Engenharia de Software

Processos de Desenvolvimento

Luís Morgado

Instituto Superior de Engenharia de Lisboa
Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Processo de Desenvolvimento

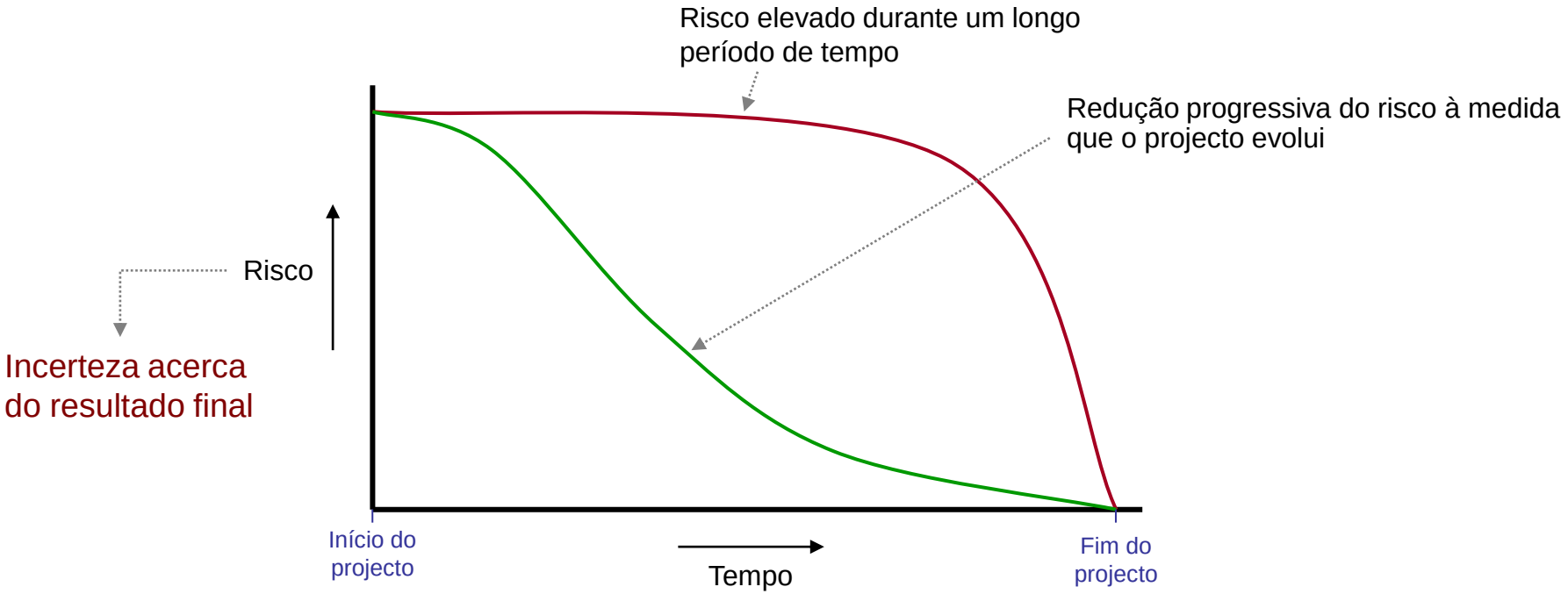
- Desenvolvimento de um sistema
  - Processo de **geração de ordem** (organização do sistema)
  - Requer **conhecimento**
    - Do **problema** a resolver
    - Da **solução** a concretizar
- Será possível obter o conhecimento necessário **todo à partida**?
  - No caso geral a resposta é negativa, devido à complexidade dos problemas e das respectivas soluções
- Será viável construir a solução com conhecimento **incompleto** e eventualmente **incorrecto**?
  - Sem conhecimento completo e correcto da solução a produzir não é possível garantir que cumpre os objectivos definidos
  - O efeito é ineficiência e falta de qualidade no trabalho realizado

Risco de Desenvolvimento

Conhecimento incompleto tem como consequência incerteza, a qual implica elevado risco de desenvolvimento. Para reduzir a incerteza é necessário aumentar o conhecimento, em particular acerca da adequação da solução produzida aos objectivos pretendidos, no entanto, esse conhecimento só é possível de obter quando existe uma versão concreta da solução para poder ser avaliada

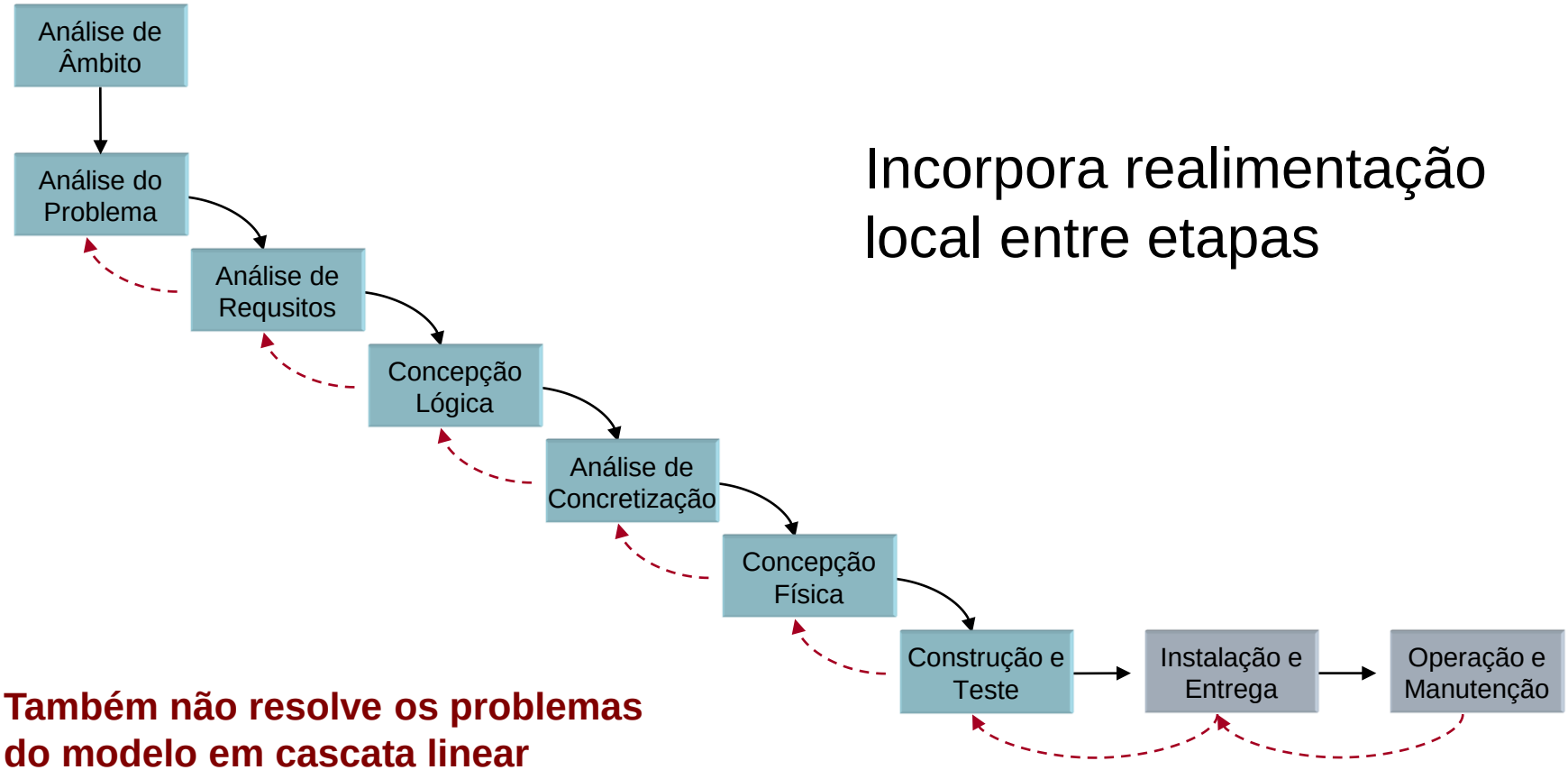
Num processo linear, uma versão concreta da solução só é produzida na fase final do projecto, mantendo o risco de desenvolvimento elevado durante um longo período de tempo

Como reduzir o risco à medida que o projecto evolui?



