

Processos e Qualidade de Software

Processos de Desenvolvimento Dirigidos por Planos e Ágeis



PUC Minas

Instituto de Ciências Exatas
e Informática

Prof. Lesandro Ponciano

Departamento de Engenharia de Software
e Sistemas de Informação (DES)

Questões Motivadoras

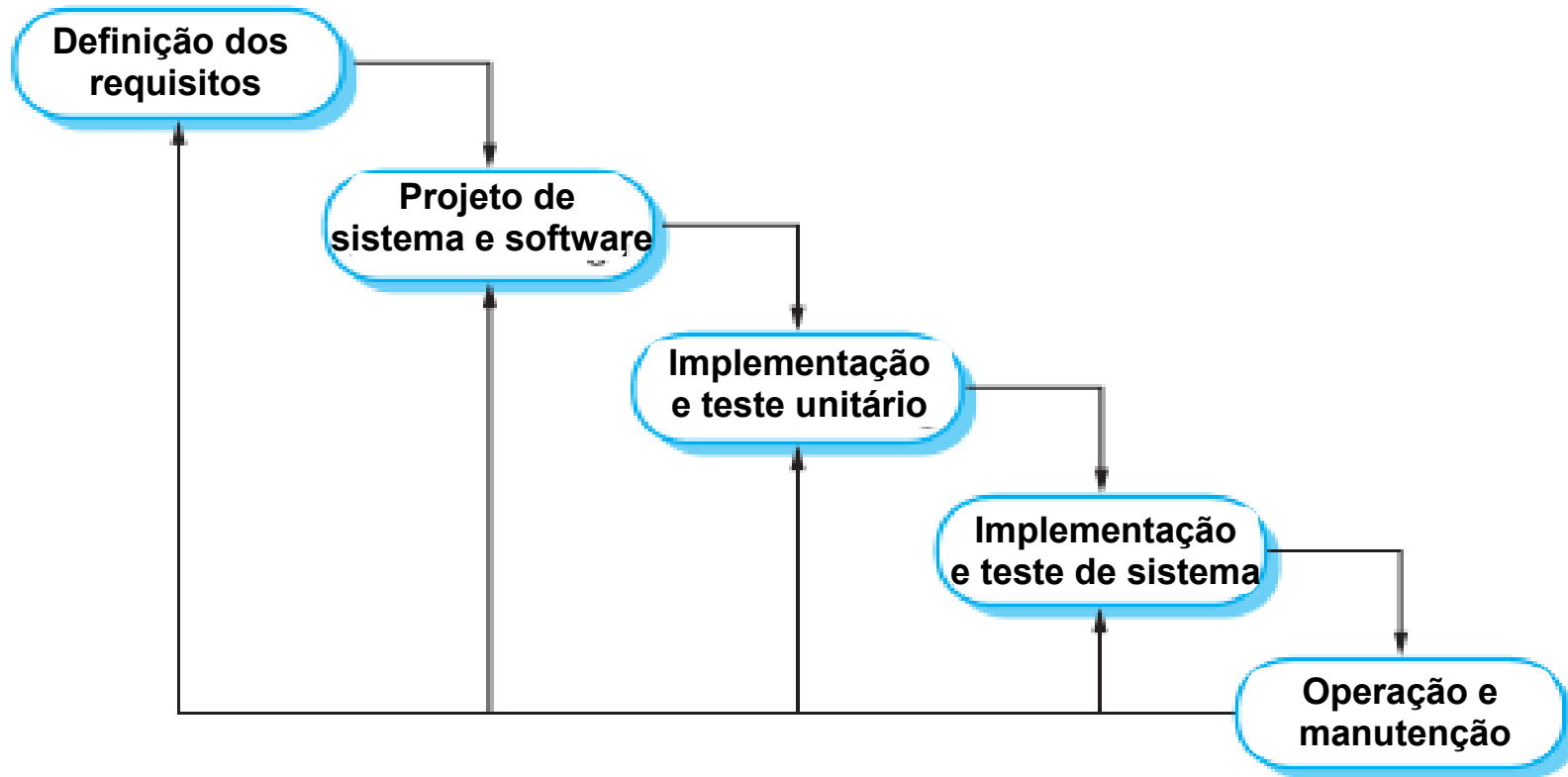
- O que é desenvolvimento baseado em planos e desenvolvimento ágil? Quais as principais diferenças?
- O que é o manifesto ágil?
- Quais são as situações em que é melhor usar desenvolvimento baseado em planos?
- Quais são os fatores humanos necessários em uma equipe de desenvolvimento ágil?

