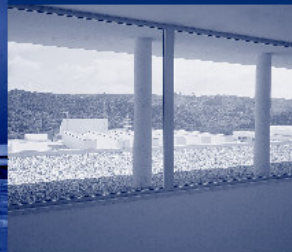




**Universidade de Coimbra**  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Informática



## ***Engenharia de Software 1***

### ***Introdução ao Rational Unified Process (RUP)***

# Rational Unified Process (RUP)

- É um processo (metodologia) de desenvolvimento de software
- Define actividades e responsabilidades para identificar e transformar requisitos dos utilizadores num produto de software (e documentação)

# Rational Unified Process (RUP)

Assenta em 6 “melhores práticas”:

1. Desenvolvimento iterativo
2. Gestão de requisitos
3. Utilização de arquitecturas baseadas em componentes
4. Modelação visual do software
5. Verificação da qualidade do software
6. Controlo de alterações ao software

# Rational Unified Process (RUP)

## 1. Desenvolvimento iterativo

É cada vez menos viável definir sequencialmente todo o problema, conceber a solução completa, desenvolver o software e depois testar o produto.

É necessária uma abordagem iterativa que permita uma compreensão gradual, através de refinamentos sucessivos, e a criação incremental de uma solução em múltiplas iterações

# Rational Unified Process (RUP)

## 1. Desenvolvimento iterativo (cont.)

- Atacar cedo as partes arriscadas
- Atacar risco através de demonstração de progresso (releases)
- Cada release = executável => foco nos resultados
- Pontos de situação frequentes => projecto nos prazos
- Abordagem iterativa também acomoda mais facilmente mudanças tácticas nos requisitos, características ou prazos.

# Rational Unified Process (RUP)

## 2. Gestão de requisitos

- Identificação
- Organização
- Documentação dos requisitos e restrições
- Seguir (*track/trace*) e documentar compromissos e decisões tomadas
- Casos de uso...



# Rational Unified Process (RUP)

## 3. Utilização de arquitecturas baseadas em componentes

Definição (cedo) de uma arquitectura robusta (antes de envolver recursos em larga escala) – deve ser flexível, acomodar mudança, compreensível, promover reutilização

Uso de componentes existentes ou novos (operacionalizados com recurso a COM, CORBA, etc)

# Rational Unified Process (RUP)

## 4. Modelação visual do software

- UML na sua base
- Captura da estrutura
- Captura do comportamento
- Ocultar pormenores
- Como os vários elementos “encaixam”



# Rational Unified Process (RUP)

## 5. Verificação da qualidade do software

Verificação da qualidade em todas as actividades, nas várias vertentes:

- Confiabilidade
- Funcionalidade
- Desempenho...