Redução de incerteza ← **Aumento do conhecimento**

Modelo em Cascata Revisto



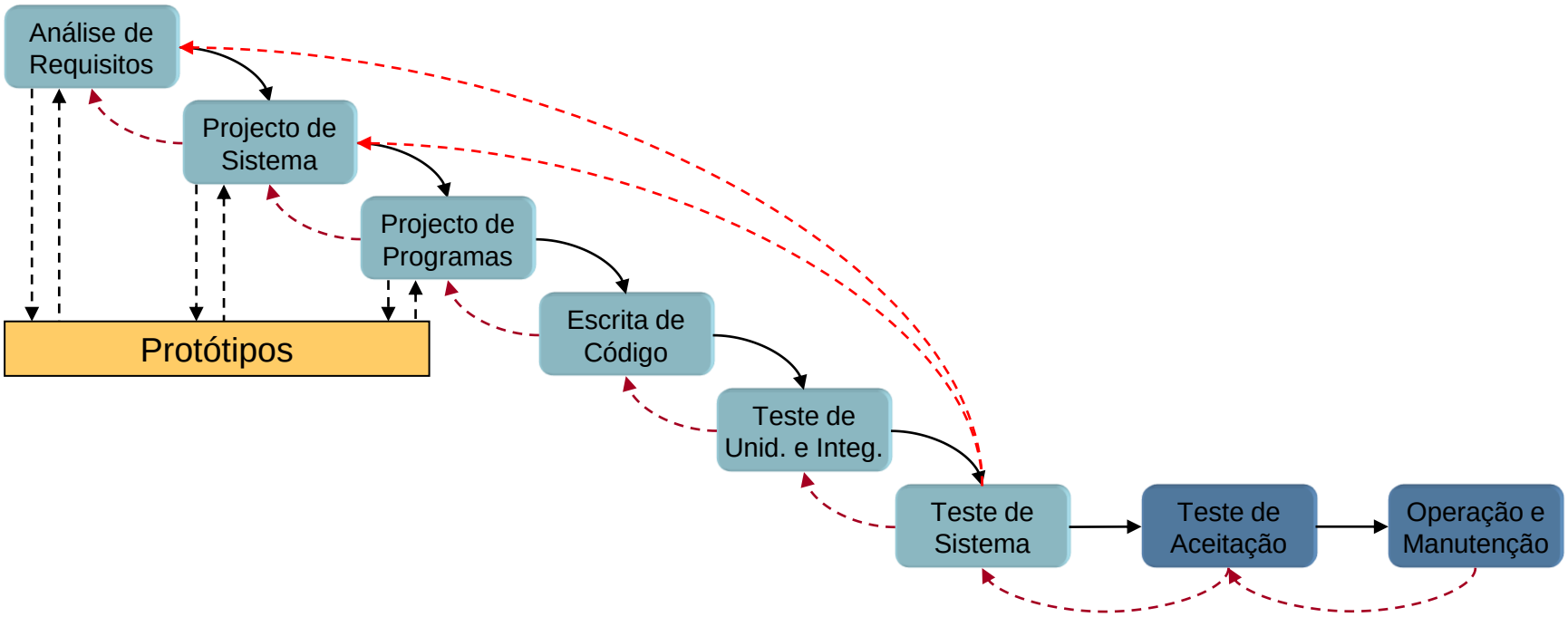
Incorpora realimentação local entre etapas

Também não resolve os problemas do modelo em cascata linear

Na prática, a realimentação não é local, é global!

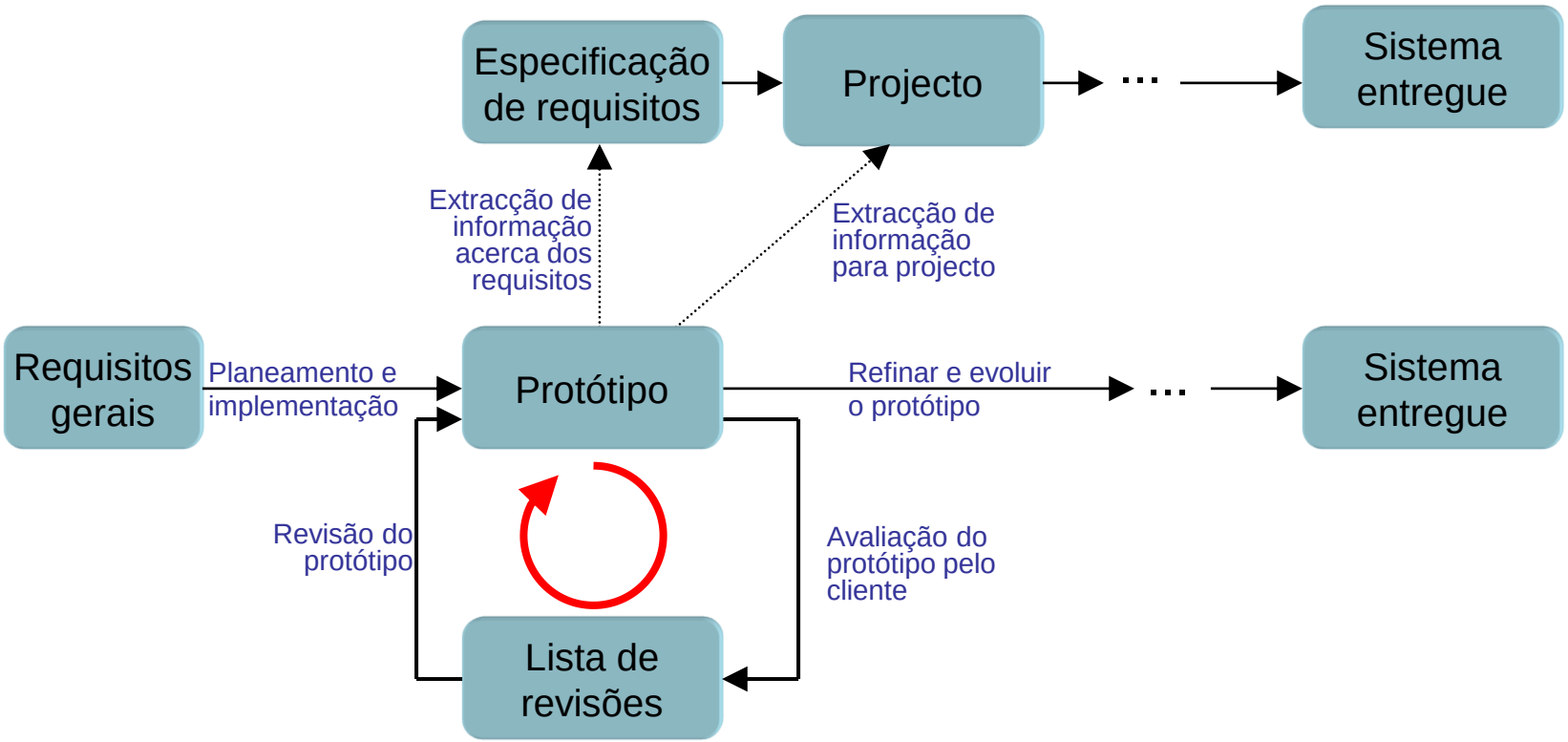
Realização de Protótipos

- Realização de protótipos intercalares
- O objectivo é tentar reduzir o risco de desenvolvimento
  - Envolvimento dos utilizadores
  - Verificação de viabilidade
  - **Convergência progressiva** para a solução pretendida



Modelo de Prototipagem

Baseia-se na construção de **versões simplificadas**, ou de partes **limitadas** do sistema, as quais podem ser analisadas **previamente** antes da produção de uma versão final



Modelo de Prototipagem

- **Objectivo**:
  - **Identificar requisitos** do problema
  - **Reduzir a incerteza** de desenvolvimento (e o risco associado)
- **Os protótipos podem variar em**:
  - **Dimensão**: limitados *vs*. abrangentes
  - **Funcionalidade**: funcionalidades limitadas *vs*. conjunto alargado de funcionalidades
- **O que fazer com o protótipo**:
  - **Refiná-lo** até a obtenção do produto final
  - **Tomá-lo como ponto de partida** para o desenvolvimento convencional
- **Alguns problemas**:
  - O cliente vê o que **parece ser uma versão operacional** do software, desconhecendo que a estrutura interna é em muitos aspectos **provisória**
  - São feitos **compromissos iniciais** no sentido de obter um protótipo **rapidamente**, o que pode levar a **opções de implementação** que mais tarde se revelam **problemáticas**

