

Processos de software

Maria Clara Silveira

Processo Desenvolvimento Software

O processo de software pode ser definido como um conjunto de atividades envolvidas na produção de software. Estas atividades estão relacionadas entre si num processo coerente e controlado.

Atividades genéricas (Ian Sommerville):

- **Especificação** – o que o sistema deve fazer
- **Design e implementação** – definição da arquitetura e produção do sistema de software
- **Validação** – verificar e testar se o software está de acordo com as necessidades do cliente
- **Evolução** – manutenção do software

Processo Desenvolvimento Software

Criação /alteração do sistema com base nos *requisitos*



Processos para desenvolver Software

- **Ciclo de vida clássico**
- **Prototipagem**
- **Modelo em espiral**

Ciclo de Vida Clássico ou Cascata

- **Decisão**
- **Análise de requisitos**
- **Concepção (Design)**
- **Implementação**
- **Instalação e formação**
- **Manutenção**

Desenvolvimento com utilização experimental

- **Mostra aos utilizadores o que faz a aplicação**
- **Mecanismo de definição de requisitos**
- **Forma de simular rapidamente a aplicação**
- **Garantir nas diferentes fases de desenvolvimento a coerência da informação**

Modelo em Espiral

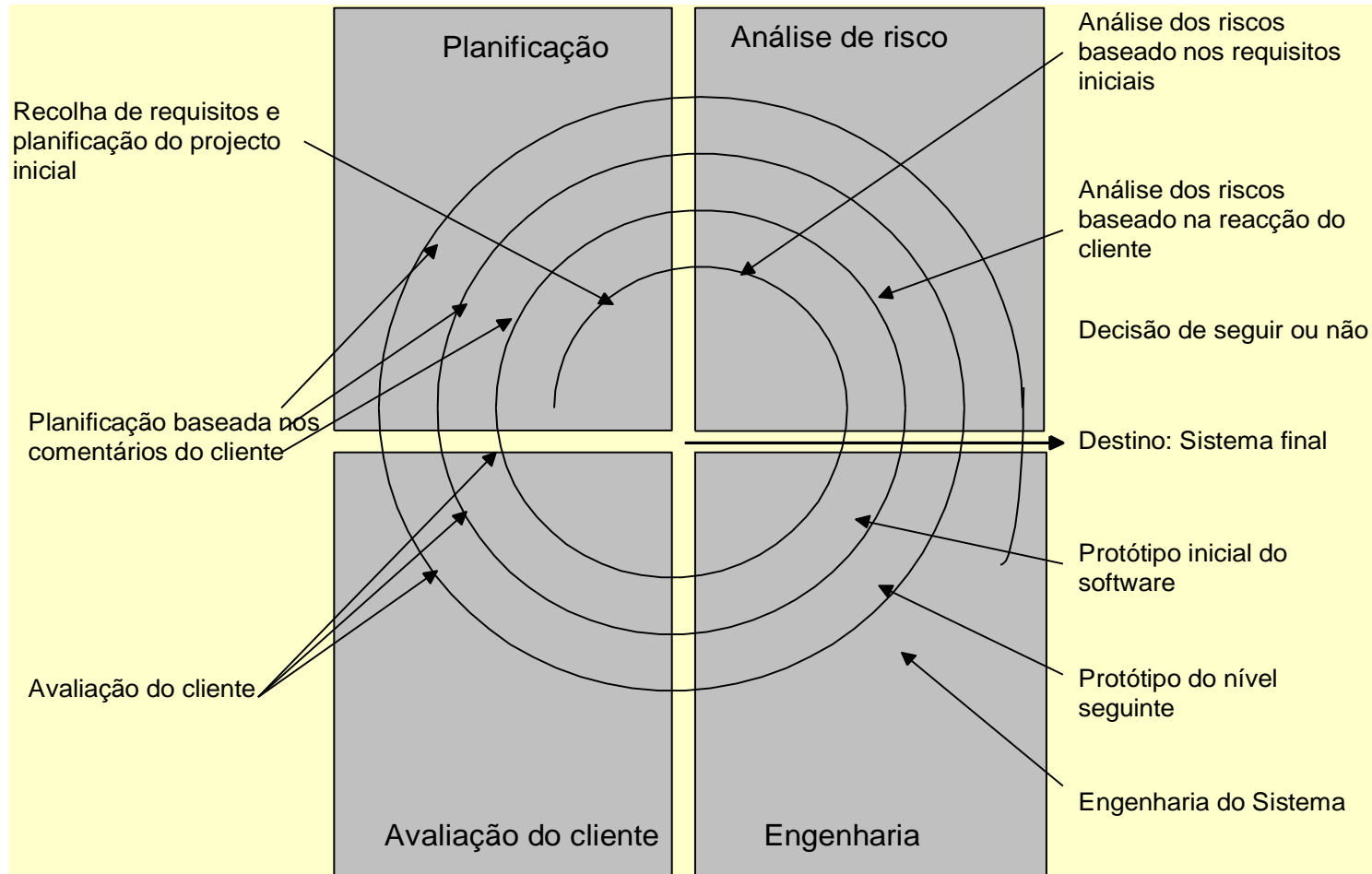
- Engloba as melhores características do Ciclo de Vida Clássico + Prototipagem, adicionando um novo elemento: a **Análise de Riscos**
- Segue a abordagem de passos sistemáticos do *Ciclo de Vida Clássico* incorporando-os numa estrutura iterativa que reflecte melhor o mundo real
- Usa a *Prototipagem*, em qualquer etapa da evolução do produto, como mecanismo de redução de riscos

