Tópicos Especiais em TI

Prof. Jailson Brito
Prof. Sergies Oliveira

Aula 1: Métodos Tradicionais e Adaptativos

Avançar →

Objetivos

Identificar as principais diferenças entre os métodos tradicionais e adaptativos de desenvolvimento de software

O mundo como conhecíamos...

O mundo como conhecíamos...

2.1 ESQUEMA DOS ELEMENTOS DAS FUNÇÕES ADMINISTRATIVAS

FUNÇÕES ADMINISTRATIVAS

PLANEJAMENTO	Visualizar o futuro e traçar um plano de ação
ORGANIZAÇÃO	Montar uma estrutura material e humana para realizar a ação
COMANDO	Dirigir e orientar o pessoal
COORDENAÇÃO	Reunir, unificar, harmonizar toda a atividade e esforço
CONTROLE	Cuidar para que tudo se realize conforme planos e ordens

O mundo como conhecemos hoje!

COMO SOBREVIVER A ESTE NOVO MUNDO?

É necessário repensar como as empresas se organizam.

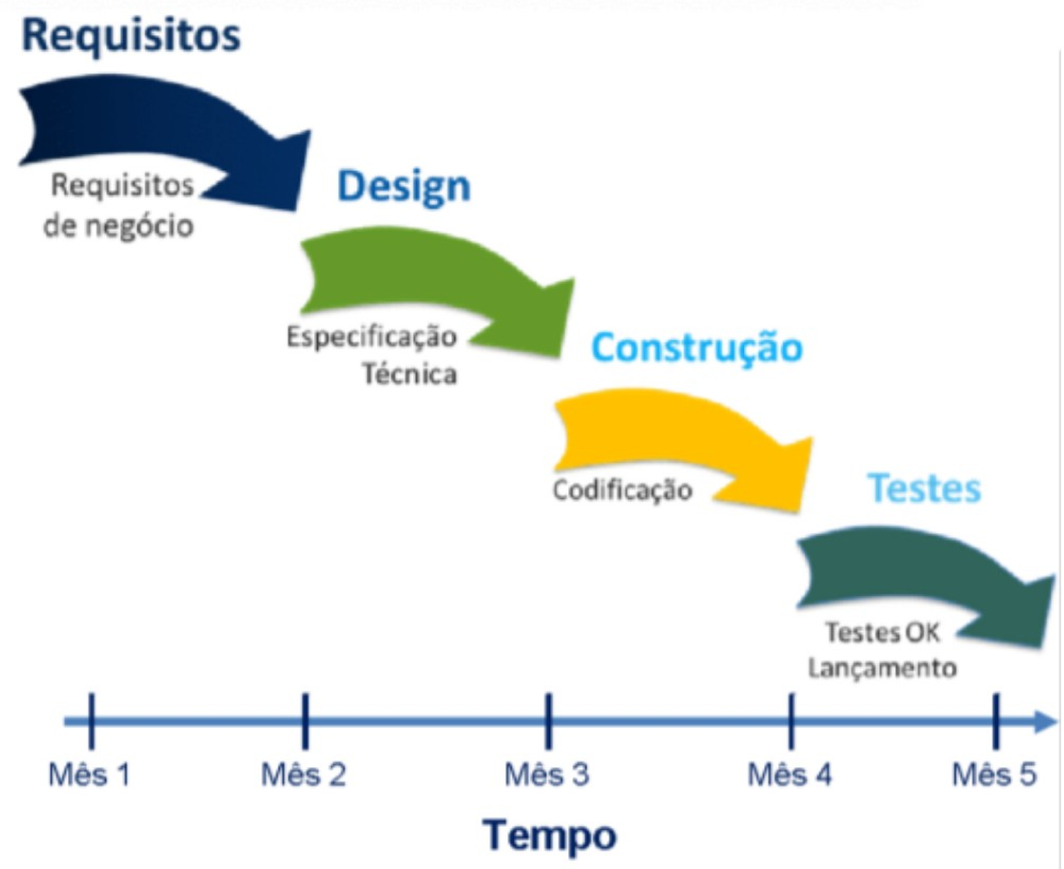
É necessário saber se adaptar.

Não é necessário mudar a cultura. O que é necessário é ter uma cultura de mudança!