Modelo Linear

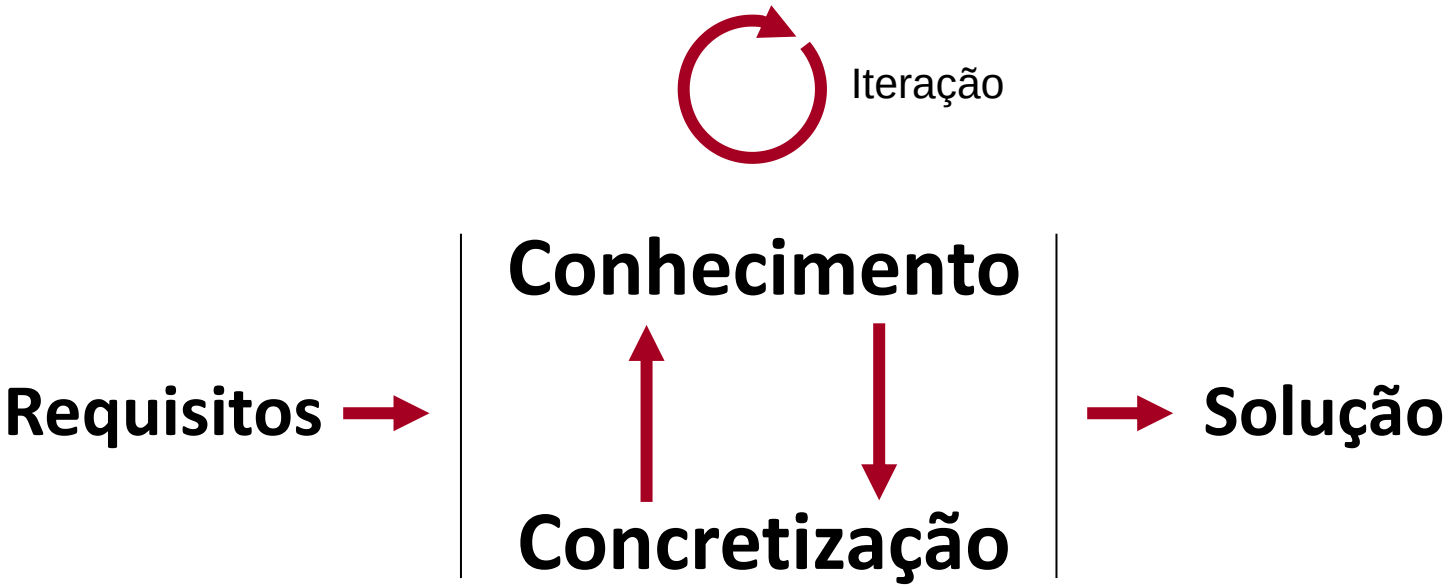
- Ênfase no **planeamento** antes da construção
- Escalonamento **linear** das tarefas
  - Facilidade de gestão
  - Facilidade de comunicação
- **Separação** clara entre actividades
  - Concepção / Construção
- **Assume-se** que o desenvolvimento de software é um processo **linear**, **previsível**
- **Mudança** é considerada um **factor perturbador**

Características do Desenvolvimento de Software

- **Volatilidade**
  - Produção e alteração de software não implica processos industriais de grande inércia
  - Muito depende da **criatividade** de quem o realiza
- **Esforço** reside essencialmente na **concepção**
  - Requer criatividade e conhecimento
- Processos criativos
  - Não lineares
  - Difícil previsibilidade
  - Difícil planeamento

Processo de Desenvolvimento

As actividades de um processo de desenvolvimento podem encadear-se ciclicamente de modo a aumentar progressivamente o conhecimento da solução a produzir para atingir o objectivo definido, através de diferentes iterações de concretização da solução, num *processo iterativo guiado por conhecimento*, onde o conhecimento obtido em cada iteração alimenta a iteração seguinte

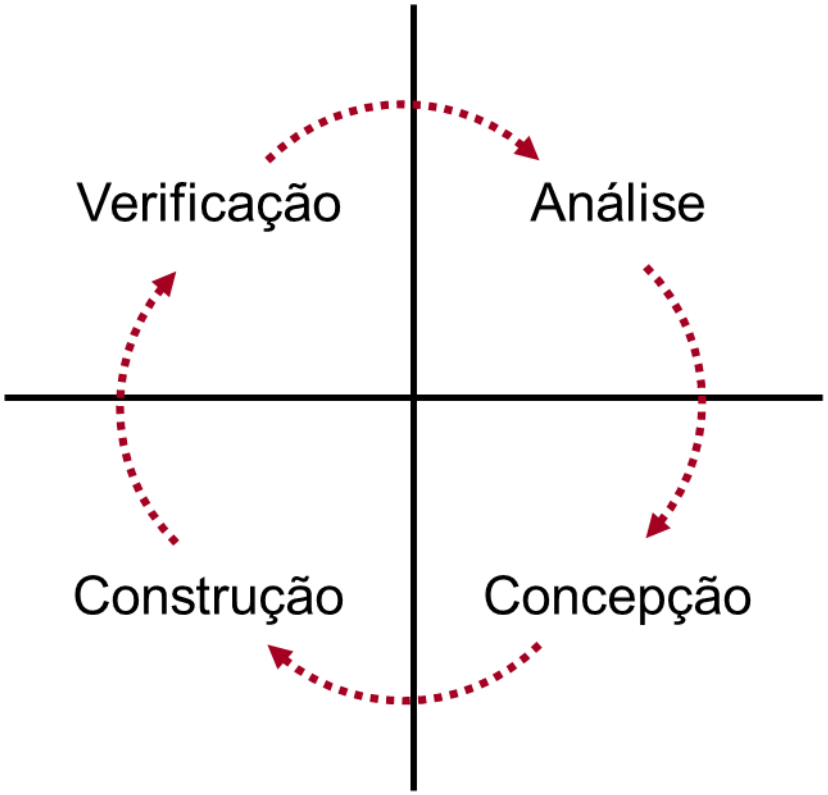


Modelo Adaptativo

- Abordagem **alternativa** ao **modelo linear** para desenvolvimento de software
- **Aceitação** do **imprevisto** e da **mudança**
- **Adaptação** à mudança
- Ênfase nas **pessoas** e na sua **capacidade de resolução de problemas**

Modelo de Desenvolvimento Cíclico

As diferentes fases e actividades de desenvolvimento são realizadas em diferentes ciclos (iterações), com aumento sucessivo do conhecimento obtido e da funcionalidade das versões produzidas, até à convergência para a versão final



Desenvolvimento Cíclico

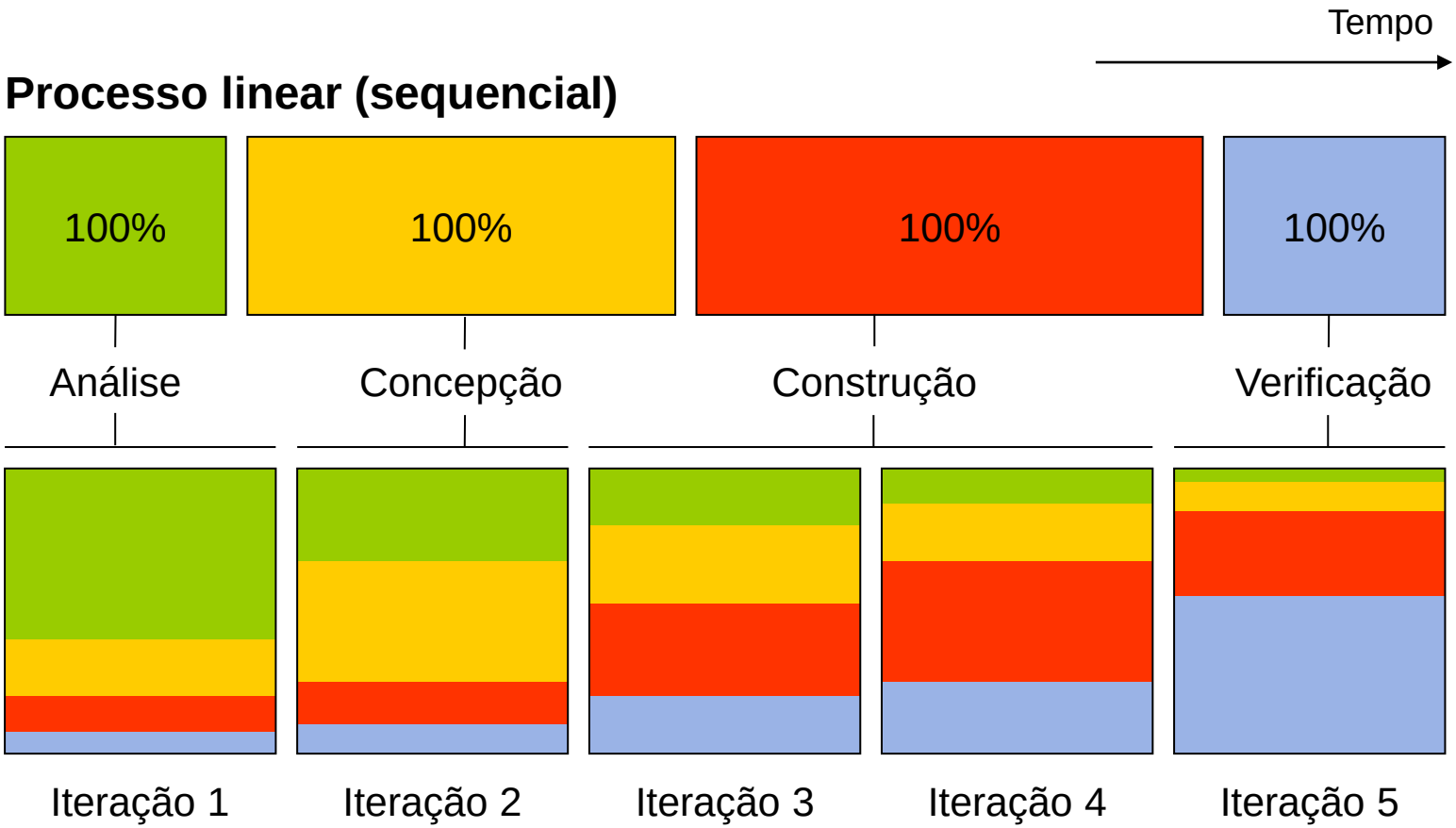
- **Organização geral**
  - O desenvolvimento cíclico é organizado com base nos seguintes conceitos principais:
    - **Fase**
      - O conceito de fase no modelo de desenvolvimento cíclico refere-se às diferentes etapas de desenvolvimento de um projecto, tendo cada fase um âmbito de actividade específico, nomeadamente, *análise*, *concepção*, *construção*, *verificação*
    - **Iteração**
      - Conjunto de *actividades* que têm como propósito a produção de uma versão incremental da solução
    - **Actividade**
      - Conjunto de acções relacionadas que têm um propósito específico no âmbito da solução a desenvolver, por exemplo, *análise de requisitos*, *concepção da arquitectura*, *construção do código*, *verificação da solução*