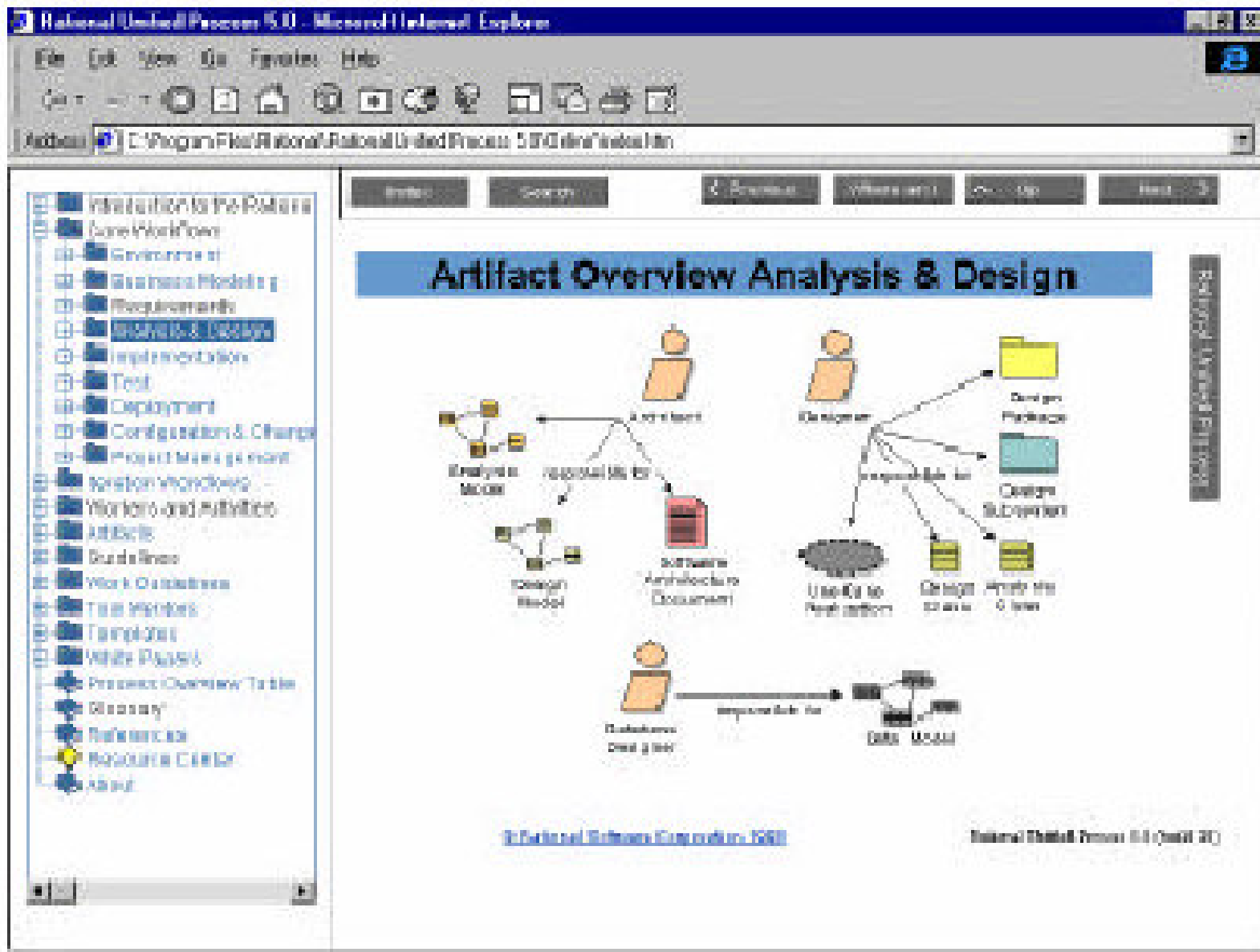
# Rational Unified Process (RUP)

## 6. Controlo de alterações ao software

Sendo as alterações inevitáveis, há que manter controlo sobre elas. Aceitável? Seguimento? Monitorização?

- Garantir que o processo iterativo é bem sucedido
- Espaços seguros para cada elemento da equipa (isolar)
- Controlo abrange modelos, código, documentos...

# Rational Unified Process (RUP)



# Rational Unified Process (RUP)

## O **processo** como um **produto**:

– Disponibilizado como um sistema baseado em web que inclui:

- Linhas de orientação
- Ferramentas (modelação, documentação, ...)
- Ajuda sobre o uso das ferramentas
- Documentos padrão (cronogramas, documentação de sw,...)
- Livro
- Mecanismos de personalização
- ...

# Rational Unified Process (RUP)

- Admite que os requisitos mudam

Num estudo de 8000 projectos, 40% dos requisitos definitivos chegaram depois da fase de análise, já o desenvolvimento tinha começado (Capers Jones)

- Promove um controlo do risco melhor e antecipado – divide o sistema em mini-projectos e centra-se primeiro as partes de maior risco

# Rational Unified Process (RUP)

- Permite “projectar um pouco, desenhar um pouco e codificar um pouco”
- Assenta no uso da linguagem UML para a modelação
- É conduzido por casos de uso, cujo objectivo é manter a ligação desde os requisitos até aos testes

# Rational Unified Process (RUP)

- Promove o desenvolvimento iterativo e incremental – redução do risco
- Recorrendo a uma arquitectura robusta que é inicialmente definida
- Permite detectar e corrigir erros mais cedo
- Promove a evolução gradual em vez do esforço total seguido do big-bang da implantação