Objetivos da Aula

- Discutir conceitos de desenvolvimento dirigido a plano
 - Visão geral
 - Aplicabilidade
 - Problemas
- Discutir conceitos de desenvolvimento ágil
 - Visão geral
 - Aplicabilidade
 - Fatores humanos
 - Problemas
- Analisar a integração e escolha entre essas abordagens

Modelos de Processos

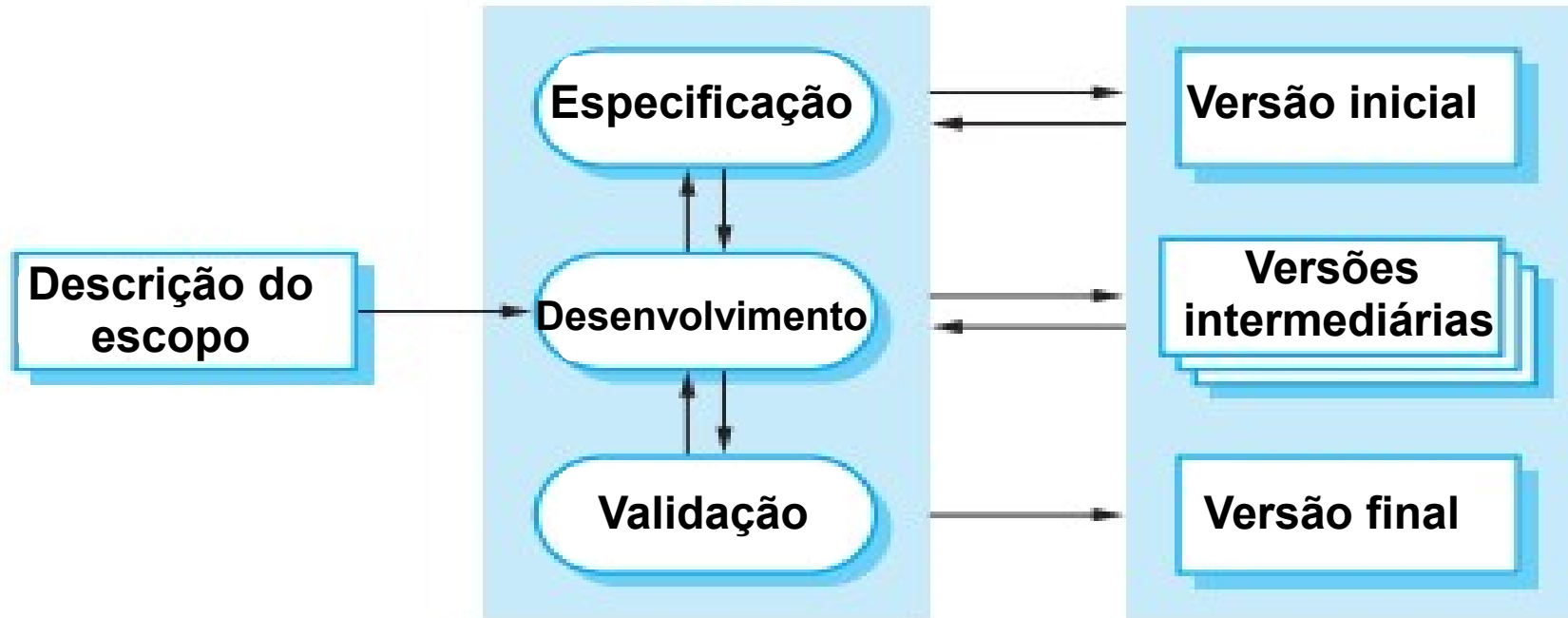
Cascata



Modelos de Processos

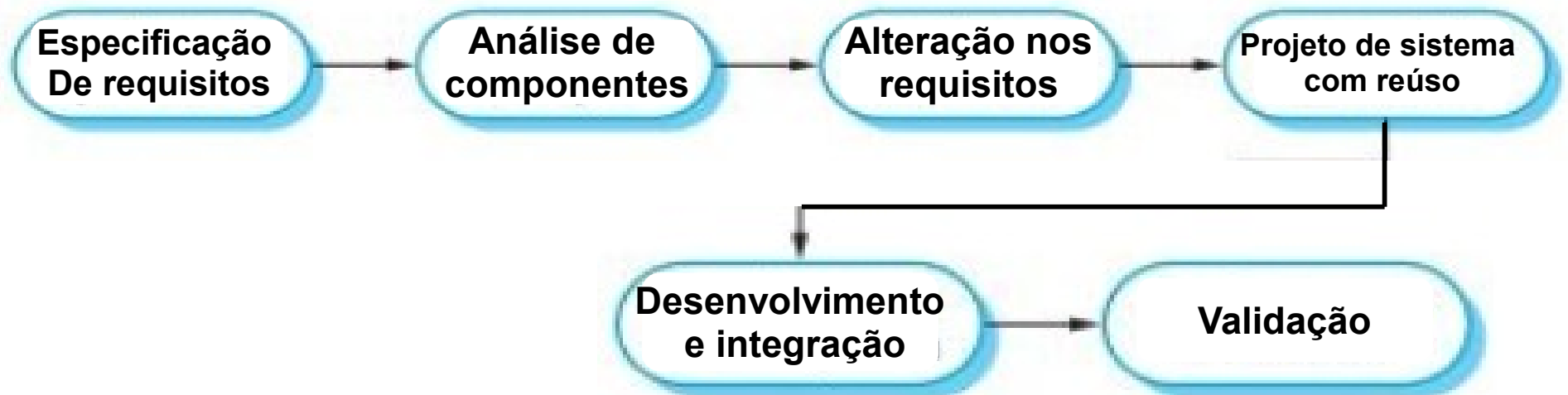
Incremental

Atividades concorrentes



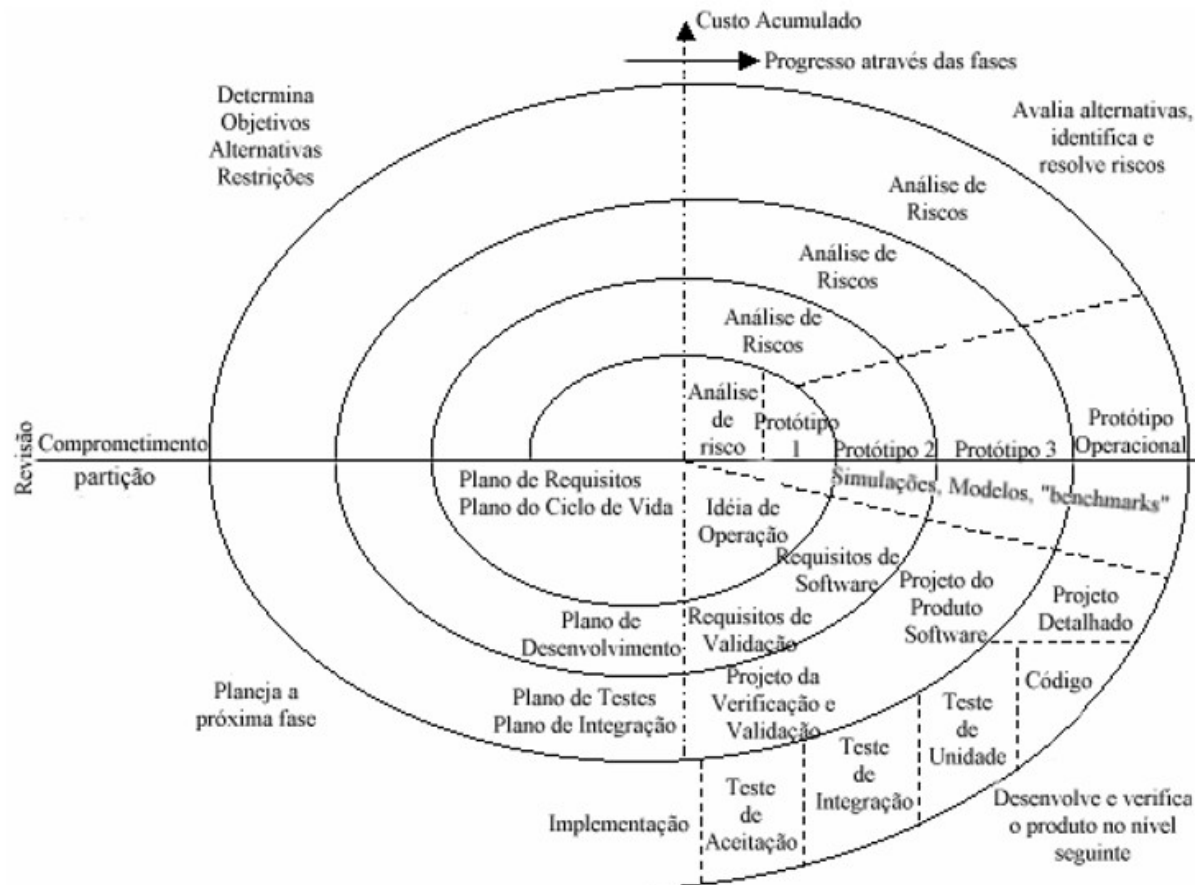
Modelos de Processos

Engenharia de Software Orientada a Reuso



