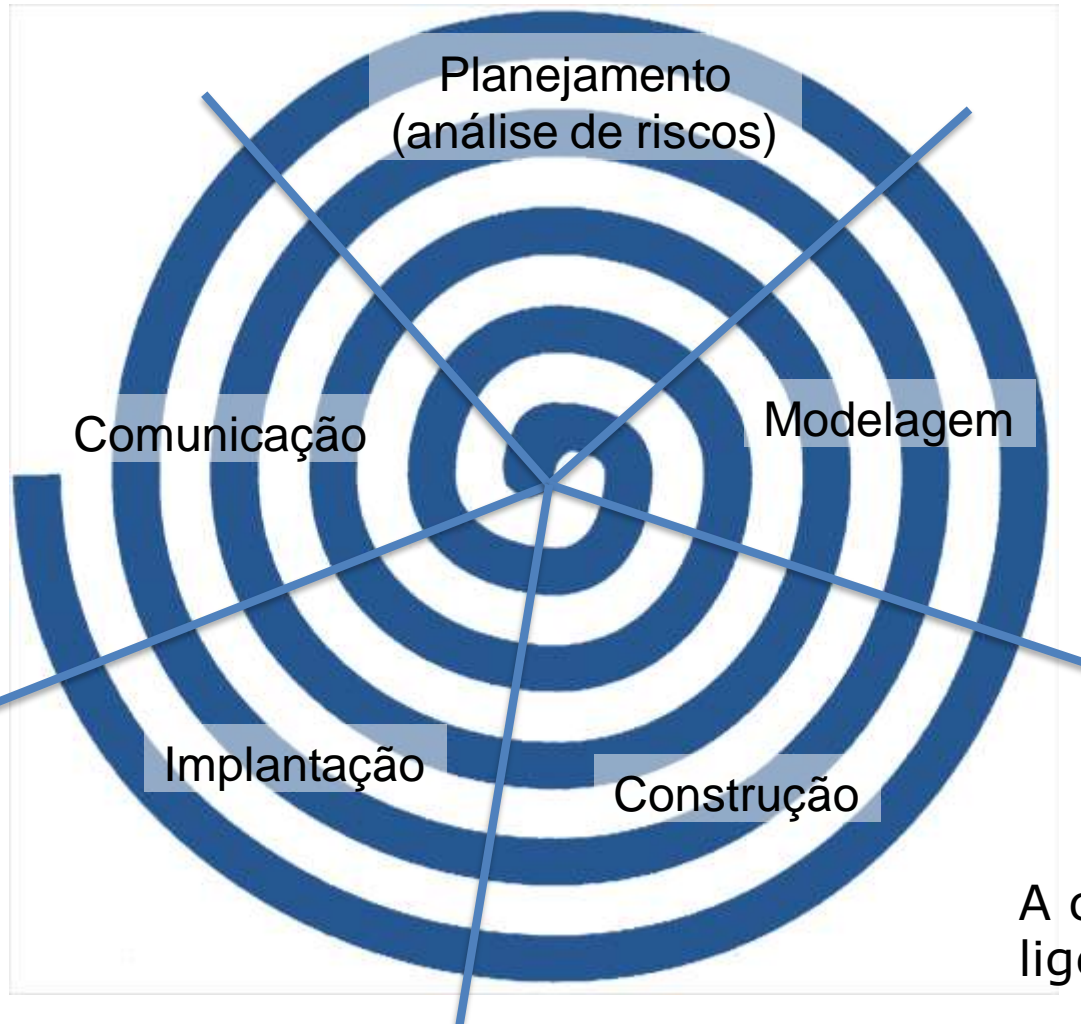


Ciclo de vida Espiral



O modelo espiral foi desenvolvido de modo a combinar as melhores características dos modelos: Linear e Prototipação. O mesmo ainda acrescenta um novo recurso, a *análise de riscos*, inexistente nesses outros modelos.