Desenvolvimento Cíclico

- Fase
- Actividade

Não são equivalentes

Cores representam tipos de actividade diferentes



Processo iterativo

As diferentes actividades são realizadas em cada iteração com ênfases variáveis

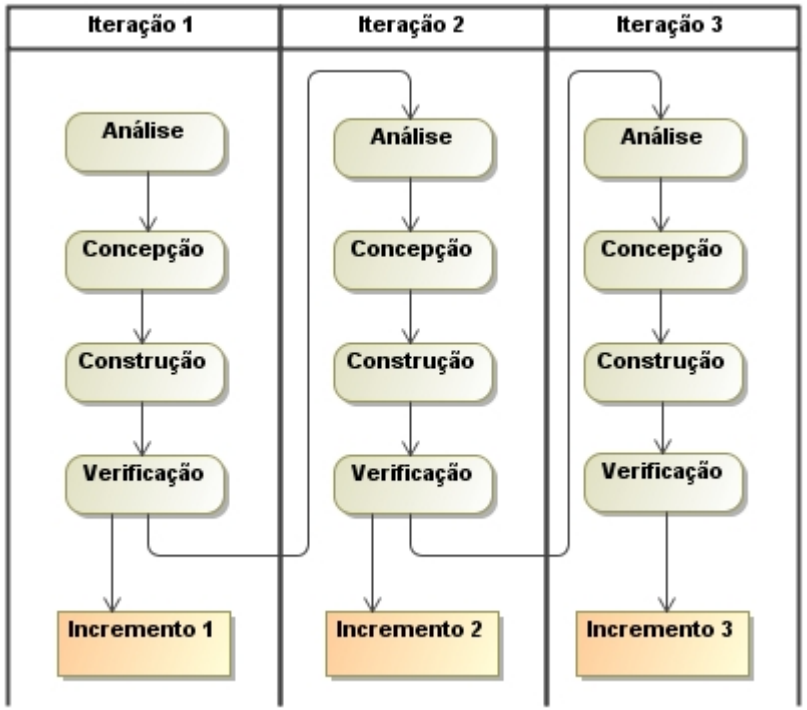
Desenvolvimento Cíclico

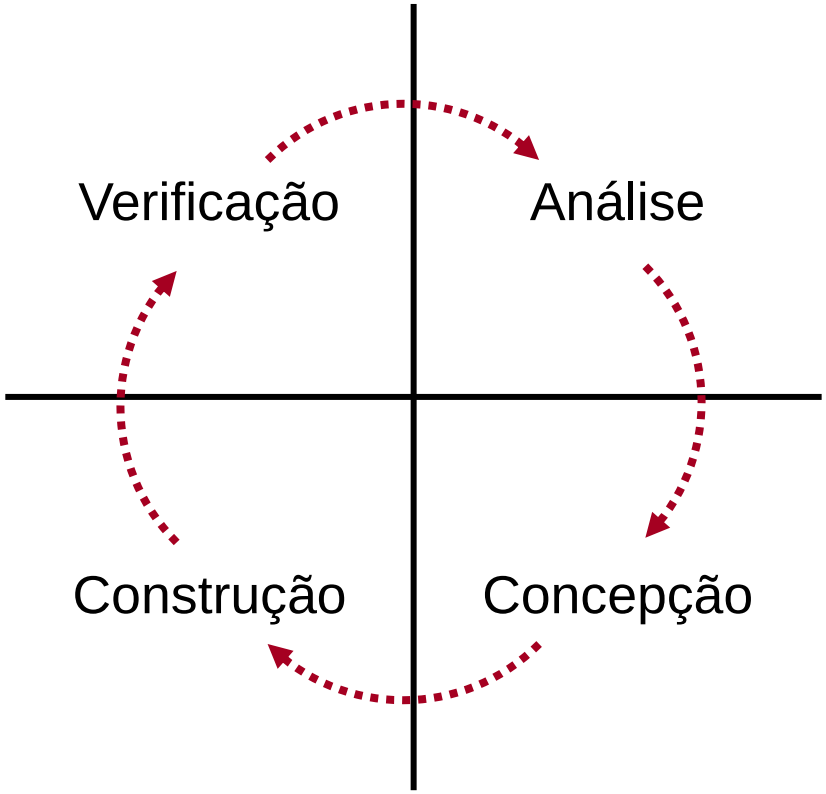
Vantagens

- Envolvimento dos utilizadores pode começar numa fase precoce
  - Versões incrementais da solução
  - Permite obter informação dos utilizadores sobre a solução produzida
- Versões frequentes permitem detectar e resolver rapidamente problemas não antecipados
- Permite a focagem em diferentes áreas de especialização consoante a versão

Desenvolvimento Incremental Iterativo

O *desenvolvimento incremental iterativo* é um modelo de desenvolvimento cíclico orientado para a produção de versões incrementais da solução, com a repetição das mesmas etapas, organizadas em iterações, até que o resultado pretendido seja obtido





Quatro actividades principais:

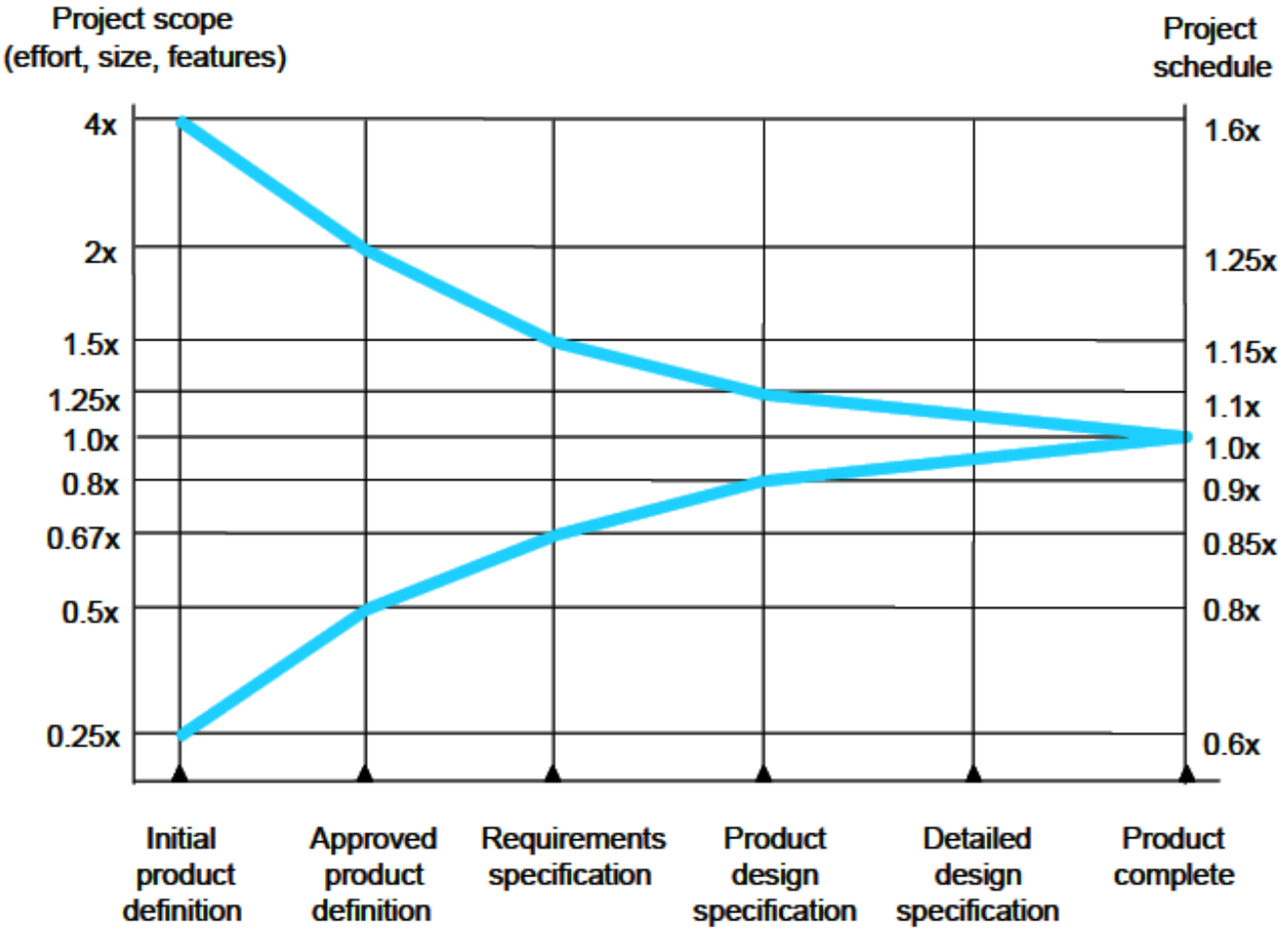
- Análise
- Concepção
- Construção
- Verificação

Foco:

- **Produção de Valor**
- **Redução de Risco**

Risco de Desenvolvimento

O desenvolvimento incremental iterativo possibilita a obtenção de conhecimento de forma incremental ao longo da realização de um projecto, reduzindo progressivamente a incerteza e, consequentemente, o risco, até à concretização da versão final da solução

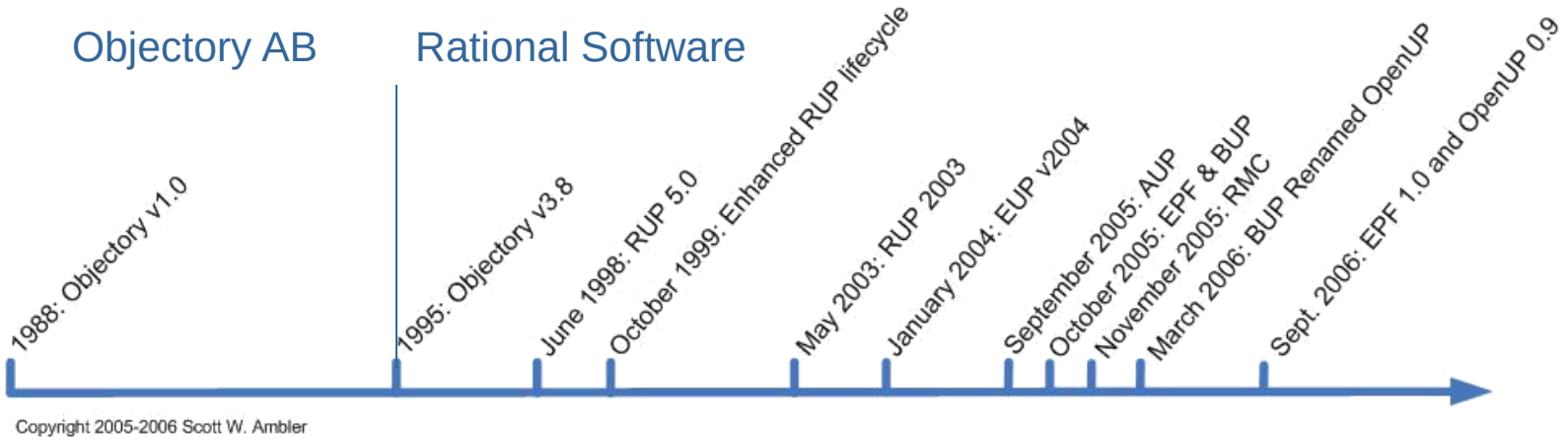


[Boehm, 2000]

Desenvolvimento Incremental Iterativo

Exemplo: *Rational Unified Process* (RUP)

- Resultou da fusão de vários métodos de desenvolvimento de software
  - OMT (James Rumbaugh, 1991)
  - OOSE (Ivar Jacobson, 1992)
  - Booch Method (Grady Booch, 1993)
- Criado pela Rational Software
- Meta-processo que pode ser concretizado de diferentes formas
- Serviu de base para processos mais recentes



Desenvolvimento Incremental Iterativo

Rational Unified Process (RUP)

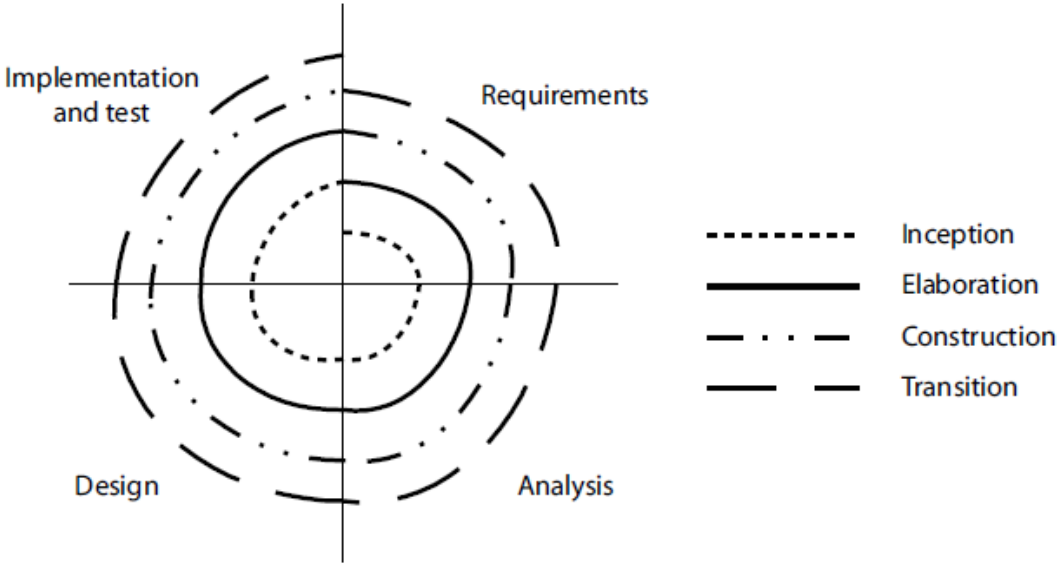
- Principais características
  - **Incremental** e **iterativo**
    - Desenvolvimento **progressivo**
    - Desenvolvimento **cíclico**
  - Guiado por **casos de utilização**
  - Centrado na **arquitectura**
  - Focado na **redução do risco**
    - Risco de desenvolvimento **explicitamente** reconhecido
    - Ênfase nos aspectos **desconhecidos** da solução a desenvolver (redução da incerteza)

Desenvolvimento Incremental Iterativo

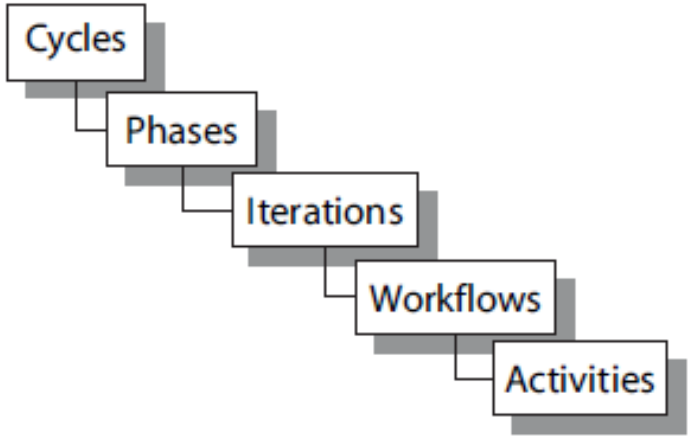
Rational Unified Process (RUP)