Modelo Espiral

Ciclo de vida clássico / prototipagem

- **Planificação - mecanismo de definição de requisitos**
- **Análise Riscos - alternativas, identificação/resolução**
- **Engenharia — desenvolvimento produto**
- **Avaliação - integração e verificação**

Modelo Espiral



- *Extreme Programming (XP) ou Agile Development* é uma nova maneira de desenvolver software onde existe uma grande preocupação em equilibrar as variáveis custo, tempo e qualidade do produto, utilizada em projetos curtos
- Para atingir estes objetivos o XP considera alguns princípios básicos: retro alimentação rápida, simplicidade, mudanças incrementais, adaptação à mudança, trabalho de qualidade

Características que a equipa deve valorizar

- Comunicação
- Simplicidade
- Retro alimentação
- Coragem

Actividades principais propostas pelo método

- Codificar
- Testar
- Ouvir
- Desenhar

Estas actividades reflectem a filosofia da programação rápida, onde não há tempo para versões preliminares

Desenvolvimento Ágil

- Assim que a equipe tem disponível todas as informações necessárias para a programação, esta actividade inicia-se, sendo o código o principal documento resultante do desenvolvimento
- A partir de então cada nova funcionalidade implementada é mostrada ao cliente para validação. Como a programação em pares é adoptada, deve existir um padrão de codificação