Ciclo de vida Espiral



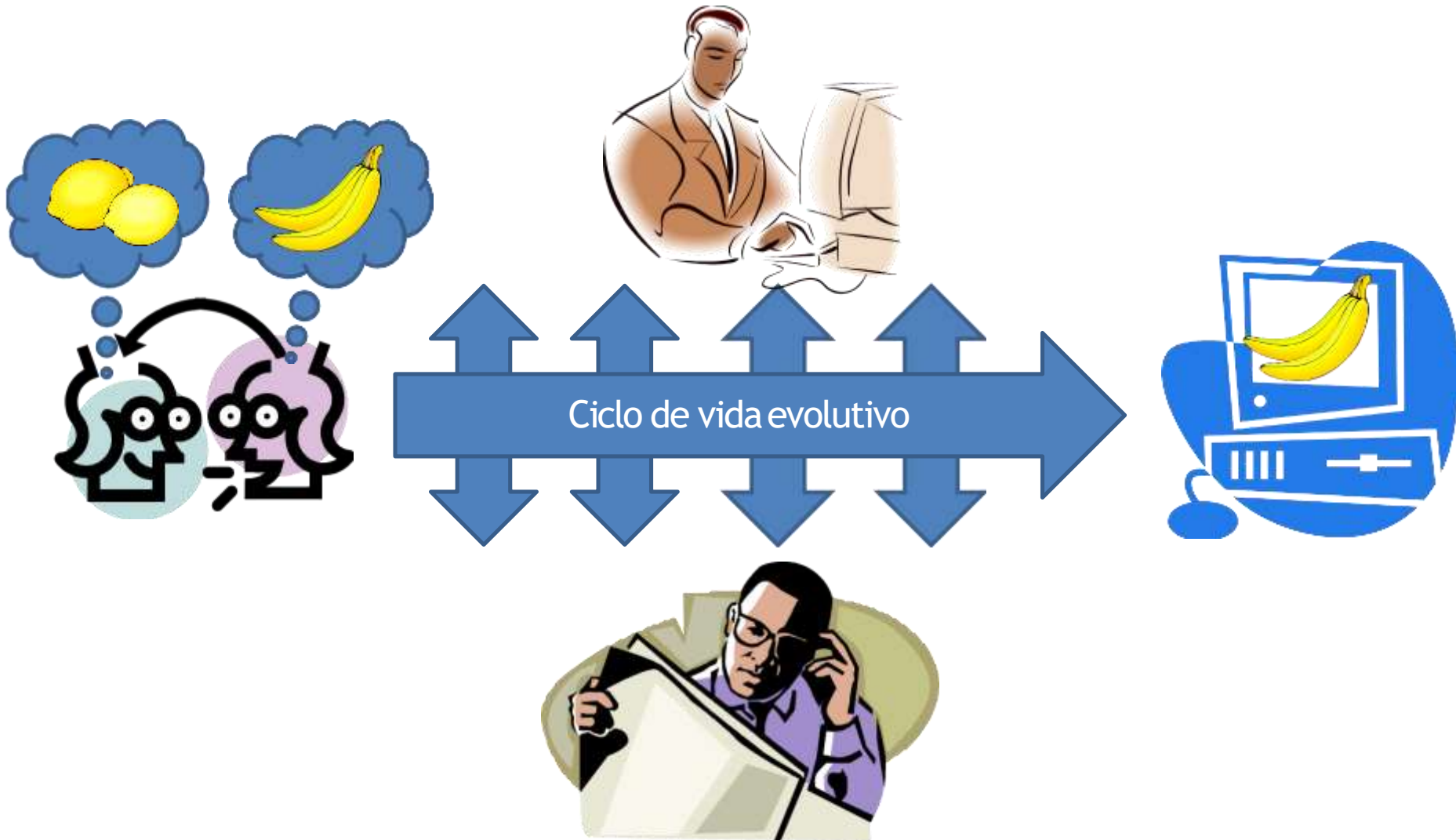
Vantagens	Desvantagens
Estimativas tornam-se mais realísticas	Muita ênfase a parte funcional
Mais versátil para lidar com mudanças	A avaliação dos riscos exige experiência
Melhora o tempo de implementação do sistema	É bem aplicado somente a sistemas de larga escala
Fácil de decidir o quanto testar	O modelo é relativamente novo e não tem sido muito utilizado.
Não faz distinção entre desenvolvimento e manutenção.	

