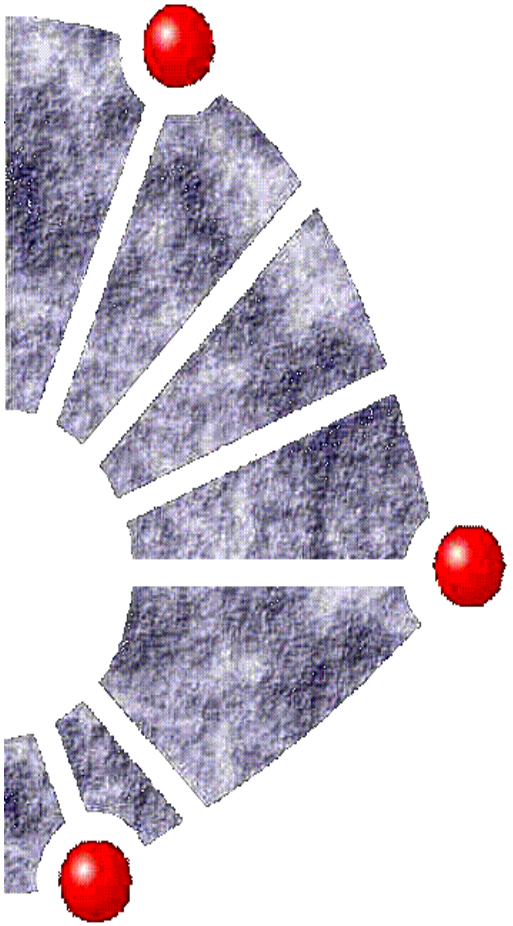
EA976 – Engenharia de Software



AULA 2
Ciclo de Vida de um Software

Ciclo de Vida de um Software

- O que é o ciclo de vida de um software ?
  - Qual a sequência ideal para um ciclo de vida ?
  - Por que essa sequência não funciona na prática ?
  - Podemos assumir que todo software é desenvolvido a partir da "estaca zero" (*from scratch*) ? O que é irreal nessa hipótese ?
  - Qual a diferença entre um **ciclo de vida** e um **modelo de ciclo de vida** ?
- Vocabulário
  - O que é um **artefato** de software ?
  - O que é o **problema do alvo móvel** ?
  - O que é o ***feature creep*** ?
  - O que é uma **falha de regressão** ?

Modelo Codificar e Corrigir

- Como é o modelo Codificar e Corrigir ?
  - Como é medido o progresso neste modelo ?
  - Quais os seus problemas ?
  - Como é o custo de se corrigir um defeito quando:
    - O defeito é encontrado cedo ?
    - O defeito é encontrado tarde ?
- Quais as dificuldades de se fazer a manutenção do sistema sem documentos de especificação e projeto ?
- Em que situações você recomendaria este modelo ?

Modelo Cascata

- O que é o Modelo Cascata ?
  - Como ele pode ser descrito ?
  - Quais os pontos fortes do modelo Cascata ?
  - Qual é o ponto crítico do Modelo Cascata ?
  - Qual o problema em se ter documentação de especificação escrita em formato texto ?
- Qual a relação entre o modelo Cascata e o ciclo Ideal para o desenvolvimento ?
  - Eles são a mesma coisa ?

Modelo Cascata

- Pontos Fortes
  - Há uma especificação
  - Há um projeto
  - Cumprimento de um enfoque rigoroso
- Pontos Críticos
  - Nenhuma fase pode ser considerada terminada até que
    - a documentação para essa fase tenha sido completada
    - Os produtos dessa fase tenham sido aprovados
  - Modificações são desastrosas.

Modelo de Prototipagem Rápida

- Como é o modelo de Prototipagem Rápida ?
  - O que é um Protótipo Rápido ?
  - Qual a diferença entre o modelo de prototipagem rápida e o modelo cascata ?
  - Quais as vantagens que esse modelo tem em relação ao modelo cascata ?