Romeo Busarello

Desenvolvimento de software - modelo em cascata

- O modelo em cascata (waterfall em inglês) é o mais tradicional.
- Sugiu na década de 1960.
- Neste modelo, o software é construído por fases.
- A principal característica deste ciclo de vida é que tudo é definido antes do início do desenvolvimento e há uma clara separação entre os processos.
- Esta característica de definir tudo antecipadamente e tentar cumprir é chamado de **preditivo** (predictive).
- Um software funcional somente é uma saída no final da cascata.



Detalhamento antecipado

- O modelo cascata segue o conceito ***Big Design Up Front (BDUF).***
- Os defensores do BDUF acreditam que o tempo gasto revisando exaustivamente a especificação garante a ausência de mudanças críticas na fase de execução.
- O modelo BDUF então faz com que o cliente precise definir tudo o que quer de forma antecipada.
- Como o cliente não terá "chance" de pedir novamente, a tendência é que este peça tudo, até algumas funcionalidades raramente utilizadas.

Detalhamento antecipado

- Há diversos fatores que tornam questionável o uso de BDUF para projetos de software:
  - → Requisitos não são completamente compreendidos no início do projeto
  - → Usuários só sabem exatamente o que querem após verem uma versão inicial do software
  - → Requisitos mudam durante o processo de desenvolvimento
  - → Novas ferramentas e tecnologias tornam a estratégia de desenvolvimento imprevisível.
- Isso se aplica a diversos outros tipos de projetos.
- O BDUF é considerado adequado apenas para projetos estáveis, com pouca ou nenhuma mudança.

O que precisamos fazer?

Precisamos:

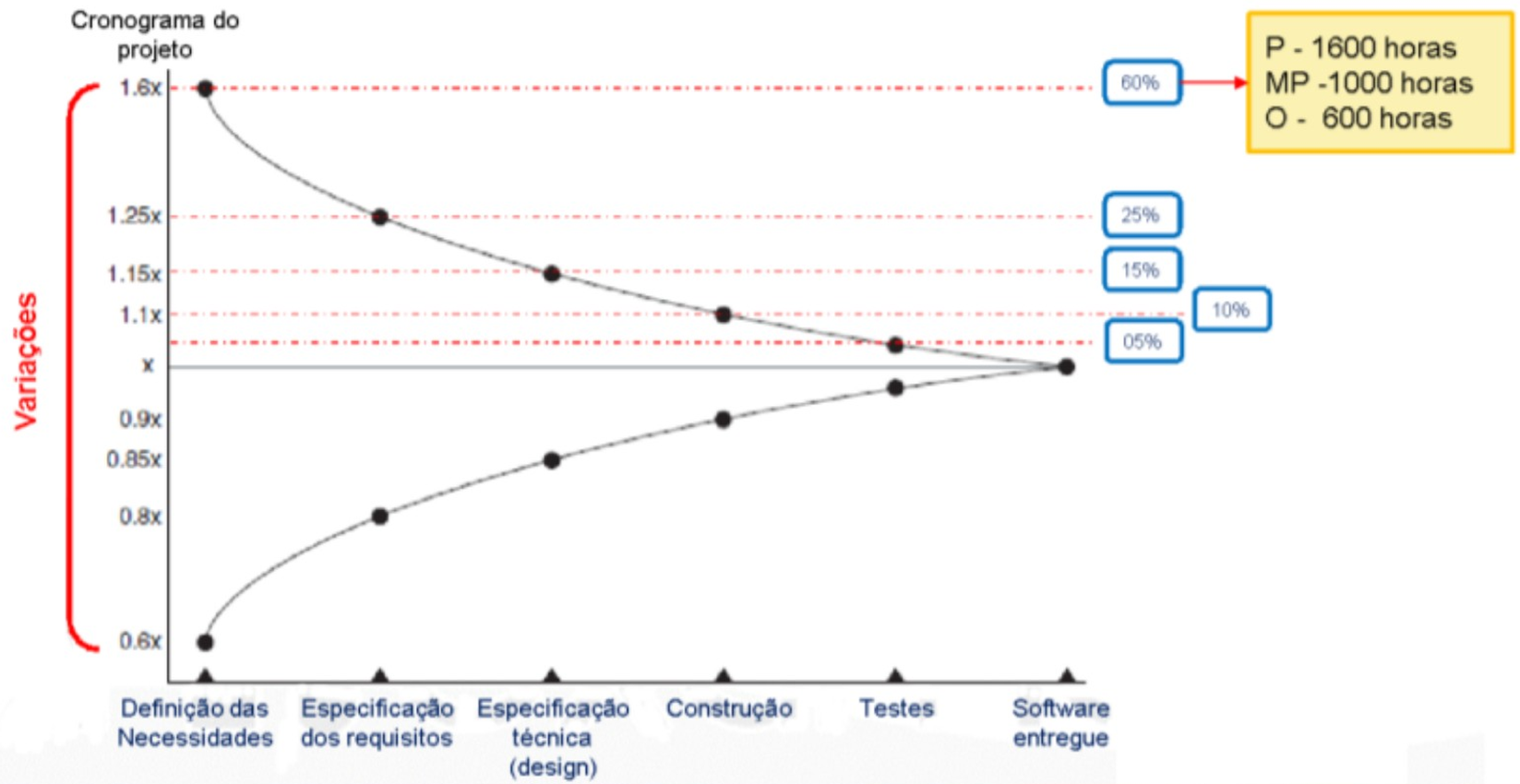
- Estar mais próximos do negócio.
- Ter a consciência de que os requisitos mudam ao longo do tempo e que os usuários só sabem o que realmente querem quando usam o software.
- Demonstrar o software constantemente aos usuários e obter feedbacks de forma contínua.
- Aprender progressivamente sobre o produto (software).
- Ajustar os itens a serem desenvolvidos conforme o entendimento acontecer.
- Conversar mais e escrever menos.
  - ✓ Escrever somente aquilo que realmente é necessário.
- Estar preparados e aceitar as mudanças de forma natural.
  - ✓ Ações preventivas contra mudanças não devem ser realizadas.