A cada ciclo, o modelo espiral gera um protótipo ligeiramente diferente do anterior sendo

Cascata x Evolutivo



Cascata x Evolutivo

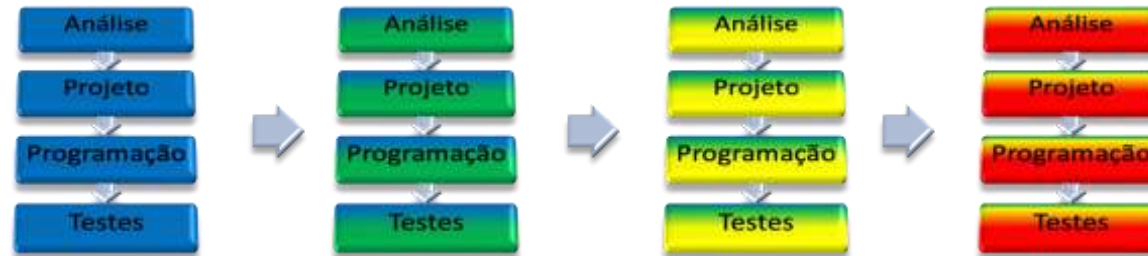


Cascata x Evolutivo

Ciclo de vida cascata



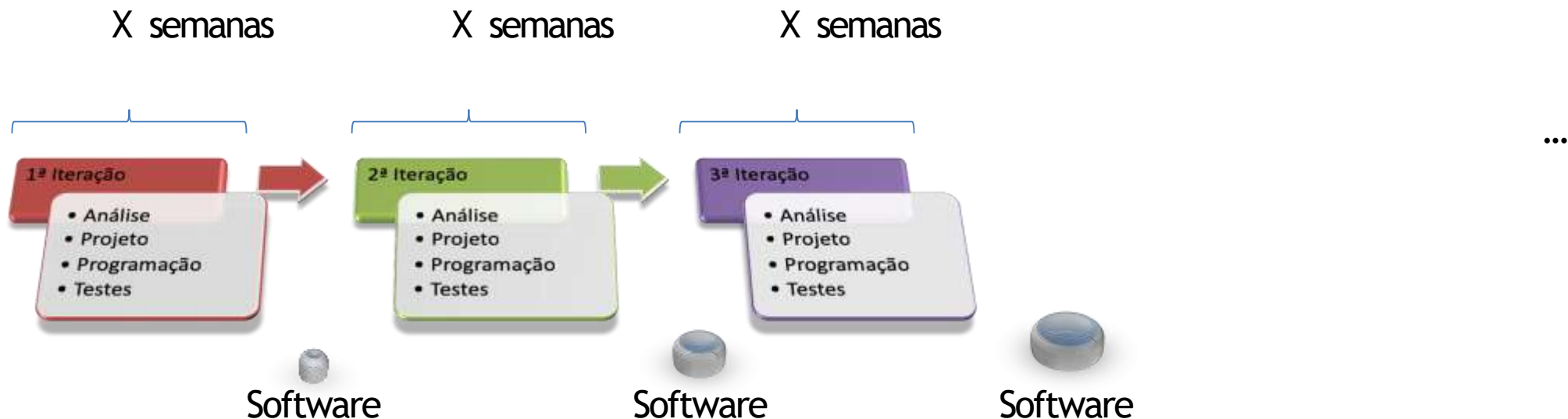
Ciclo de vida evolutivo



- Objetivo: Processo Unificado com aspectos de...
 - Desenvolvimento iterativo
 - Desenvolvimento evolutivo
 - Desenvolvimento ágil