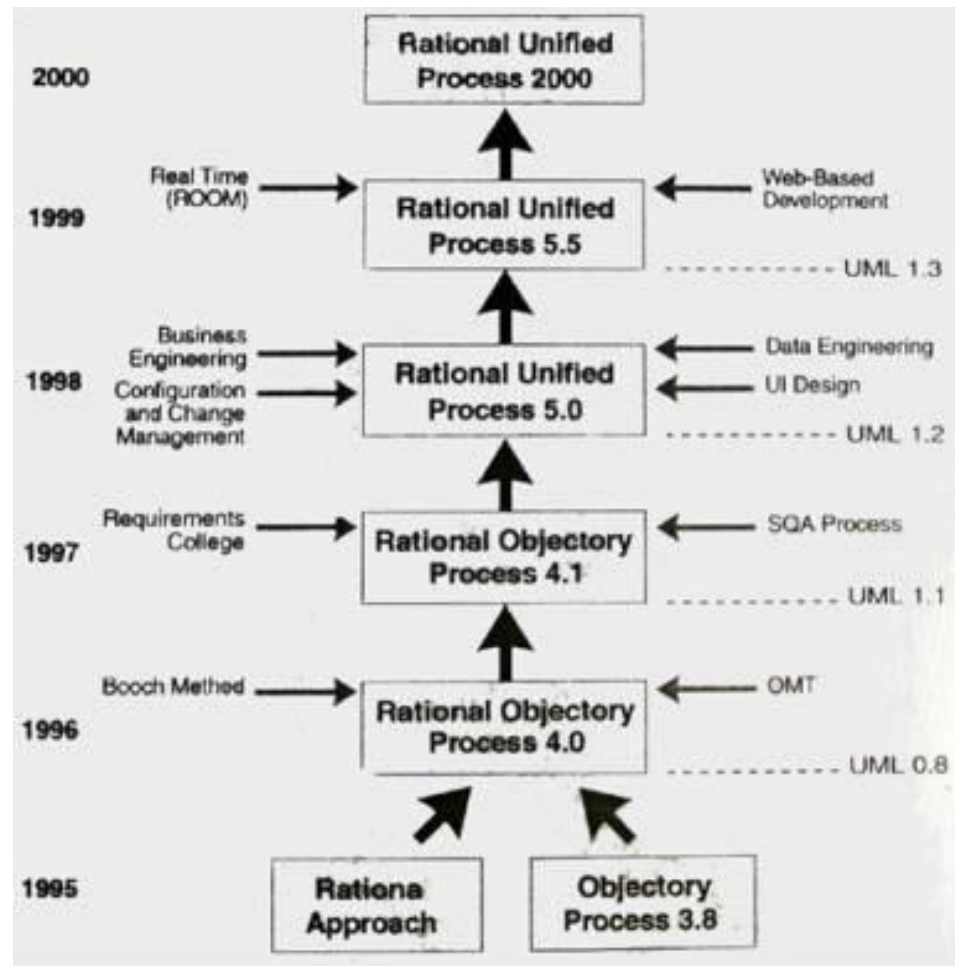
Práticas a ter em conta:

- **Planeamento**
- **Versões simples e incrementais**
- **Desenho simples**
- **Testes**
- **Programação em pares**
- **Integração contínua**
- **Semana de 40 horas**
- **Cliente presente**

RUP - Rational Unified Process

- Rational Unified Process foi criado pela Rational em 1998, derivou do anterior Rational Objectory Process. Criado pela Rational Software Corporation, adquirida pela IBM, passando a IRUP (IBM Rational Unified Process)
- O objetivo do RUP é agrupar as melhores práticas no processo de desenvolvimento de software, definindo as pessoas e respectivos papéis que desempenham