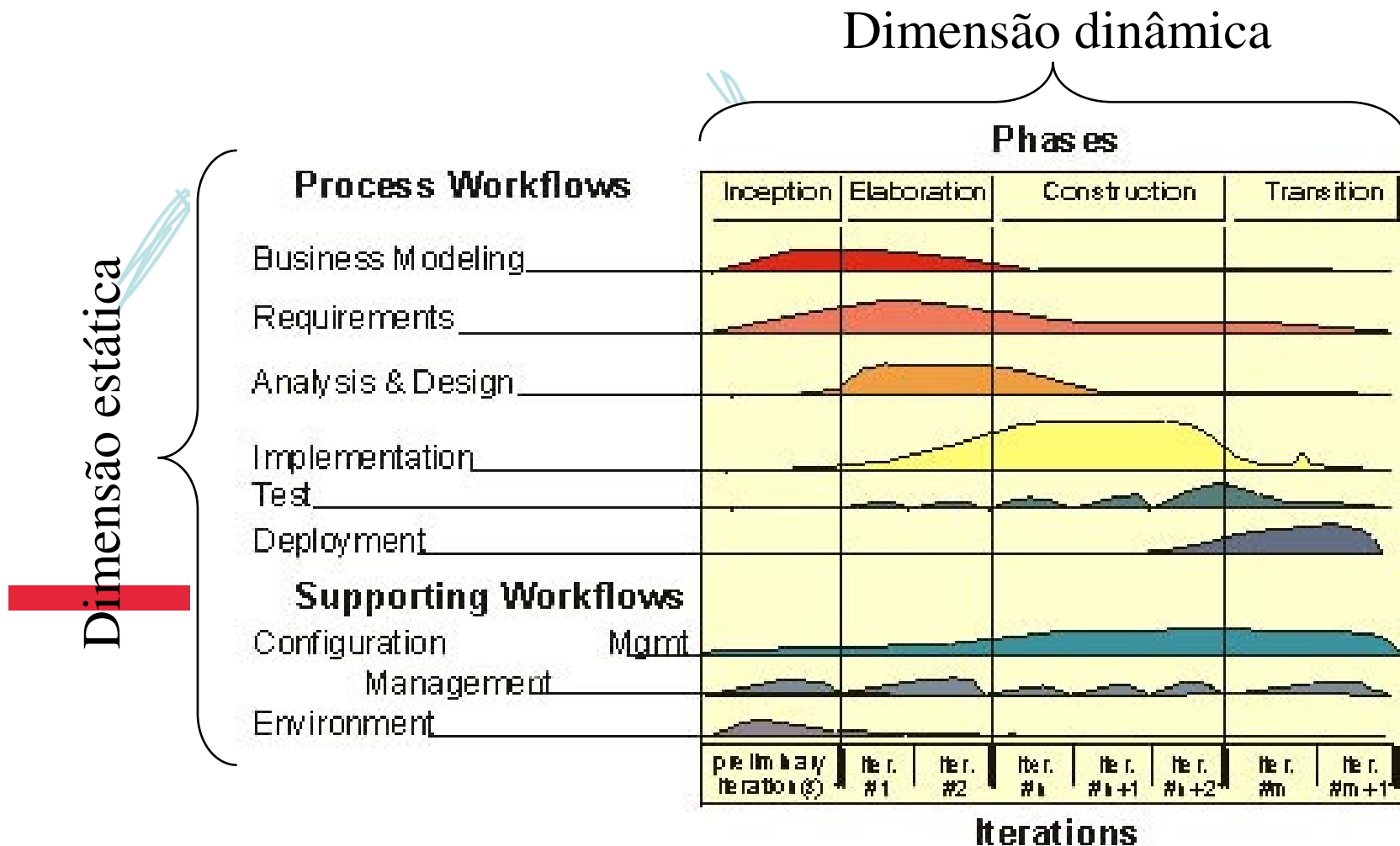
# Rational Unified Process (RUP)

- Promove o envolvimento, desde cedo, de todos os intervenientes (incluindo testadores, integradores e produtores de documentação)
- Permite melhorias ao software, dando o tempo necessário à evolução natural
- • Promove a estabilidade sólida de base (de resto, necessária para a abordagem evolutiva)

# Rational Unified Process (RUP)

- Permite que o sistema (ou parte dele) esteja disponível para utilização mais cedo do que com abordagens tradicionais