Desenvolvimento Iterativo

- O desenvolvimento é organizado em “mini-projetos”
 - Cada “mini-projeto” é uma iteração
 - Cada iteração tem duração curta e fixa (de 2 a 6 semanas)
 - Cada iteração tem atividades de análise, projeto, programação e testes
 - O produto de uma iteração é um software parcial

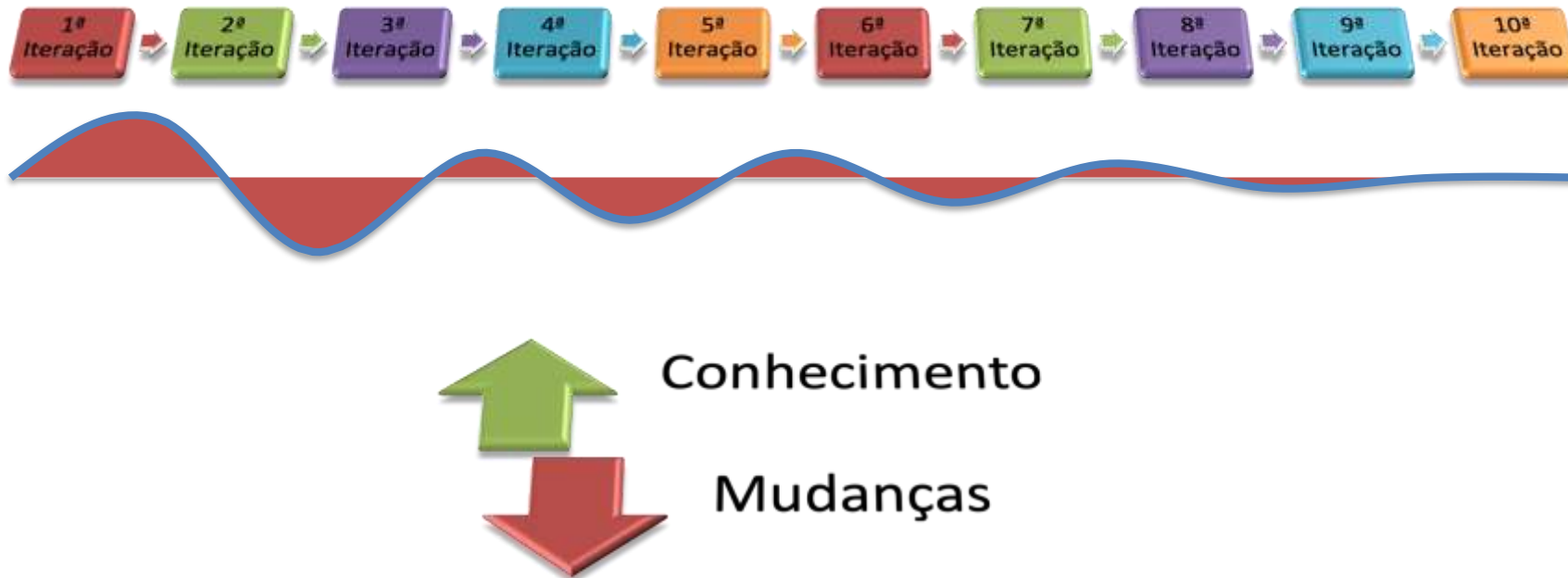


Desenvolvimento Iterativo

- A iteração deve ser fixa
 - Tarefas podem ser removidas ou incluídas
 - A iteração nunca deve passar da duração previamente estipulada
- O resultado de cada iteração é um software...
 - Incompleto
 - Em desenvolvimento (não pode ser colocado em produção)
 - Mas não é um protótipo!!!
- Esse software pode ser verificado e validado parcialmente
 - Testes
 - Usuários
- Podem ser necessárias diversas iterações (e.g. 10 a 15) para ter uma versão do sistema pronta para entrar em produção