Principais aspectos de organização do processo

Desenvolvimento cíclico



Principais conceitos

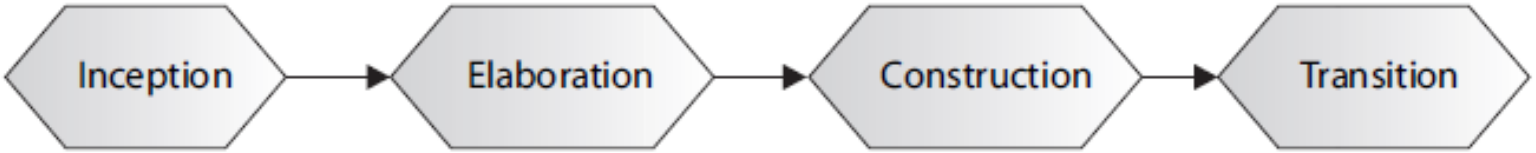


[Hunt, 2003]

Desenvolvimento Incremental Iterativo

Rational Unified Process (RUP)

Fases de desenvolvimento



Inception: Understand what to build
- Vision, high-level requirements, business case
- Not detailed requirements

Elaboration: Understand how to build it
- Baseline architecture, most requirements detailed
- Not detailed design

Construction: Build the product
- Working product, system test complete

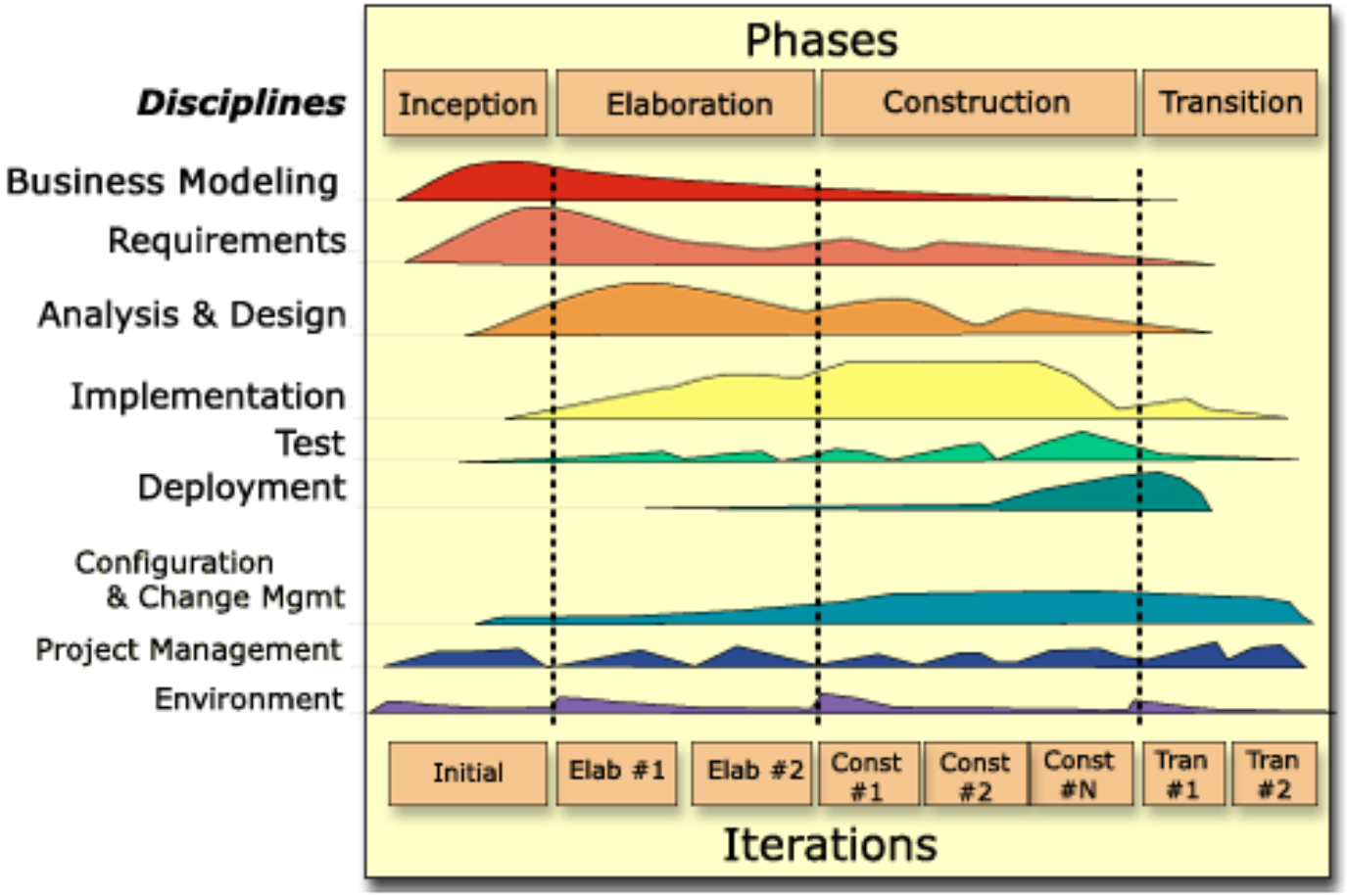
Transition: Validate solution
- Stakeholder acceptance

[Hunt, 2003]

Desenvolvimento Incremental Iterativo

Rational Unified Process (RUP)

Iterações incrementais

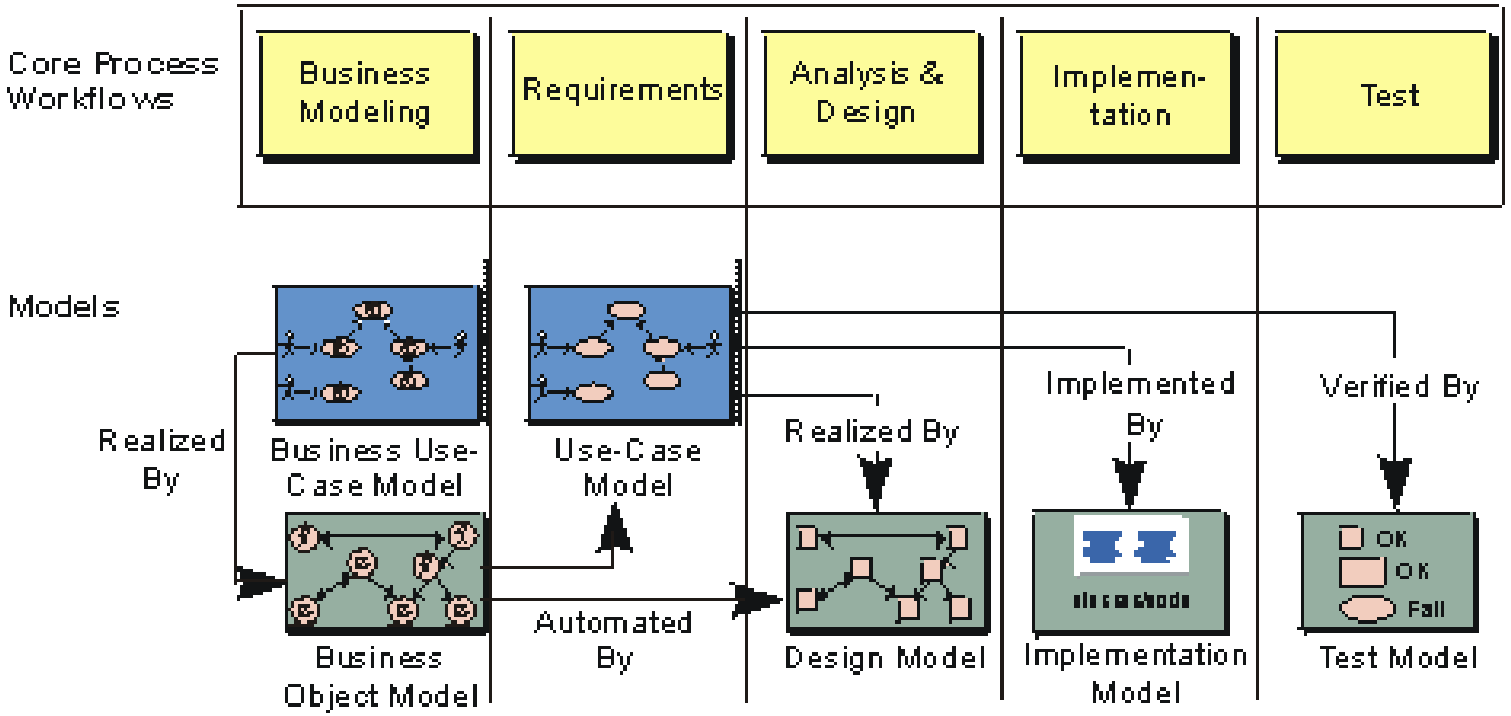


[Rational, 2003]