# Rational Unified Process (RUP)



# Rational Unified Process (RUP)

## Duas dimensões

### – Dinâmica

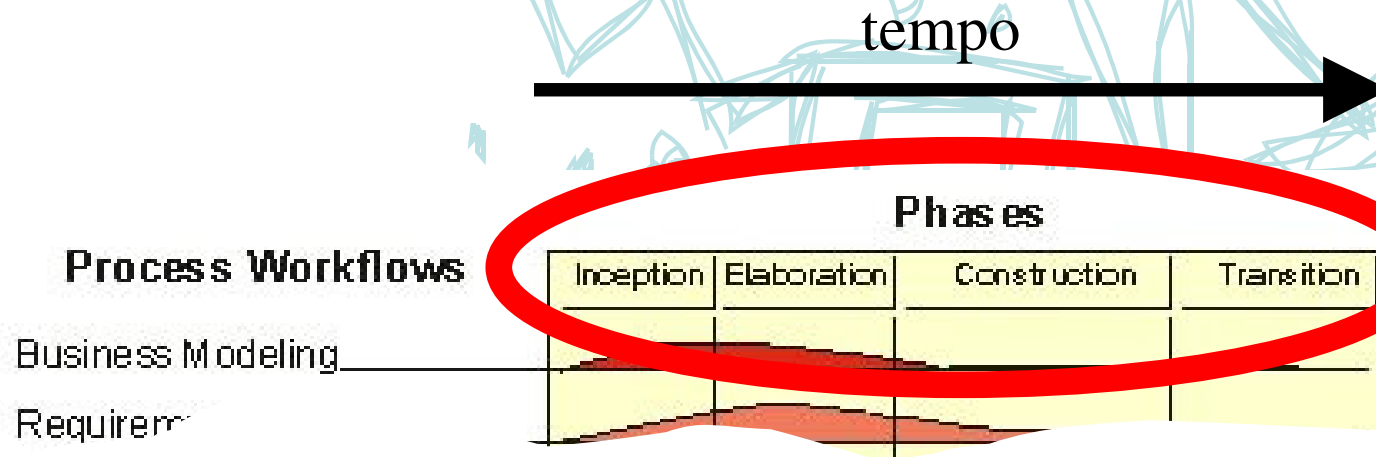
- Desenrolar do processo com o tempo – ciclos, fases, iterações e marcos

### – Estática

- Organização do processo em termos de actividades, artefactos, intervenientes e fluxos de trabalho

# Rational Unified Process (RUP)

- Fases do processo (dimensão dinâmica)
  - Concepção (*inception*)
  - Elaboração (*elaboration*)
  - Construção (*construction*)
  - Transição (*transition*)



# Rational Unified Process (RUP)

- Fase de concepção (*inception*)
  - Durante esta fase estabelece-se o *business case* do sistema e esclarece-se o âmbito do projecto. O *business case* deve incluir os critérios de sucesso, avaliação de risco, uma estimativa dos recursos necessários e um cronograma dos marcos temporais mais importantes.
  - Durante esta fase identificam-se todos os actores externos com que o sistema irá interagir e define-se a natureza dessa interacção a um nível de abstracção elevado. Tal envolve identificar todos os **casos de uso** e descrever os mais importantes.
  - Em resumo: “o que vai ser feito? Quanto tempo demora? Quando custa?”
  - No final desta fase há um marco de avaliação em que se decide avançar ou proceder a reformulações

# Rational Unified Process (RUP)

- Fase de concepção (*inception*)
  - Resultados a obter no final da fase:
    - Documento com visão geral dos requisitos nucleares do projecto, funcionalidades chave e constrangimentos mais importantes
    - Um primeiro modelo de **casos de uso** (10% a 20% completo)
    - Glossário inicial do projecto
    - Relatório inicial de avaliação do risco
    - *Business case* (justificação de negócio) inicial
    - Plano de projecto
    - Protótipos iniciais



# Rational Unified Process (RUP)

- Fase de concepção (*inception*)
  - Critérios de avaliação a aplicar no final da fase:
    - Concordância dos *stakeholders* em relação ao âmbito do projecto, custos e prazos
    - Compreensão dos requisitos, evidenciada pela precisão de descrição dos casos de uso mais importantes
    - Credibilidade das estimativas de custo e prazos, prioridades, riscos e processo de desenvolvimento
    - Comparação de gastos planeados com gastos realizados
- O projecto pode ser cancelado ou fortemente reformulado se falhar este marco

# Rational Unified Process (RUP)

- Fase de elaboração (*elaboration*)
  - É objectivo desta fase analisar o domínio do problema, projectar uma arquitectura sã, realizar um plano do projecto e eliminar os elementos de maior risco.
  - As decisões arquitecturais devem levar em linha de conta a totalidade do sistema (implica a descrição da maioria dos casos de uso): âmbito, principais funções e requisitos não funcionais (ex.: desempenho).
  - Como prova de conceito cria-se um sistema que demonstre as opções tomadas e implemente **casos de uso** chave (maiores **riscos**) – protótipo.