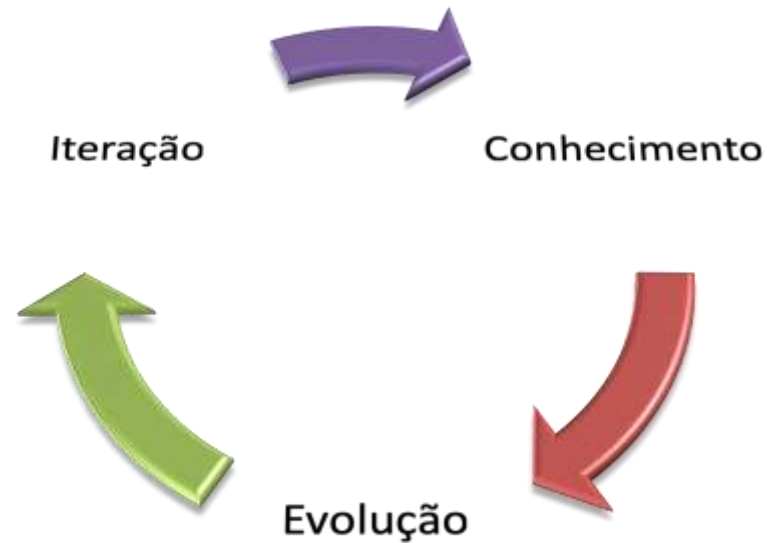
Desenvolvimento Iterativo

- Iterações curtas privilegiam a propagação de conhecimento
 - Aumento do conhecimento sobre o software
 - Diminuição das incertezas, que levam às mudanças



Desenvolvimento Evolutivo

- As especificações evoluem a cada iteração
 - A cada iteração, uma parte do software fica pronta
 - O conhecimento sobre o software aumenta
 - As especificações são evoluídas para retratar esse aumento de conhecimento sobre o que é o software



Desenvolvimento Evolutivo

- Mudanças sempre acontecem em projetos de software
 - Requisitos mudam
 - O ambiente em que o software está inserido muda
 - As pessoas que operam o software mudam
- Estratégias para lidar com mudanças
 - Evitar as mudanças (corretivas) fazendo uso de boas técnicas de engenharia de software
 - Acolher mudanças por meio de um processo evolutivo

Desenvolvimento Ágil



- São dadas respostas rápidas e flexíveis a mudanças
 - O projeto é replanejado continuamente
 - São feitas entregas incrementais e constantes do software, refletindo as mudanças solicitadas