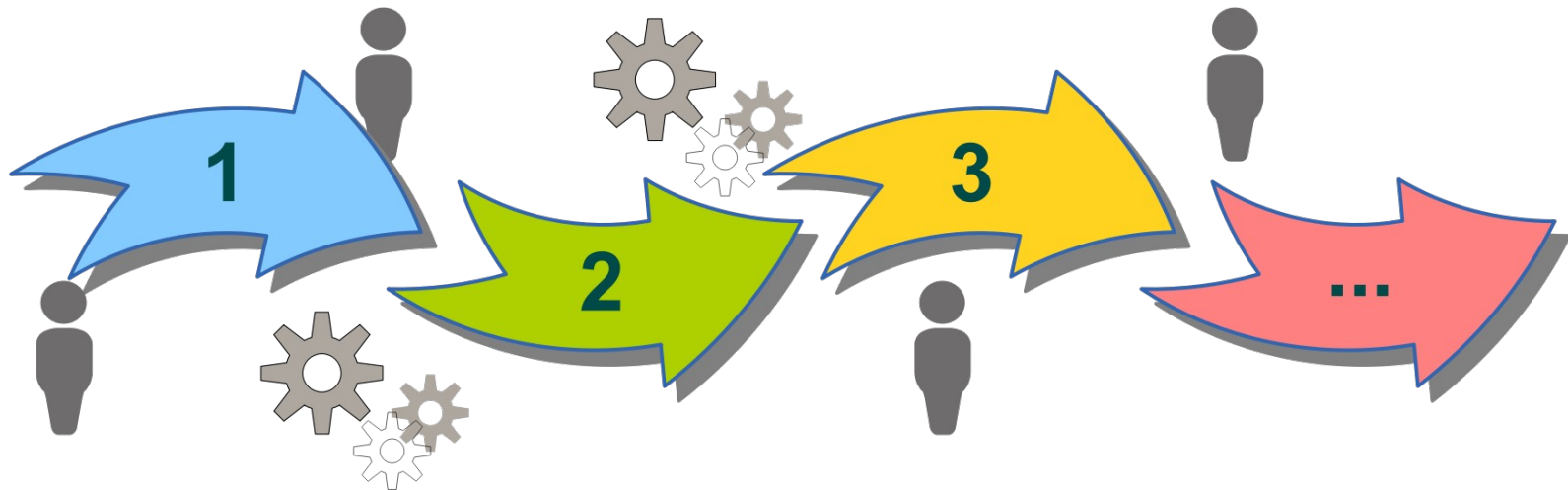
Modelos de Processos

Espiral



Processo de Software

- Um processo de software consiste de um conjunto de **atividades** realizadas por **pessoas** e cujo objetivo é o desenvolvimento ou evolução de um **artefato**, que é o software e sua documentação



Processo de Software

- Conceitos básicos
 - **Papel** é uma função desempenhada por um membro da equipe de desenvolvimento
 - **Atividade** é uma tarefa realizada por uma pessoa em um determinado papel a fim de gerar um determinado artefato
 - Atividades consomem e produzem artefatos
 - Estão sujeitas a pré-condições e pós-condições
 - **Artefato** é o produto de uma atividade, como um modelo, documento, código

Processos Dirigidos a Planos

- São processos em que as atividades são “planejadas com antecedência”
 - Processos e modelos tradicionais/convencionais
- Trabalha com ideia de **previsibilidade**
 - A evolução é medida em relação ao plano
- Dinâmica
 - 1) especificar completamente os requisitos
 - 2) projetar completamente o sistema
 - 3) construir todo o sistema
 - 4) testar todo o sistema implementado