Desenvolvimento Incremental Iterativo

Rational Unified Process (RUP)

Fluxos de Trabalho (*Workflows*) e Modelos



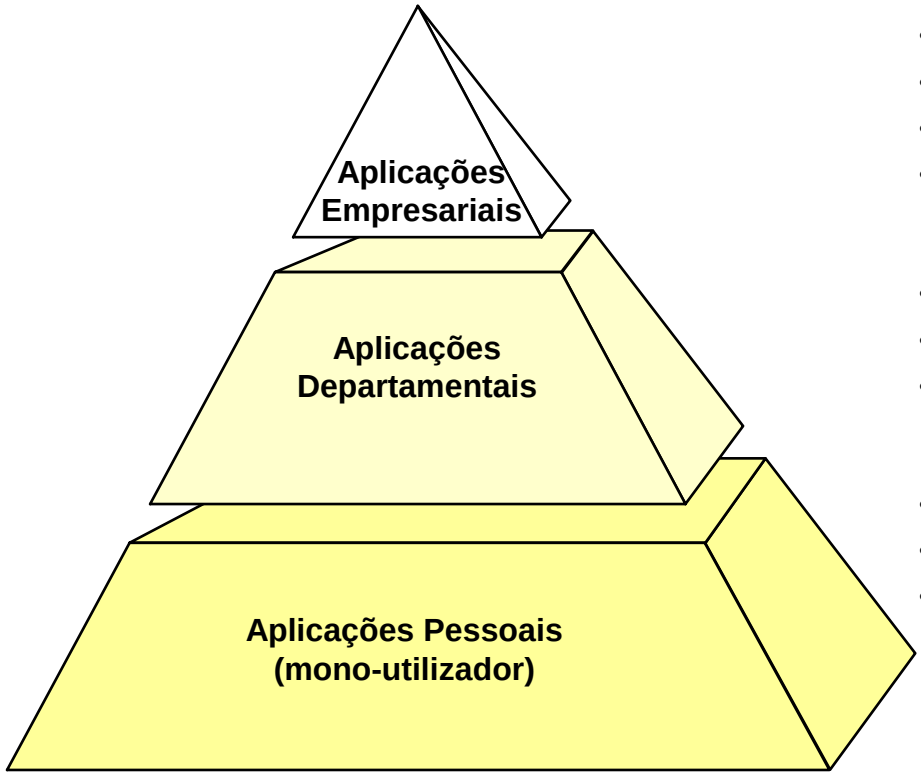
[Rational, 2003]

Que tipo de processo de desenvolvimento utilizar?

Apesar do desenvolvimento incremental iterativo apresentar características relevantes para lidar com aspectos como a incerteza e a redução do tempo de desenvolvimento, também pode apresentar limitações, nomeadamente, quando os sistemas a produzir disponibilizam funcionalidades críticas para os seus utilizadores, individuais ou organizações, na produção dos quais o foco não é tanto a rapidez mas o controlo do desenvolvimento

Desenvolvimento controlado

Desenvolvimento rápido



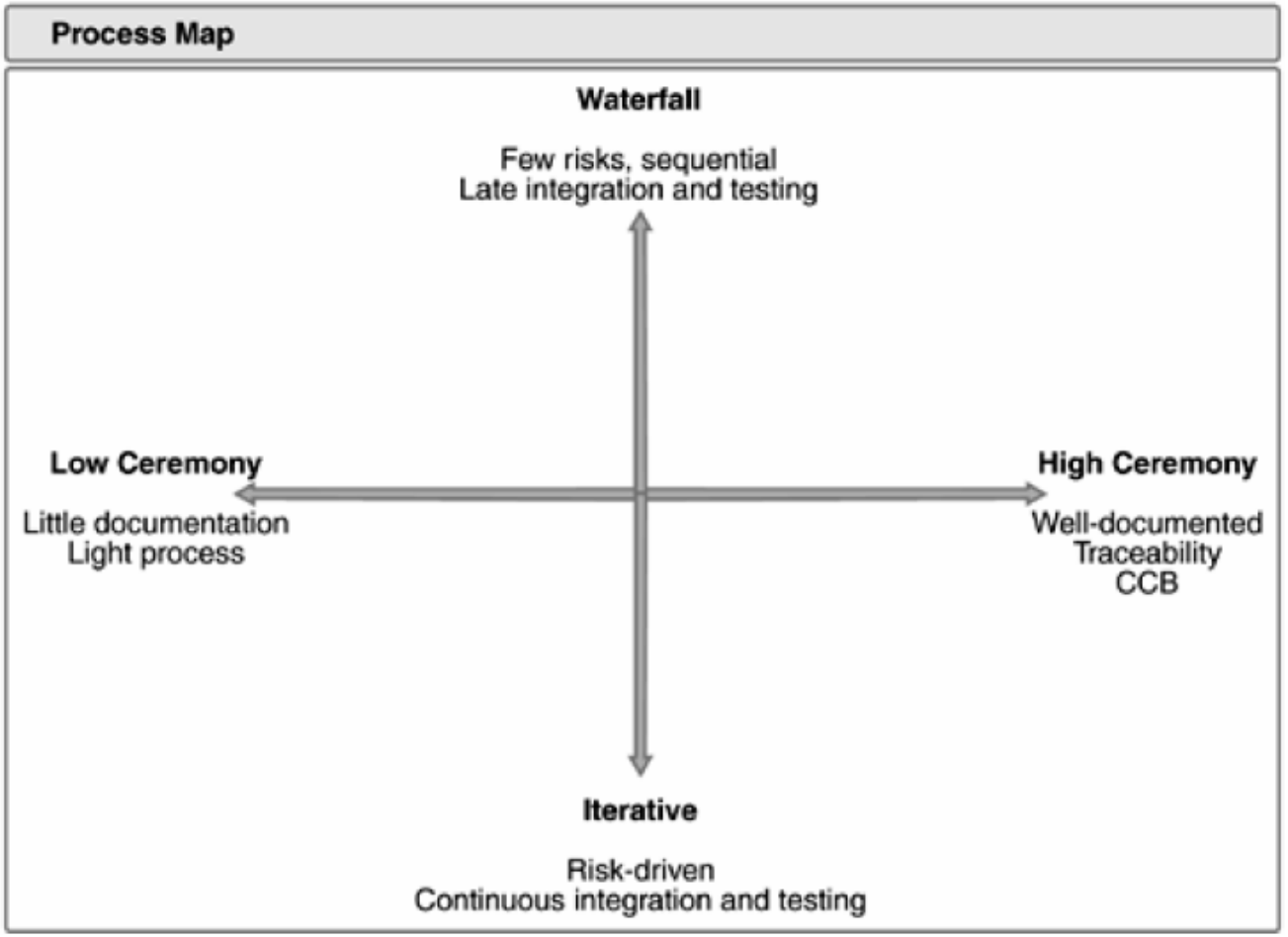
- Sistemas críticos ("*mission critical*")
- Número elevado de utilizadores
- Risco médio/alto
- Multi-plataforma

- Âmbito limitado
- Risco baixo/médio
- Uni/multi-plataforma

- Custo baixo
- Risco baixo
- Uni-plataforma

Que tipo de processo de desenvolvimento utilizar?