RUP - Rational Unified Process

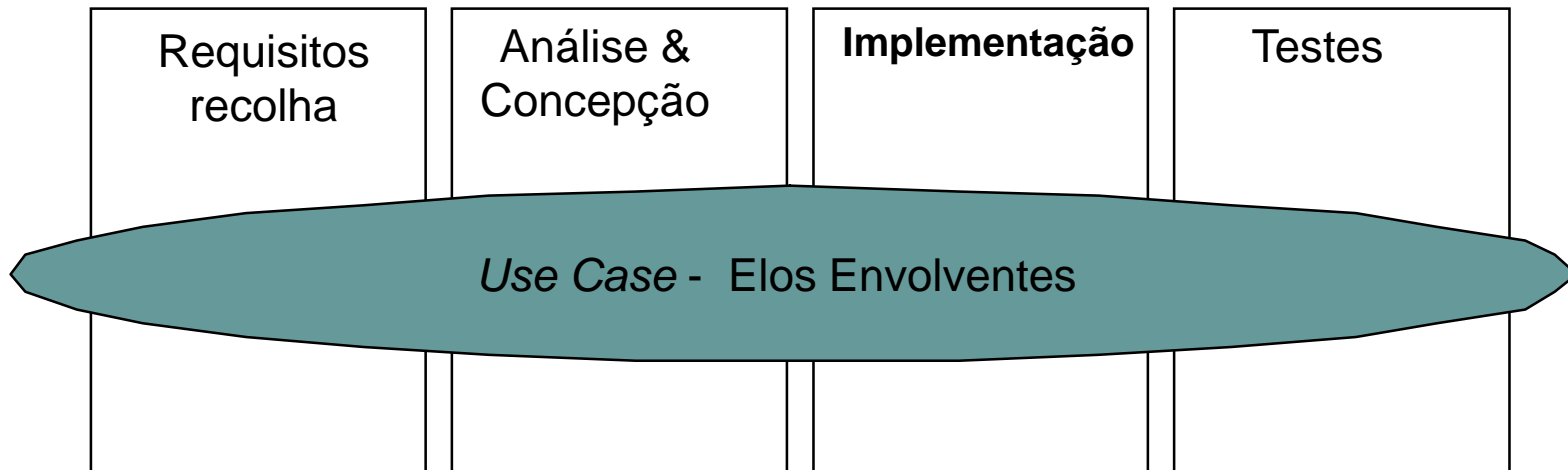


IBM Rational Unified Process

1. Início (*Inception*)
2. Elaboração
3. Construção
4. Transição

Controlado pelos Use Case

Todas as actividades guiadas pelos casos de uso



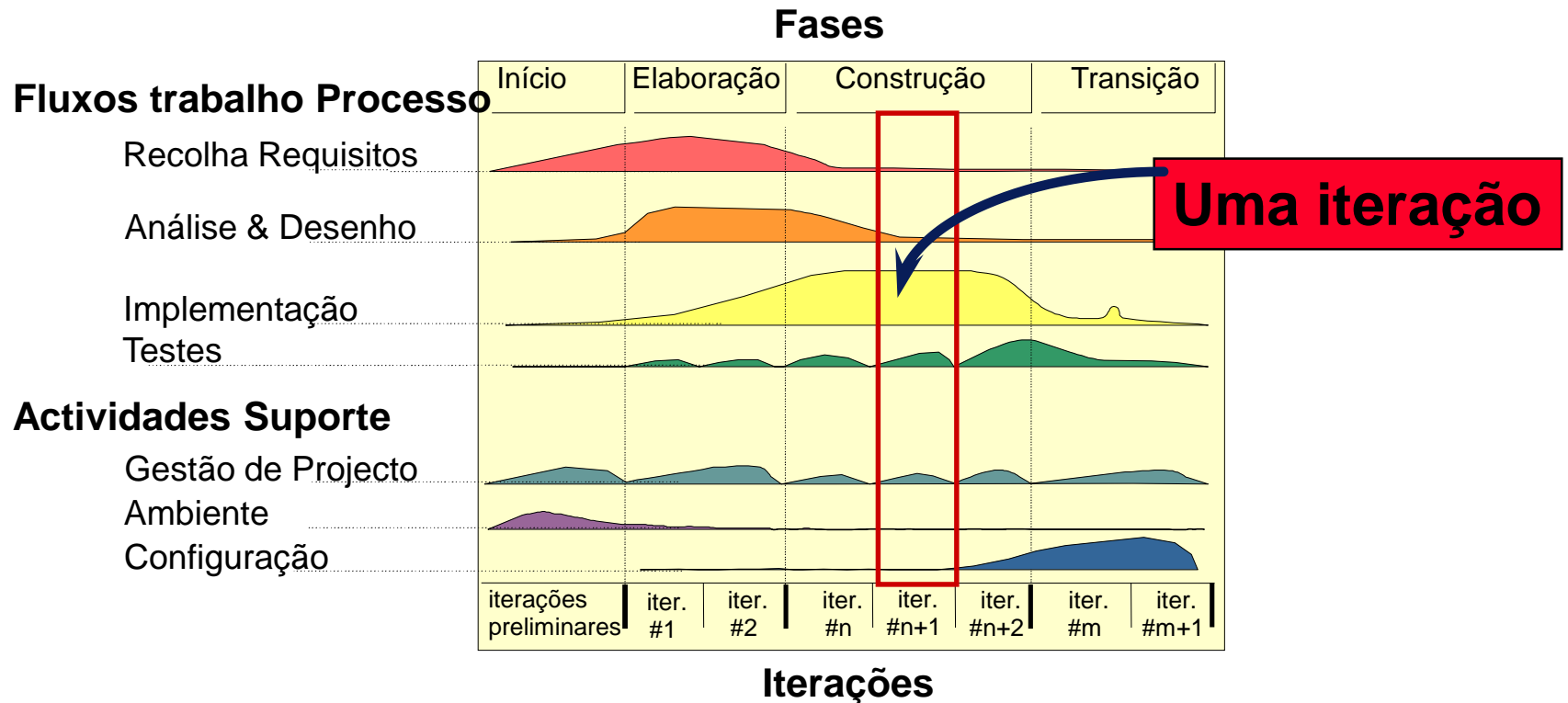
Fonte: *Jacobson*, 1999

Centrado sobre a Arquitetura

- Satisfazer as necessidades expressas nos *Use Case*
- Evolução e restrição de utilização
- Arquitetura simples
- Compreensão intuitiva