Modelo para Software Aberto

- O que é o modelo para Software Aberto ?
  - Qual a diferença entre o desenvolvimento de um software aberto e um fechado ?
  - Qual a diferença para com o modelo "codificar e corrigir" ?
  - Como são os testes nesse modelo ?
  - Qual a diferença entre o "grupo central" e os "grupos periféricos", neste modelo ?
  - Qual a lógica da máxima "Lance o produto cedo, lance o produto frequentemente" ?
  - Como alguns projetos com esse modelo são tão bem sucedidos ?

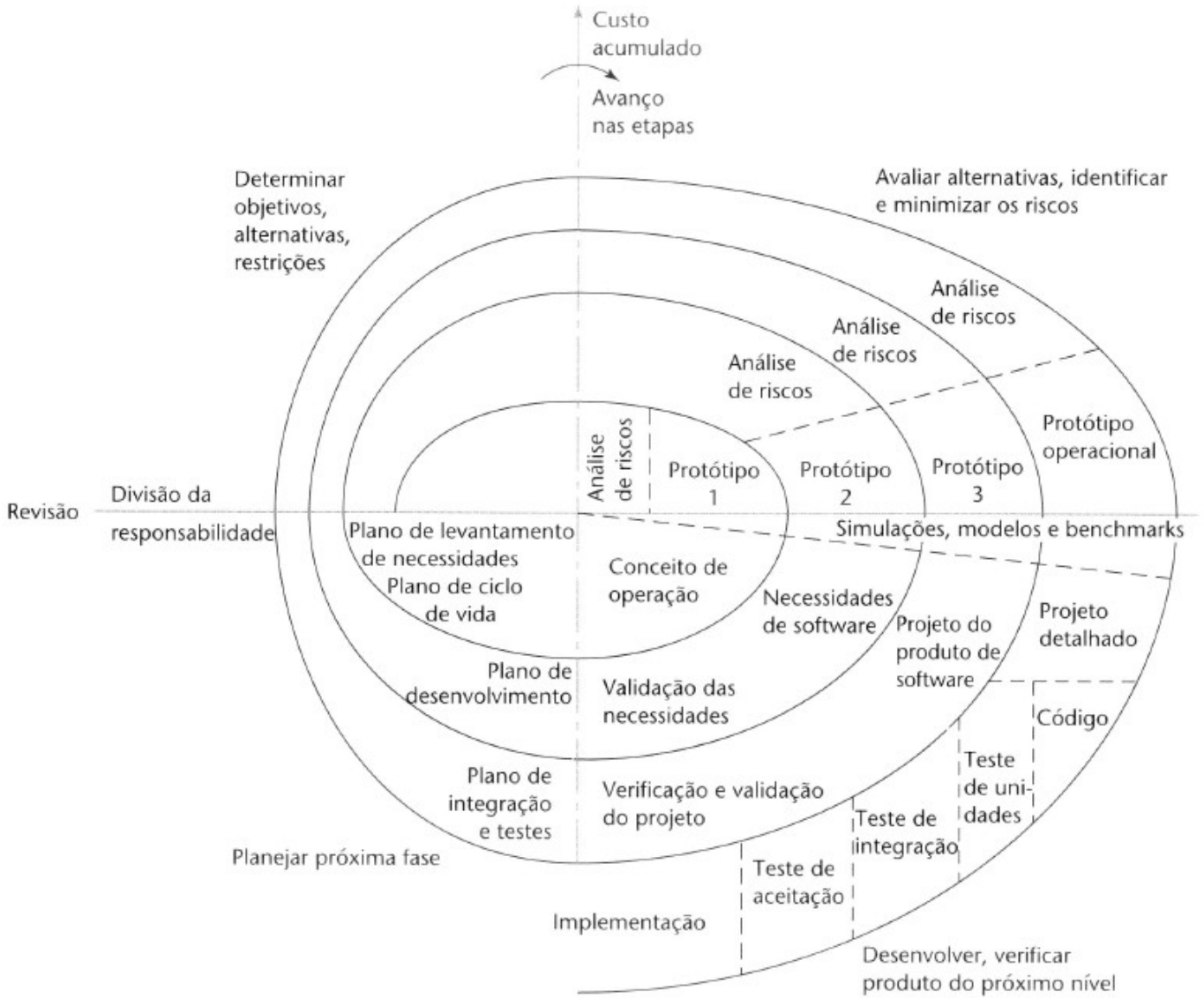
Modelo de Sincronizar e Estabilizar

- Como é o modelo Sincronizar e Estabilizar ?
  - Que companhia adota emblematicamente este modelo ?
  - Como é feito o levantamento de necessidades ?
  - Qual sua relação (similaridades/dissimilaridades) com os outros modelos apresentados ?

Modelo Espiral

- Como é o Modelo Espiral ?
  - Por que a análise de risco é tão importante no modelo espiral ?
  - Todos os riscos podem ser contornados por este modelo ?
  - Quais os pontos positivos do modelo ?
  - Quais as restrições em sua aplicabilidade ?
  - Qual sua aplicação principal ?