Planejamento tradicional



Desenvolvimento Ágil



Desenvolvimento Ágil

12 Princípios Ágeis

Satisfazer o cliente com software que agregue valor

Acolher modificações nos requisitos

Entregar o software funcional com frequência

Trabalhar junto ao cliente

Manter as pessoas motivadas e confiar nelas

Promover conversas face a face

Medir o progresso com software funcionando

Manter um ritmo constante de trabalho

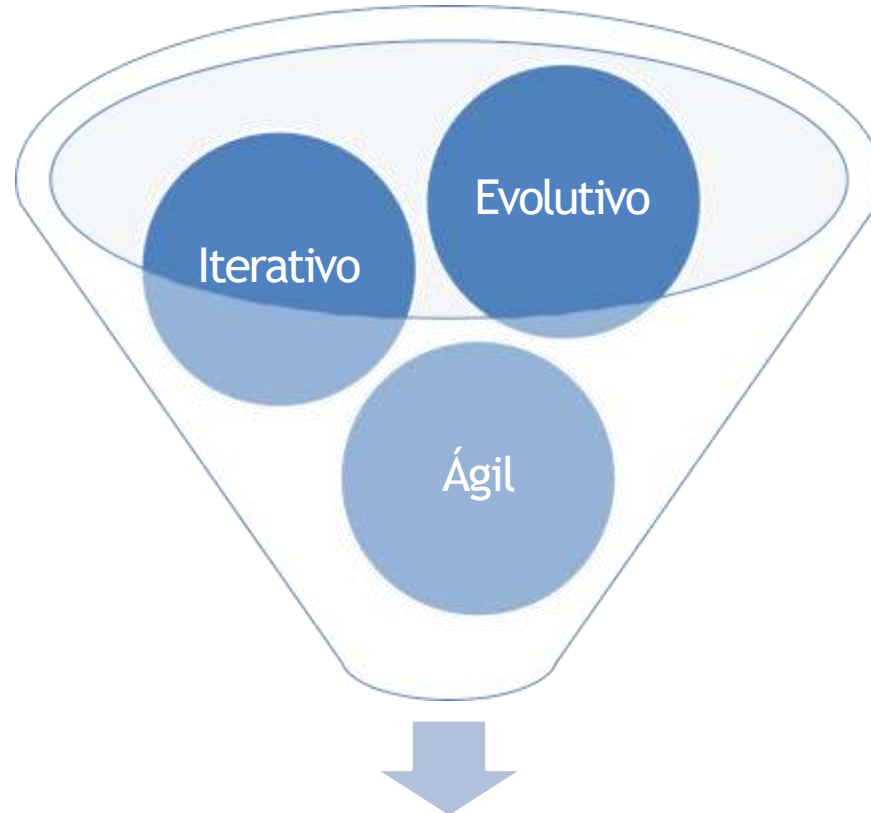
Prezar por excelência técnica

Buscar por simplicidade

Trabalhar com equipes auto-organizadas

Ajustar o comportamento da equipe buscando mais efetividade

Processo Unificado



Processo Unificado

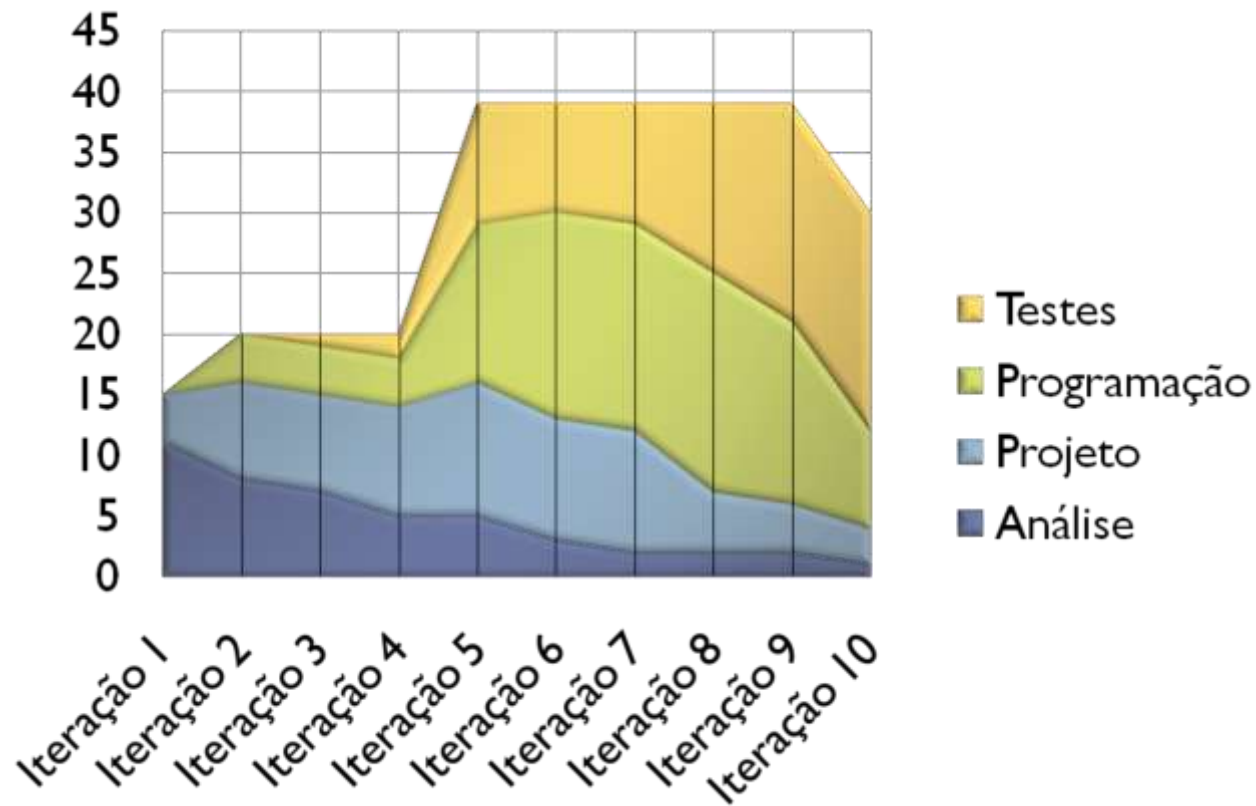
Processo Unificado (benefícios esperados)