→ Reutilização

RUP: Iterativo e Incremental



Fonte: Jacobson 1999

Objetivos:

- especificar o âmbito do projeto
- identificar entidades externas com as quais o sistema irá interagir (atores) e a natureza desta interação é definida a um alto nível (Casos de uso). Um ator é um papel desempenhado por uma pessoa física ou sistema que interage com o sistema
 - quem é que usa diretamente o sistema
 - quem é responsável pela manutenção do sistema
 - hardware externo usado pelo sistema
 - outros sistemas que interagem com o sistema

Identificação de caso de usos

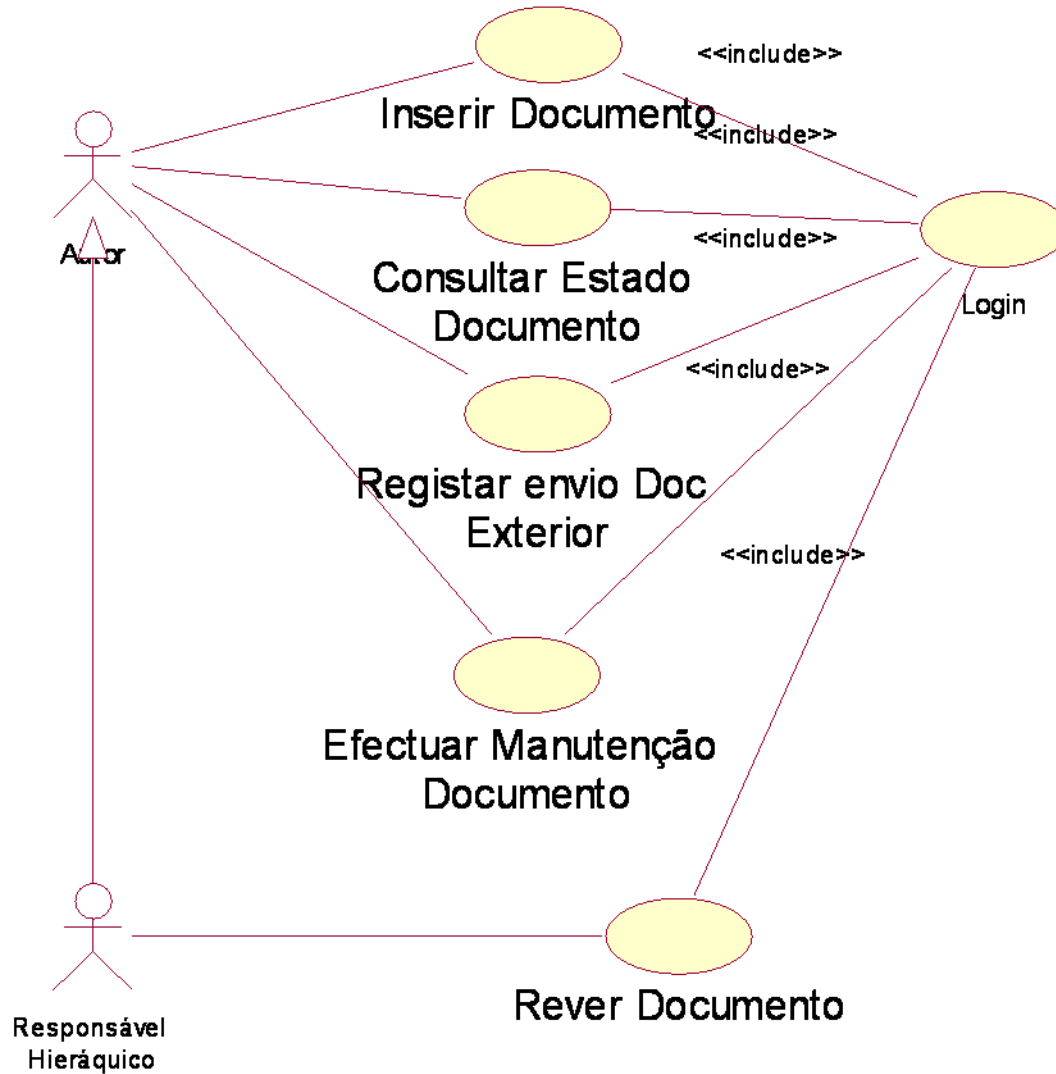
- Um caso de uso é um modo específico de usar um sistema - alguma funcionalidade é realizada pelo sistema como resposta a um estímulo de um ator. Os casos de uso formam um diálogo entre o ator e o sistema. Os casos de uso providenciam um veículo para:
 - 1 - capturar os requisitos sobre o sistema,
 - 2 - comunicar com os utilizadores finais e o peritos na matéria;
 - 3 - testar o sistema