Visão Geral

- Há um contrato formal entre o cliente e a equipe de desenvolvimento
 - Documento de requisitos
 - Muita análise e especificação (planos) antes do início do desenvolvimento
- Interações com documentos formais
 - Para planejamento, execução e análise da evolução
- Pode não ser necessariamente baseado em um modelo em cascata, desenvolvimento incremental dirigido a planos também é possível

Aplicabilidade

- Sistemas em que uma análise completa do sistema é essencial
 - Os planos funcionam como uma formalização
 - Sistemas críticos
- Ambiente de negócio que não se caracterizam por mudanças rápidas
- Muitas pessoas diferentes estão envolvidas na manutenção do software durante sua vida
 - Documentação extensiva é necessária

Problemas

- Mudanças nos requisitos e descoberta de problemas nos requisitos fazem com que projeto, implementação e testes precisem ser adaptados
 - Em processos baseados em modelos como Cascata, isso exigiria muito tempo
- *Overhead* de documentação
 - Gasta-se mais tempo em atividades de análise e documentação do que no desenvolvimento e teste do software
 - O cliente demora a ver algo funcional

Métodos Ágeis

- Origem a partir da metade de 1990
- Acomodar mudanças e tratar o *overhead* no planejamento, projeto e documentação do sistema
 - Trabalha com a ideia de **adaptabilidade**
- *Overhead* de documentação é injustificado em
 - Sistemas para empresas de pequeno porte
 - Sistemas que serão alterados rapidamente em função da dinâmica do negócio
 - Sistemas que serão mantidos por pouco tempo
 - Manutenção inexistente ou pouca

Ideias Básicas

- Conversa face a face é a maneira mais eficiente e efetiva de transmitir informações
 - Oposição a contratos e documentos formais
- Software funcional é a principal medida de progresso
 - Garantir a satisfação do consumidor entregando rapidamente e continuamente software funcional
 - Ciclos de interação com o cliente
 - Novas funcionalidades são adicionadas a cada ciclo
- Desenvolvimento por uma equipe