- Mitigação de riscos precoce
- Visibilidade do progresso
- Envolvimento e comprometimento do usuário
- Controle sobre a complexidade
- Aprendizado incremental
- Menos defeitos
- Mais produtividade

Processo Unificado (exemplo)

- Analisar os requisitos no início do projeto
 - Casos de uso
 - Lista de requisitos não funcionais
- Priorizar os casos de uso
 - Significativos para a arquitetura como um todo
 - Alto valor de negócio
 - Alto risco
- Em cada iteração
 - Selecionar alguns casos de uso por ordem de prioridade para serem analisados em detalhes
 - Atribuir tarefas para a iteração a partir da análise detalhada desses casos de uso
 - Fazer projeto e programação de parte do software
 - Testar a parte do software recém projetada e programada e criar a *baseline* da iteração
 - Apresentar a *baseline* da iteração ao usuário

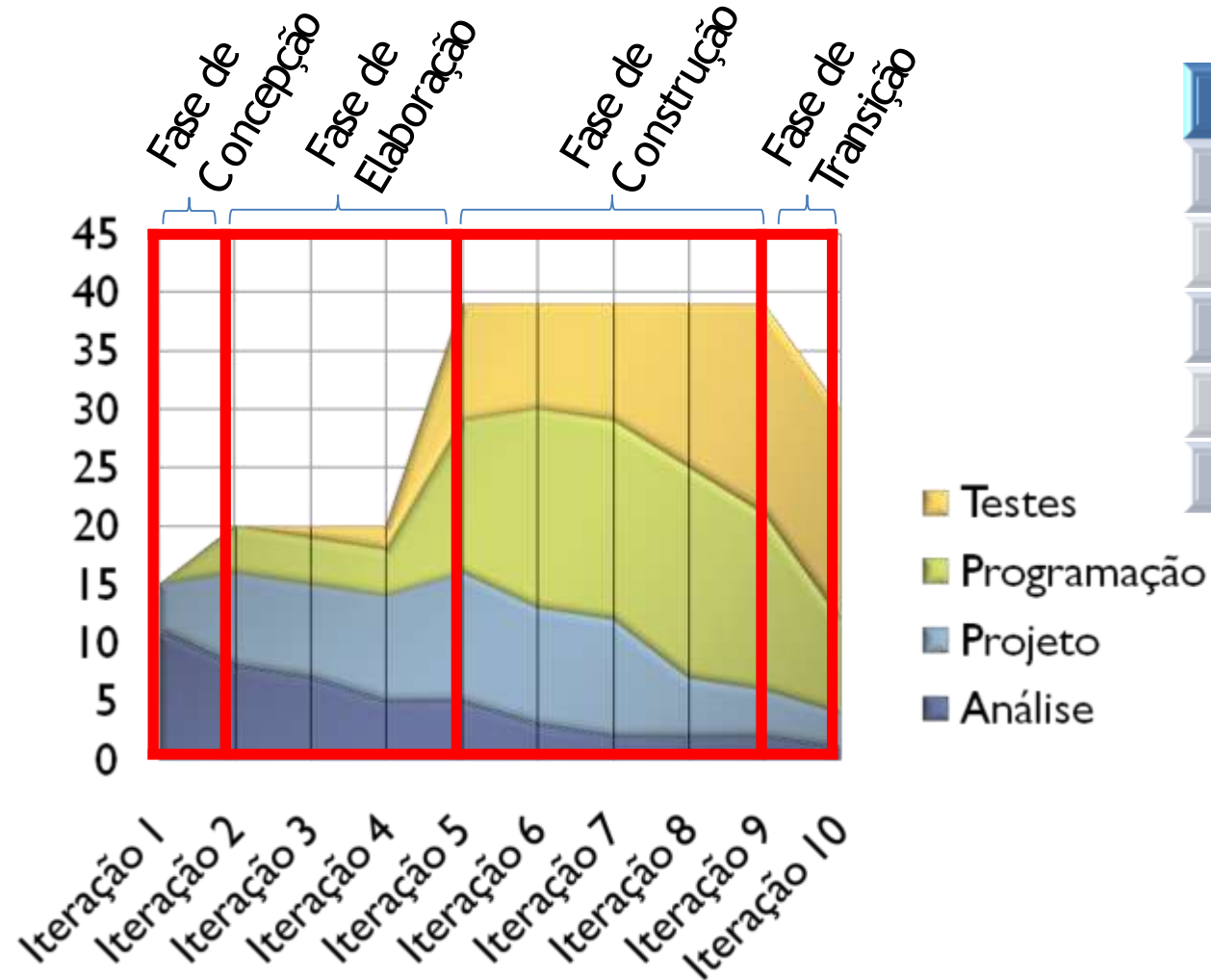
Processo Unificado (exemplo)