Efetuar Encomenda Internet

Efectuar Encomenda Internet (Cenário Principal)

Pré-condição	O cliente é um utilizador válido no sistema.
Descrição	1. O use case começa quando o cliente selecciona a opção de Encomendar.
	2. Em simultâneo com a sua encomenda é mostrado o catálogo de produtos.
	3. O cliente adiciona produtos à encomenda através da introdução do código do produto.
	4. Automaticamente, o sistema mostra o nome, descrição e preço do produto.
	5. De cada vez que é adicionado um produto, o valor total da encomenda é calculado.
	6. O cliente confirma a sua encomenda através da opção Confirmar.
	7. O sistema pede então os detalhes do cartão de crédito.
	8. O sistema confirma os dados do pagamento e atribui um número de identificação à encomenda.
Pós-Condição	A encomenda será entregue na morada do cliente.

Diagrama Casos de Uso



RUP - Elaboração

Objetivos da fase elaboração:

- Analisar o domínio do problema
- Estabelecer a arquitetura
- Desenvolver o plano do projeto
- Eliminar os elementos de maior risco do projeto

A descrição do comportamento do sistema, expressa em casos de uso, o contexto do sistema, os atores, os cenários e um modelo de classes

Para os projetos de tamanho médio, poderão existir:

- Algumas dezenas de caso de uso
- Uma centena de cenários principais
- Algumas centenas de cenários secundários
- Entre cinquenta e cem classes

Elaboração - atividades

Desenvolvimento do plano do projeto - passos

- 1 - Identificar e estabelecer prioridades para os principais riscos do projeto
- 2 - Selecionar um pequeno número de cenários referentes aos riscos mais prioritários
- 3 - Selecionar as classes e os relacionamentos a serem implementadas pela análise dos cenários
- 4 - Implementar as classes e os relacionamentos selecionados
- 5 - Desenvolver casos de teste
- 6- Testar as classes e os relacionamentos para verificar funcionalidades
- 7 - Integrar e testar com iterações prévias
- 8 - Avaliar e atribuir qualquer correção

Recorda-se a principal razão para o uso de uma abordagem iterativa:

- **Nem todos os requisitos são conhecidos a princípio**
- **Pode acontecer que alguns riscos não sejam eliminados de acordo com o planeado, que novos riscos sejam descobertos e que algum código seja eliminado devido à avaliação de uma iteração**

O objetivo da fase de construção é desenvolver um produto de software que esteja pronto a ser introduzido na comunidade de utilizadores. O produto é desenvolvido como uma série de iterações

**Uma iteração é tipicamente a implementação de um ou mais casos de uso.
O processo de construção de uma iteração envolve:**

- 1- identificação de classes e relacionamentos a serem implementados**
- 2 - criação do código para a iteração**
- 3 - criação/atualização da documentação para a iteração**
- 4 - teste da iteração**
- 5 - integração e teste da iteração com qualquer iteração prévia**

O processo de construção de iterações continua até o produto de software estar completo

Transição

O **objetivo** da fase de transição é lançar o produto de software na comunidade de utilizadores e dar-lhes formação

Os principais produtos com destino aos utilizadores incluem:

- Programas executáveis
- Os manuais para os utilizadores
- A documentação de distribuição e instalação

Os principais produtos com destino aos responsáveis do projeto:

- Os modelos revistos
- Os critérios de avaliação de cada iteração
- A descrição das entregas
- Os resultados do controlo de qualidade