Desenvolvimento de software é complexo

- Desenvolvimento de software é um problema complexo.
- Tipicamente envolve muitas pessoas, tecnologia, conhecimento, regras de negócio complexas e exige diversas integrações em diferentes ambientes.
- Esta complexidade é ainda multiplicada pela presença de vários outros fatores:
  - → **Evolução dos requisitos** – um componente que foi desenvolvido "hoje" para poucos usuários pode não funcionar quando possuir muitos usuários "amanhã"
  - → **Desenvolvedores** que trabalham no projeto podem **não estar disponíveis** no futuro e a produtividade pode variar
  - → A **tecnologia** utilizada hoje pode estar **obsoleta** amanhã
  - → O mercado e conceitos de negócio evoluem com muita rapidez exigindo **alterações constantes no software**
- Tais problemas complexos dependentes do tempo são também chamados de problemas adaptativos complexos.
- A quantidade de itens "desconhecidos" no início de um projeto é realmente grande!

Cone da incerteza

- O planejamento tradicional usa técnicas científicas de planejamento para determinar prazo e custo logo no início do projeto.
- O problema é que em um sistema adaptativo complexo, quanto mais cedo, maiores são as incertezas, tornando o plano pouco assertivo.



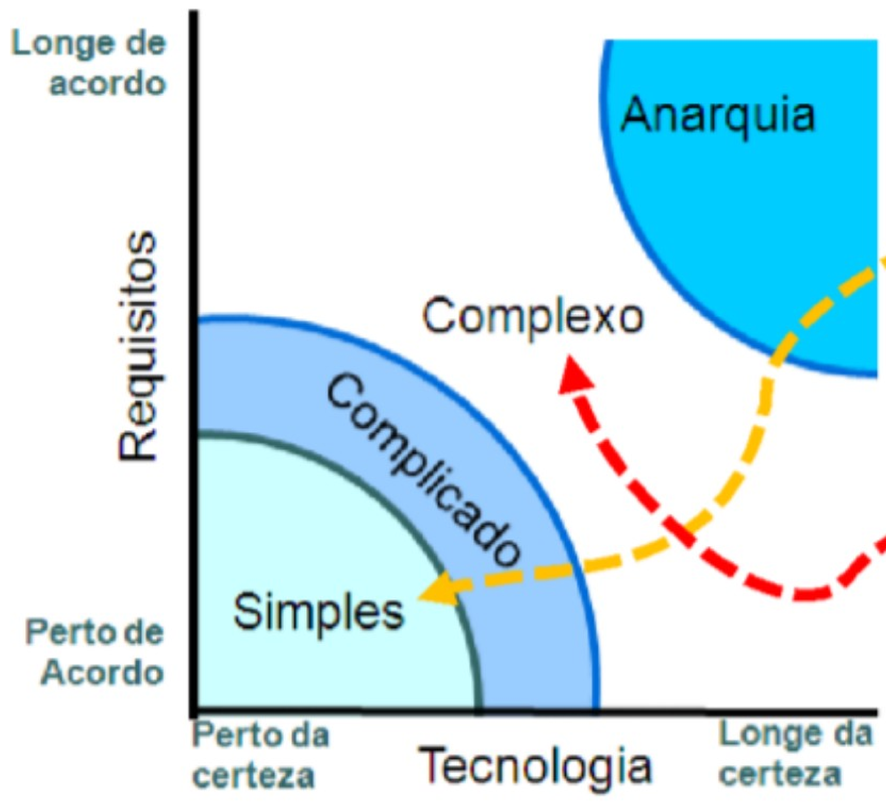
PCC

Planejamento
Comando
Controle

PCA

Propósito
Cultura
Autonomia

Processo empírico



Processos definidos