# Rational Unified Process (RUP)

- Fase de elaboração (*elaboration*)
  - Riscos:
    - De requisitos
    - Tecnológicos
    - Das competências
    - Políticos
    - ...

# Rational Unified Process (RUP)

- Fase de elaboração (*elaboration*)
  - Resultados a obter no final da fase:
    - Modelo de casos de uso (cerca de 80% completo)
      - todos os casos e actores identificados e a maioria das descrições realizadas
    - Requisitos adicionais descrevendo aspectos não funcionais e aspectos não directamente ligados a qualquer caso de uso
    - Descrição da arquitectura do software
    - Protótipo executável da arquitectura
    - Lista de riscos e *business case* revistos
    - Plano de projecto total
    - Um manual de utilizador preliminar (opcional)

# Rational Unified Process (RUP)

- Fase de elaboração (*elaboration*)
  - Critérios de avaliação a aplicar no final da fase:
    - Estabilidade da visão para o produto
    - Estabilidade da arquitectura
    - Demonstração, com base no executável produzido, que os elementos de maior risco foram eliminados
    - Detalhe para a fase seguinte suficientemente detalhado e baseado em estimativas temporais credíveis
    - Concordância dos *stakeholders* de que a visão pode ser concretizada caso o plano seja cumprido
    - Comparação entre gastos planeados e realizados aceitável
- O projecto pode ser cancelado ou fortemente reformulado se falhar este marco

# Rational Unified Process (RUP)

- Fase de construção (*construction*)
  - Durante a fase de construção desenvolve-se **iterativa** e **incrementalmente** um producto completo que depois transitará para a comunidade de utilizadores.
  - Para tal descrevem-se os casos de uso restantes, preenchendo o design, completa-se a implementação e testa-se o software.
  - Em alguns casos há a possibilidade de paralelizar tarefas de construção (uma boa arquitectura tem aqui um papel fundamental)
  - Há a preocupação de gerir bem os recursos, otimizar os custos, os prazos e a qualidade.
  - No final desta fase decide-se se o produto está pronto para entrar em operação.

# Rational Unified Process (RUP)

- Fase de construção (*construction*)
  - Resultados a obter no final da fase:
    - O produto de software – primeira versão é a **Beta**
    - Manuais de utilizador
    - Uma descrição da versão corrente.



# Rational Unified Process (RUP)

- Fase de construção (*construction*)
  - Critérios de avaliação a aplicar no final da fase:
    - A versão corrente do produto está suficientemente estável e madura para ser disponibilizada à comunidade de utilizadores?
    - Os vários *stakeholders* estão prontos para a transição?
    - Os gastos realizados vs. previstos estão dentro do aceitável?

- A disponibilização pode ser adiada se os critérios não forem cumpridos

# Rational Unified Process (RUP)

- Fase de transição (*transition*)
  - Nesta fase disponibiliza-se o software à comunidade de utilizadores. É então vulgar surgirem questões que exigem desenvolvimento adicional para afinar o sistema, corrigir erros previamente não detectados, ou finalizar características que tivessem sido adiadas. Esta fase inicia-se normalmente com a vulgarmente chamada **versão Beta**.
  - No final desta fase decide-se se os objectivos foram atingidos e se deve ser iniciado um novo **ciclo**.
  - É também nesta altura que se reflecte sobre lições retiradas do projecto para melhorar o processo.