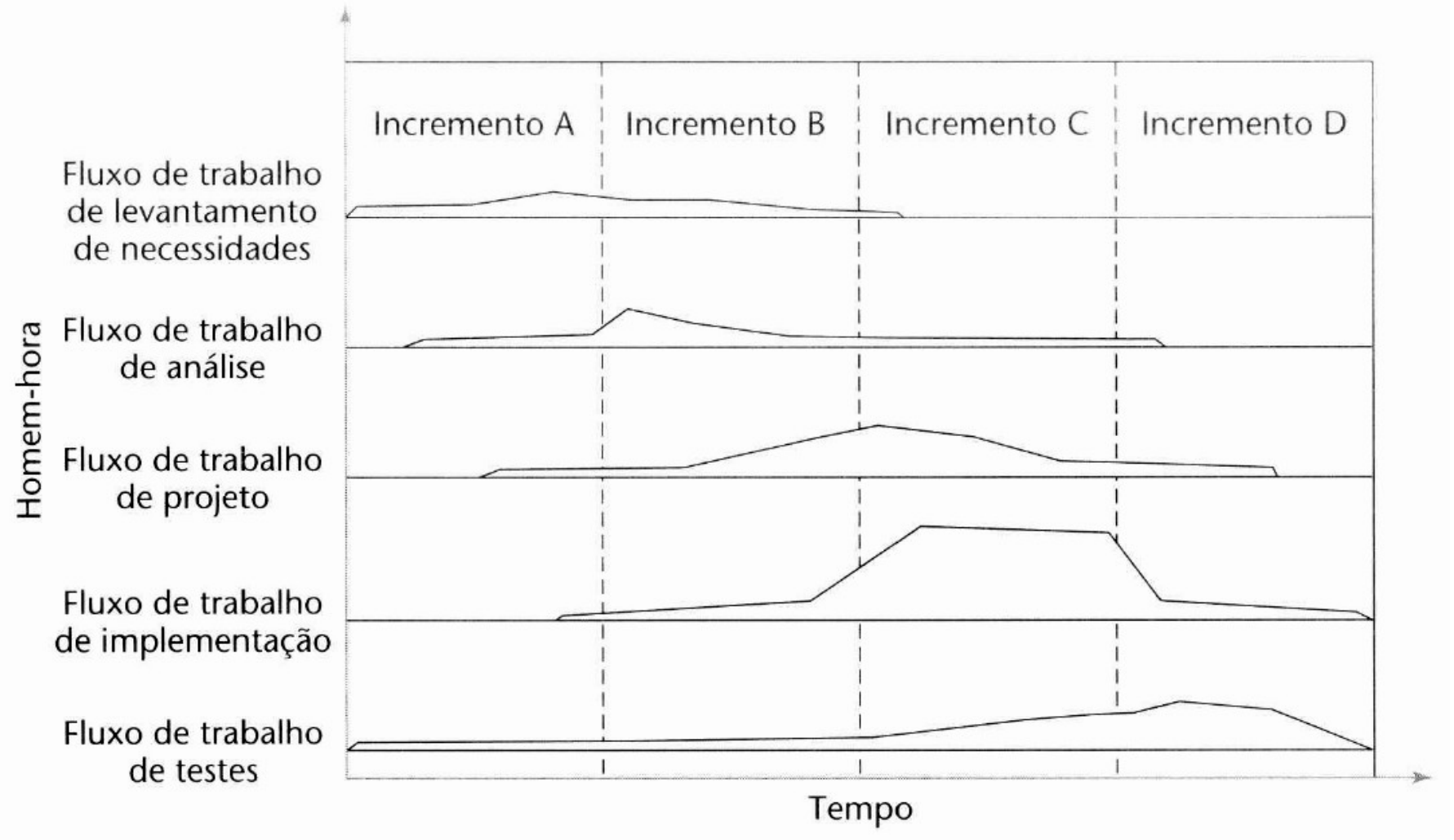
Modelo Espiral



Modelo Iterativo e Incremental

- O que é a Lei de Miller ?
  - Qual sua implicação em engenharia de software ?
  - O que é o **refinamento gradual** ? Qual sua ligação com a lei de Miller ?
- O que é uma Iteração ?
- O que é um Incremento ?
  - Qual a diferença entre iteração e incremento ?
  - Pode existir um ciclo de vida que seja iterativo mas não incremental ?
  - E um que seja incremental e não iterativo ?
  - O que é um **fluxo de trabalho**, em um ciclo iterativo e incremental ?

Modelo Iterativo e Incremental



Modelo Iterativo e Incremental

- Qual a relação entre o modelo Cascata e o conjunto de iterações dentro de um incremento ?
  - Quais as vantagens do modelo iterativo e incremental, com relação ao modelo Cascata e os outros modelos ?
  - Qual a diferença entre o modelo iterativo e incremental e o modelo espiral ?

Modelo Iterativo e Incremental

- Vantagens
  - Diversas oportunidades para verificar se o software está correto
  - Robustez da Arquitetura pode ser determinada relativamente cedo
  - Minimiza riscos
  - Versão funcional do software
  - Evidência empírica de sucesso

Comparação entre Modelos

Modelo de ciclo de vida	Pontos Fortes	Pontos Fracos
Modelo árvore de evolução (Seção 2.2)	Aproxima-se da produção de software no mundo real Equivalente ao modelo iterativo e incremental	