São processos em que se conhecem todas as variáveis, há poucas ou nenhuma mudança e são repetitivos e previsíveis. Geralmente existe uma documentação aplicada na execução do processo.

Processos empíricos
São aqueles em que não se conhecem todas as variáveis, há mudanças e não são repetitivos nem previsíveis. Geralmente são baseados na experiência e no conhecimento de quem executa o processo.

O empirismo afirma que o conhecimento vem da experiência e devemos tomar decisões com base no que se conhece.

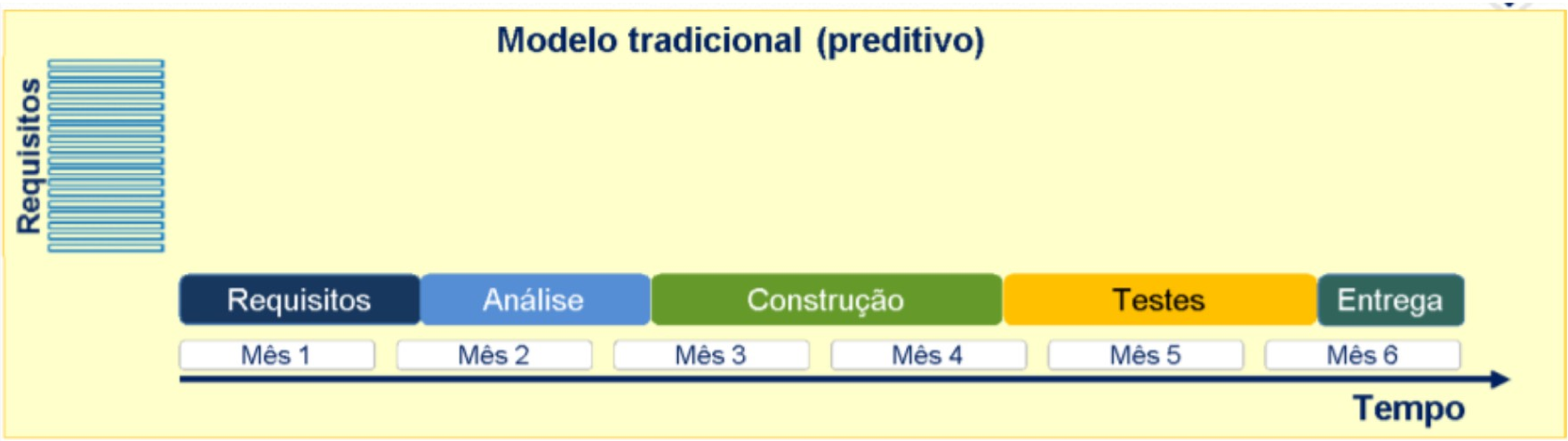
Processo empírico

- A teoria do empirismo é baseada no conceito que problemas complexos são difíceis de serem previstos. O empirismo ajuda as pessoas a tratarem problemas complexos.
- Isso requer dar um passo de cada vez, com a realização de uma pequena quantidade de trabalho para adquirir experiência.

Iterativo e incremental

- Para facilitar as mudanças, no desenvolvimento ágil utilizam-se ciclos adaptativos. No nosso contexto isso significa dizer que desenvolvimentos softwares de forma iterativa e incremental.
- O software é construído por partes (incrementos) e para cada parte executa-se um ciclo (iteração), que inclui todas as atividades necessárias para completar as funcionalidades acordadas para o incremento.

Ciclo adaptativo: iterativo e incremental



Agilidade

Agilidade = ser ágil

+

ser adaptativo

+

usar um framework

Dúvidas?