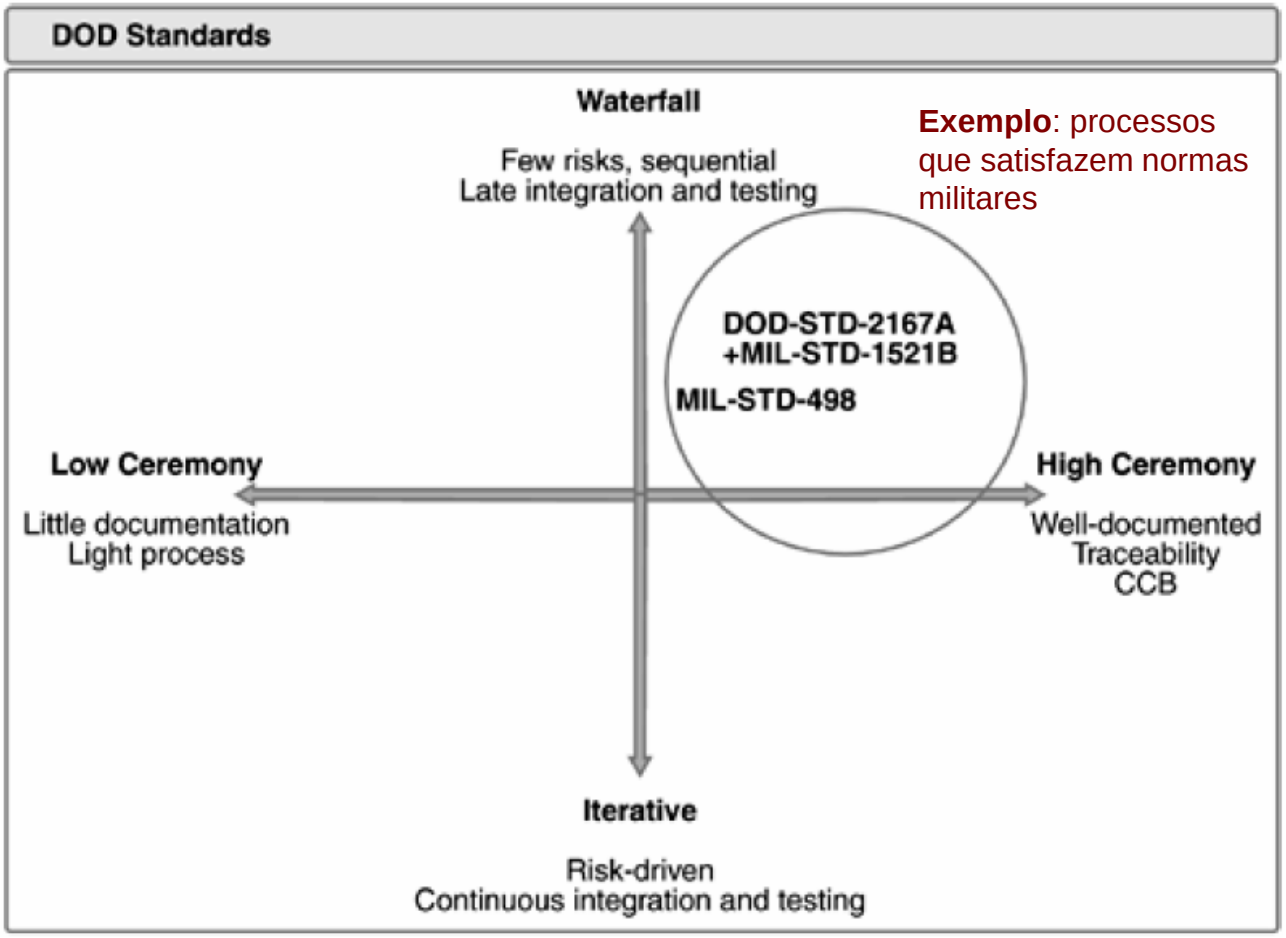
Para identificar as características do processo de desenvolvimento mais adequadas a diferentes tipos de aplicações, é possível definir duas dimensões principais de avaliação: o ***nível de formalidade*** (rigor do processo e documentação produzida) e o **tipo de processo, *cascata ou iterativo***



[Kruchten, 2003]

Que tipo de processo de desenvolvimento utilizar?

Para o desenvolvimento de sistemas de elevada complexidade ou com requisitos estritos, como sistemas críticos, por exemplo sistemas militares, os níveis de formalidade devem ser altos e a evolução das várias fases deve ser mais controlada, ou seja, com uma organização mais próxima da linear, em cascata

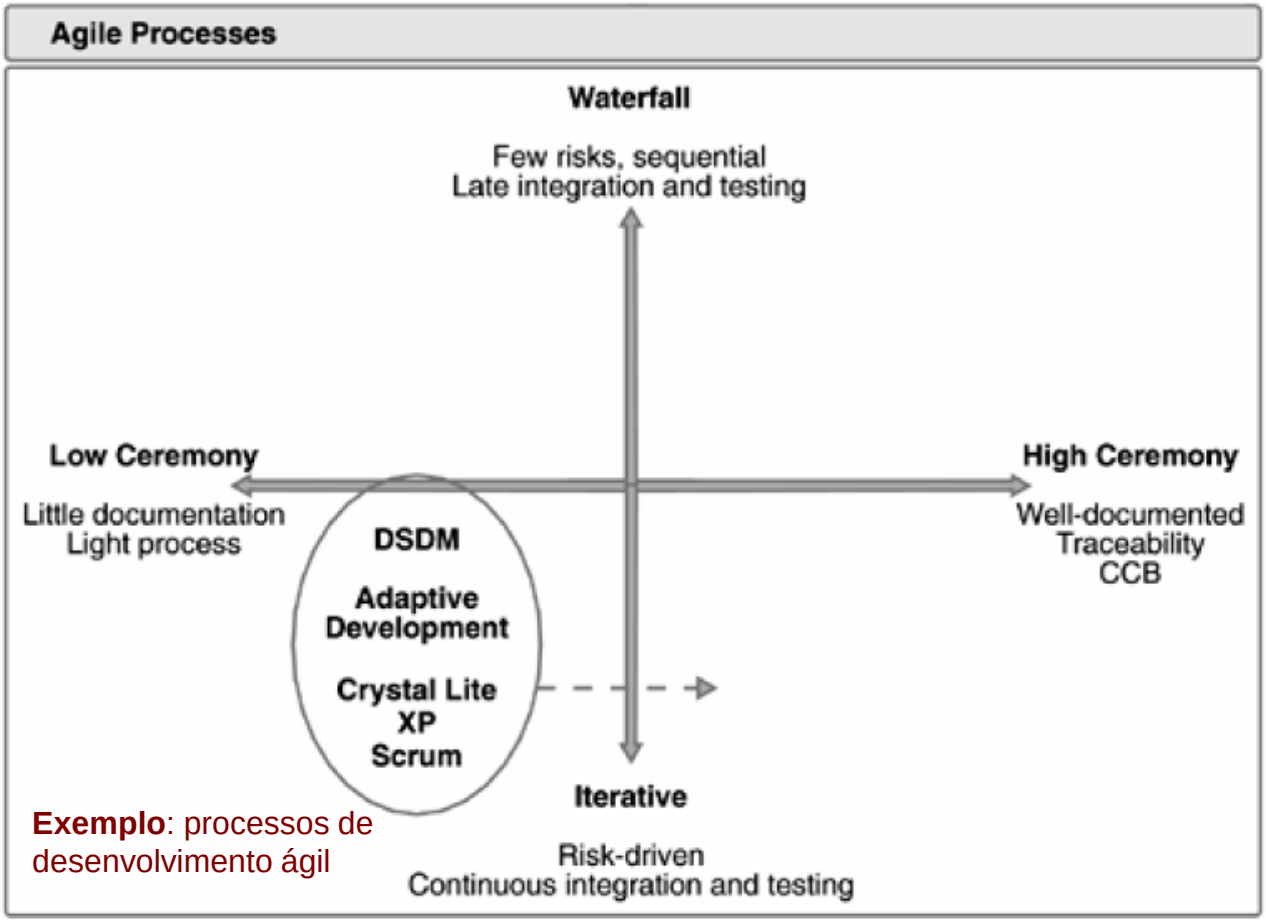


Exemplo: processos que satisfazem normas militares

[Kruchten, 2003]

Que tipo de processo de desenvolvimento utilizar?

Para o desenvolvimento de sistemas em contextos de grande incerteza e em que o tempo de desenvolvimento é importante, níveis de formalidade mais baixos e um carácter incremental e iterativo são mais adequados, podendo o nível de formalidade aumentar em função da complexidade e do nível de criticidade da solução a produzir



Exemplo: processos de desenvolvimento ágil

[Kruchten, 2003]

Bibliografia

[Pressman, 2003]
R. Pressman, *Software Engineering: a Practitioner's Approach,* McGraw-Hill, 2003.

[Weaver, 1948]
W. Weaver, *Science and Complexity*, American Scientist, 36: 536, 1948.

[Hitchins, 1992]
D. Hitchins, *Putting Systems to Work*, John Wiley, 1992.

[Boehm, 2000]
B. Boehm, *Software Cost Estimation with COCOMO II*, 2000.

[DAUP, 2001]
Systems Engineering Fundamentals. Defense Acquisition University Press, 2001