Processo Unificado (fases)

- O desenvolvimento pode ser decomposto em fase, com o intuito de retratar a ênfase principal das iterações
 - Concepção
 - Elaboração
 - Construção
 - Transição
- Plano da fase
 - Abrangente e superficial
- Plano da iteração
 - Específico e detalhado

Processo Unificado (exemplo)



Atividade	Esforço
Análise	10%
Projeto	15%
Programação	30%
Testes	15%
Gerência	30%

Processo Unificado (concepção)

- Consiste de
 - Identificação de riscos
 - Listagem inicial dos requisitos
 - Esboço dos casos de uso
 - Identificação de arquiteturas candidatas
 - Estimativas iniciais de cronograma e custo
- Principais características
 - Menor fase do projeto
 - Escopo ainda vago
 - Estimativas ainda vagas
- Esforço e duração aproximados
 - 5% do esforço do projeto
 - 10% da duração do projeto

Processo Unificado (elaboração)

- Consiste de
 - Mitigação dos riscos
 - Detalhamento da maioria dos requisitos e casos de uso
 - Estabelecimento e validação da arquitetura do software
 - Detalhamento das estimativas de cronograma e custo
- Principais características
 - Grande parte das atividades de análise e projeto já concluída
 - Diminuição significativa das incertezas
 - *Baseline* da arquitetura é estabelecida
- Esforço e duração aproximados
 - 20% do esforço do projeto
 - 30% da duração do projeto

Processo Unificado (construção)

- Consiste de
 - Implementação dos demais componentes da arquitetura
 - Preparação para a implantação
- Principais características
 - Maior fase do projeto
 - *Baseline* de testes do produto é estabelecida
- Esforço e duração aproximados
 - 65% do esforço do projeto
 - 50% da duração do projeto

Processo Unificado (transição)

- Consiste de
 - Execução de testes finais
 - Implantação do produto
 - Treinamento dos usuários
- Principais características
 - *Baseline* de liberação do produto é estabelecida
- Esforço e duração aproximados
 - 10% do esforço do projeto
 - 10% da duração do projeto

Processo Unificado (características)

- Os requisitos não são completamente definidos antes do projeto
- O projeto não é completamente definido antes da programação
- A modelagem não é feita de forma completa e precisa
- A programação não é uma tradução mecânica do modelo para código
- As iterações não duram meses, mas sim semanas
- O planejamento não é especulativo, mas sim refinado durante o projeto

Bibliografia

- Larman, C.; 2007. Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3 ed. Bookman.
- Pressman, R. S.; 2004. Software Engineering: A Practitioner's Approach. 6 ed. McGraw-Hill.