Modelo iterativo e incremental (Seção 2.5)	Aproxima-se da produção de software no mundo real Base do Processo Unificado	
Modelo codificar-e-corrigir (Seção 2.9.1)	Bom para programas curtos que não exigem nenhuma manutenção	Totalmente insatisfatório para programas não triviais
Modelo cascata (Seção 2.9.2)	Método rigoroso Dirigido por documentação	Pode ser que o produto entregue não atenda às necessidades do cliente
Modelo de prototipagem rápida (Seção 2.9.3)	Garante que o produto entregue atenda às necessidades do cliente	Ainda não foi comprovado totalmente
Modelo com software aberto (Seção 2.9.4)	Funcionou perfeitamente bem em um número pequeno de casos	Aplicabilidade limitada Normalmente não funciona
Processos ágeis (Seção 2.9.5)	Funciona bem quando as necessidades do cliente são vagas	Parece funcionar apenas em projetos de pequena escala
Modelo sincronizar-e-estabilizar (Seção 2.9.6)	Necessidades futuras do cliente são atendidas Garante que os componentes possam ser integrados com sucesso	Ainda não foi usado amplamente, a não ser pela Microsoft
Modelo espiral (Seção 2.9.7)	Dirigido por riscos	Pode ser usado apenas para produtos internos de larga escala. Os desenvolvedores têm de ser competentes na análise de riscos e na sua solução.