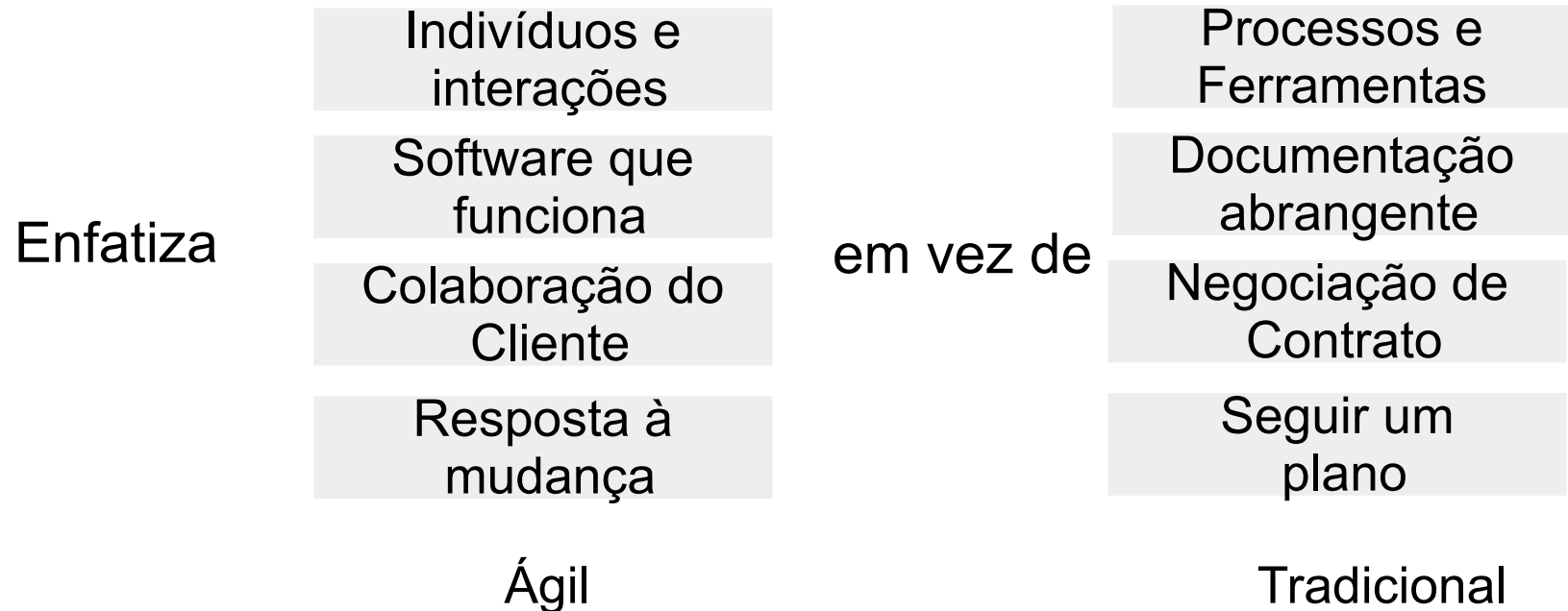
Visão Geral

- Abordagem **incremental** e **iterativa**
- Projeto e implementação são atividades centrais que incorporam as outras atividades
- As saídas das atividades são negociadas durante o desenvolvimento do software
- Pagamento pelo tempo em vez de ser pelo sistema
 - Como não há um documento formal de requisitos, o pagamento passa a ser pelo tempo dos ciclos de desenvolvimento

Valores do Manifesto Ágil



Manifesto Ágil: Valores



Manifesto ágil, Kent Beck et al. (2001)
Manifesto for Agile Software Development <http://agilemanifesto.org/>

Os 12 Princípios



Princípios

- | | | |
|--|---|---|
| 1 Satisfazer o cliente com a entrega adiantada e contínua | 5 Envolver, motivar, apoiar e confiar nos indivíduos | 9 Excelência técnica e bom design aumentam a agilidade |
| 2 Aceitar e se adequar as mudanças | 6 Priorizar uma conversa direta e pessoal | 10 Simplicidade: Evitar trabalho que não precisa ser feito |
| 3 Entregas com frequências e períodos curtos | 7 As entregas representam o progresso do trabalho | 11 Times auto-organizáveis geram melhores resultados |
| 4 Times Multidisciplinares trabalhando diariamente | 8 Processos ágeis promovem um ambiente sustentável | 12 Reflexões, ajustes e otimização em intervalos regulares |

Os 5 Princípios Principais

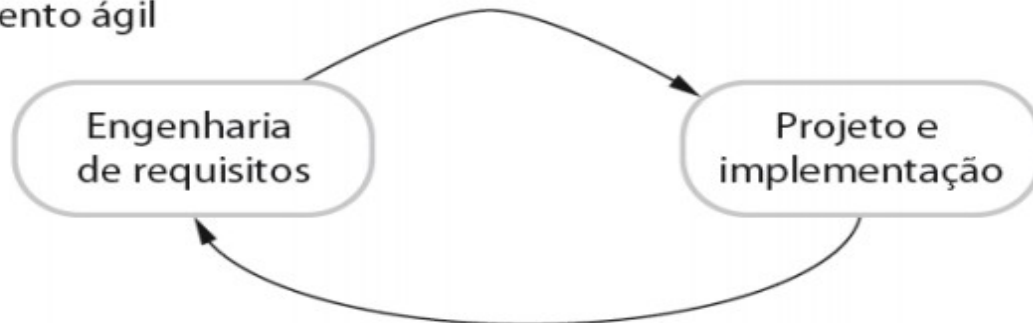
Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Especificação

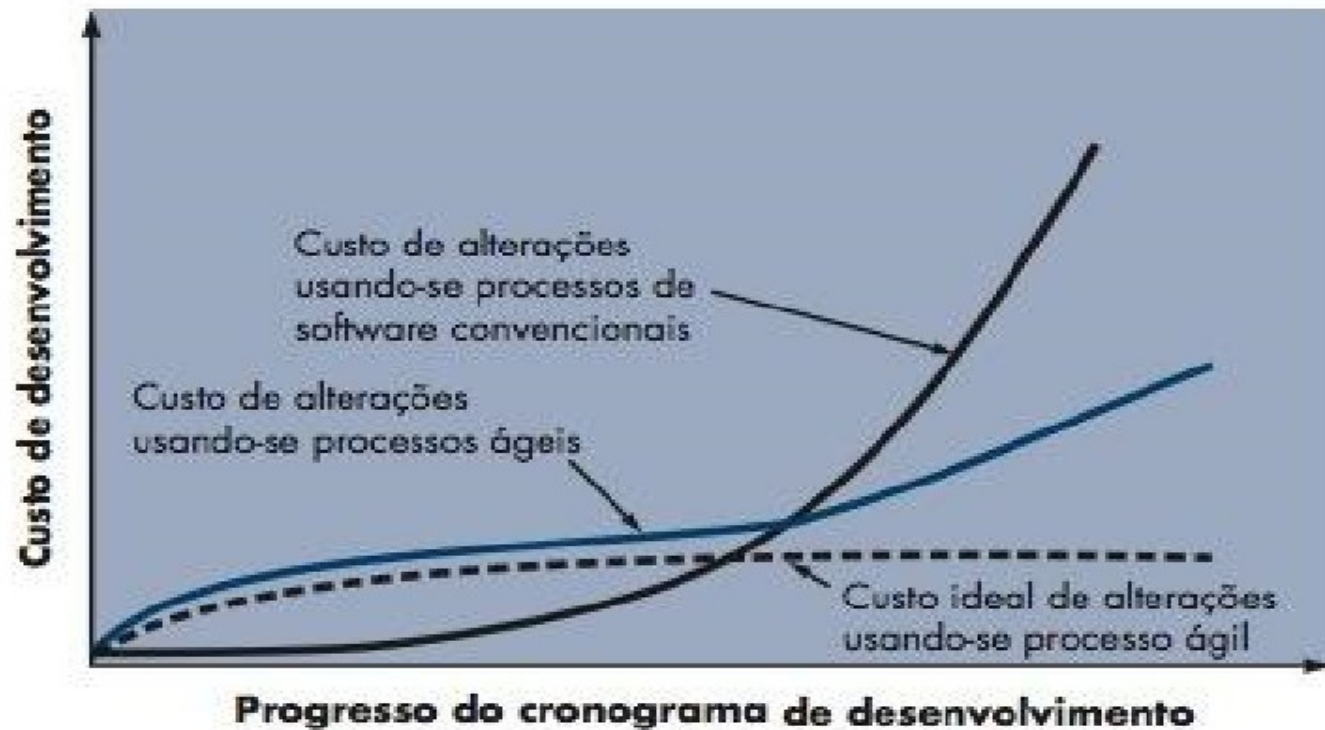
Desenvolvimento baseado em planos



Desenvolvimento ágil



Mudanças e Custo



Fatores Humanos

- Traços-chave entre as pessoas na equipe ágil e a equipe em si
- **Competência**
 - É necessário habilidade e conhecimento sobre o processo
- **Foco comum**
 - Todos os membros devem estar focados no mesmo objetivo: “entregar ao cliente um incremento do software funcionando e no prazo”

Fatores Humanos

- Colaboração
 - Troca de informações entre os membros da equipe e demais envolvidos
- Habilidade na Tomada de Decisão
 - Controlar o próprio destino
 - Autonomia e autoridade à equipe
- Habilidade de Solução de Problemas Confusos
 - Ambiguidade e mudanças contínuas
 - Aceitar que o problema que se está solucionando hoje pode não ser o problema que necessita de ser solucionado amanhã

Fatores Humanos

- **Confiança Mútua e Respeito**
 - Equipe tão forte e unida que “o todo fica mais forte do que a soma das partes”
- **Auto-organização**
 - Trabalho a ser feito
 - Processo no contexto
 - Cronograma e entrega

Problemas em Métodos Ágeis

- Perfil dos clientes
 - *Stakeholders* com diferentes prioridades de mudança podem gerar dificuldades
 - Clientes precisam estar dispostos e ser capazes de passar tempo com a equipe de desenvolvimento
 - Alguns clientes (empresas) podem ter dificuldade de lidar com a informalidade de métodos ágeis
- Tamanho e interação na equipe
 - Funciona bem em equipes pequenas (< 10 pessoas), mas pouco se sabe sobre o funcionamento em equipes grandes (> 20 pessoas)
 - Requer membros com personalidade para um intenso envolvimento na equipe

Problemas em Métodos Ágeis

- Manutenção do software
 - Pouco se sabe sobre a manutenção de sistemas desenvolvidos usando métodos ágeis
 - É possível? Dada a pouca existência de documentação
 - Pode-se usar métodos ágeis nesta atividade?
- Continuidade da equipe de desenvolvimento
 - Membros da equipe conhecem sobre aspectos do sistema sem documentação
 - Quando há mudanças na equipe esse pressuposto é quebrado e há dificuldades
- Manter a simplicidade exige trabalho extra

Aplicabilidade de Métodos Ágeis

- Desenvolvimento de produto
 - Quando a empresa de software está desenvolvendo um produto pequeno ou médio para venda
- Desenvolvimento de sistema personalizado
 - Quando existe um compromisso claro do cliente em se envolver no processo de desenvolvimento
 - Quando não há muitas regras e regulamentos externos que afetam o software

Equilíbrio entre as Abordagens

- A maioria dos projetos incluem elementos de processos dirigidos a planos e ágeis
- Decidir no equilíbrio depende de uma série de questões técnicas, humanas e organizacionais

1: Especificação Detalhada

- É importante ter uma especificação e um projeto muito detalhados antes de passar para a implementação?
 - Se **SIM**, considere usar abordagem dirigida a planos

2: Entregas + *Feedback*

- É realista uma estratégia de entrega incremental em que você entrega o software aos clientes e rapidamente obtém um *feedback*?
 - Se **SIM**, considere usar métodos ágeis

3: Tamanho do Sistema

- Qual o tamanho do sistema a ser desenvolvido?
 - **Abordagem dirigida a planos**: sistemas grandes, equipes grandes comunicando formalmente
 - **Métodos ágeis**: sistemas pequenos, equipes pequenas comunicando informalmente

4: Tipo do Sistema

- Qual o tipo do sistema a ser desenvolvido?
 - **Abordagem dirigida a planos** podem ser necessárias em sistemas que requerem muita análise antes da implementação
 - Sistemas que operem em tempo real
 - Sistemas com requisitos de temporização complexos

5: Tempo de Vida do Sistema

- Qual o tempo de vida esperado do sistema?
 - **Abordagem dirigida a planos** pode ser adequada em sistemas com longo tempo de vida
 - A documentação de projeto é fundamental para comunicar as intenções originais dos desenvolvedores do sistema para a equipe de suporte
 - A documentação precisa ser atualizada
 - **Métodos ágeis** podem ser adequados em sistemas com curto tempo de vida
 - Será rapidamente descartado ou substituído

6: Apoio Tecnológico

- Quais tecnologias estão disponíveis para apoiar o desenvolvimento do sistema?
 - **Abordagem dirigida a planos** pode ser adequada se existirem poucas ferramentas
 - O que torna a existência de documentação mais importante
 - **Métodos ágeis** geralmente contam com boas ferramentas de análise e visualização do programa para controle do projeto e desenvolvimento
 - Exemplos: testes automáticos, análises de cobertura, ferramentas de *refactoring*

7: Organização da Equipe

- Como é organizada a equipe de desenvolvimento?
 - **Abordagem dirigida a planos** pode ser adequada se a equipe for distribuída ou se parte do desenvolvimento for terceirizado
 - Documentos serão necessários para comunicação entre as equipes

8: Cultura Organizacional

- Existem questões culturais ou organizacionais que podem afetar o desenvolvimento do sistema?
 - Organizações tradicionais de engenharia têm uma cultura de desenvolvimento dirigido a planos
 - Planos são o padrão em engenharia

9: Qualidade da Equipe

- Quão bons são os projetistas e programadores na equipe de desenvolvimento?
 - Métodos ágeis podem exigir mais altos níveis de habilidade

10: Regulamentação Externa

- O sistema está sujeito à regulamentação externa?
 - Se o sistema está sujeito a uma regulamentação externa, então provavelmente pode ser exigida uma documentação detalhada

Considerações Finais

- Abordagens dirigidas a planos e métodos ágeis possuem vantagens e desvantagens
- Diversos fatores devem considerados ao se escolher/combina essas abordagens
 - Técnicos, humanos, organizacionais
- Ambos coexistem na Engenharia de Software

Atividade de Fixação

- 1) O que é desenvolvimento baseado em planos e desenvolvimento ágil? Quais as principais diferenças?
- 2) O que é o manifesto ágil?
- 3) Quais são as situações em que é melhor usar desenvolvimento baseado em planos?
- 4) Quais são os fatores humanos necessários em uma equipe de desenvolvimento ágil?

Referências

- SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson ISBN 9788579361081. (Capítulo 3)
- PRESSMAN, Roger. Engenharia de software. 8. Porto Alegre ISBN 9788580555349. (Capítulo 3)