# Rational Unified Process (RUP)

- Fase de transição (*transition*)
  - Resultados a obter no final da fase:
    - Auto-suficiência dos utilizadores
    - Concordância dos *stakeholders* the as *baselines* de disponibilização estão completas e consistentes com a visão inicial
    - Atingir a *baseline* final o no melhor compromisso custo/tempo

# Rational Unified Process (RUP)

- Fase de transição (*transition*)
  - Critérios de avaliação a aplicar no final da fase:
    - Satisfação dos utilizadores
    - Os gastos realizados vs. previstos dentro do aceitável
- A disponibilização pode ser adiada se os critérios não forem cumpridos

# Rational Unified Process (RUP)

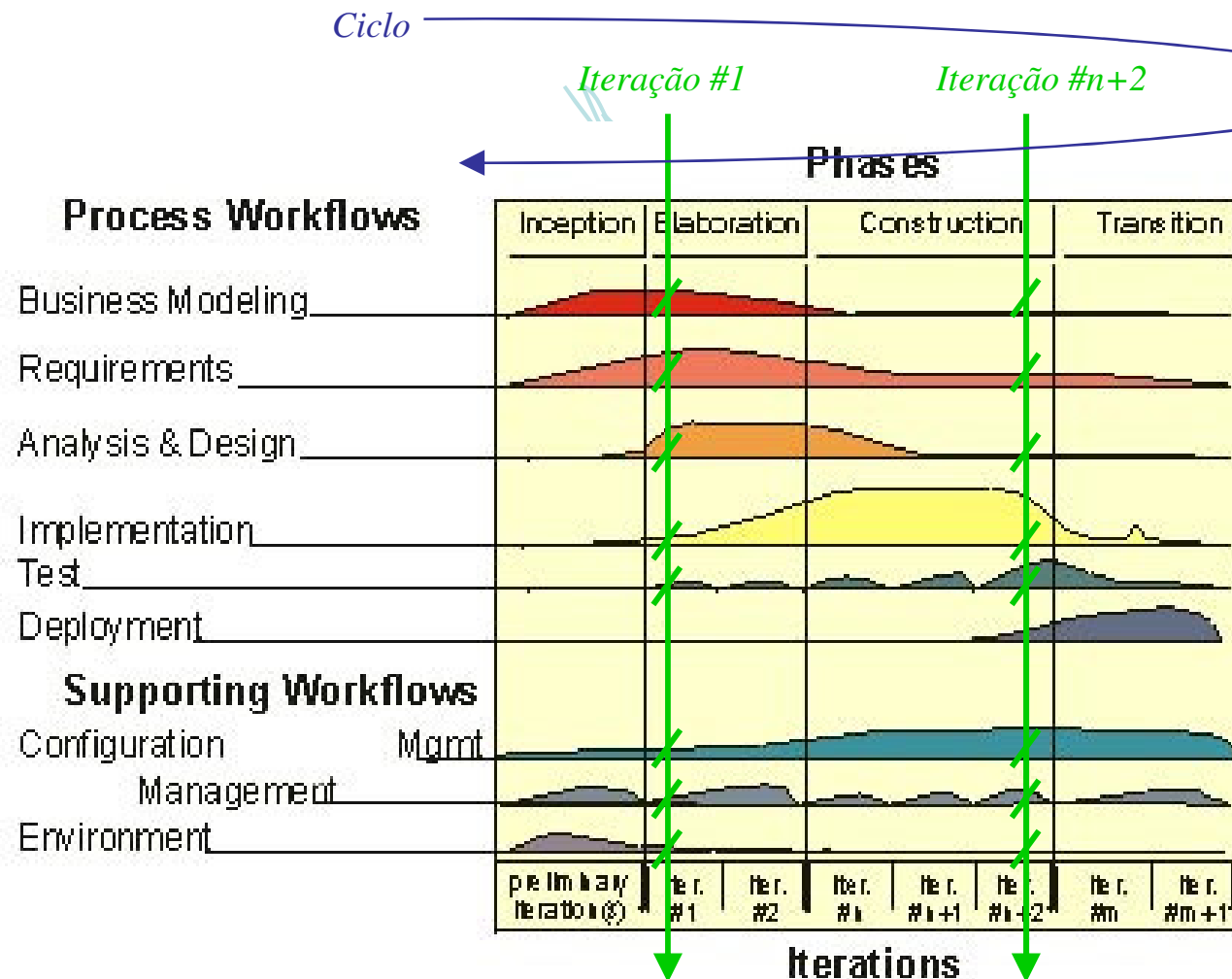
## Abordagem Iterativa

A aplicação sequencial das **quatro fases** descritas constitui um **ciclo**

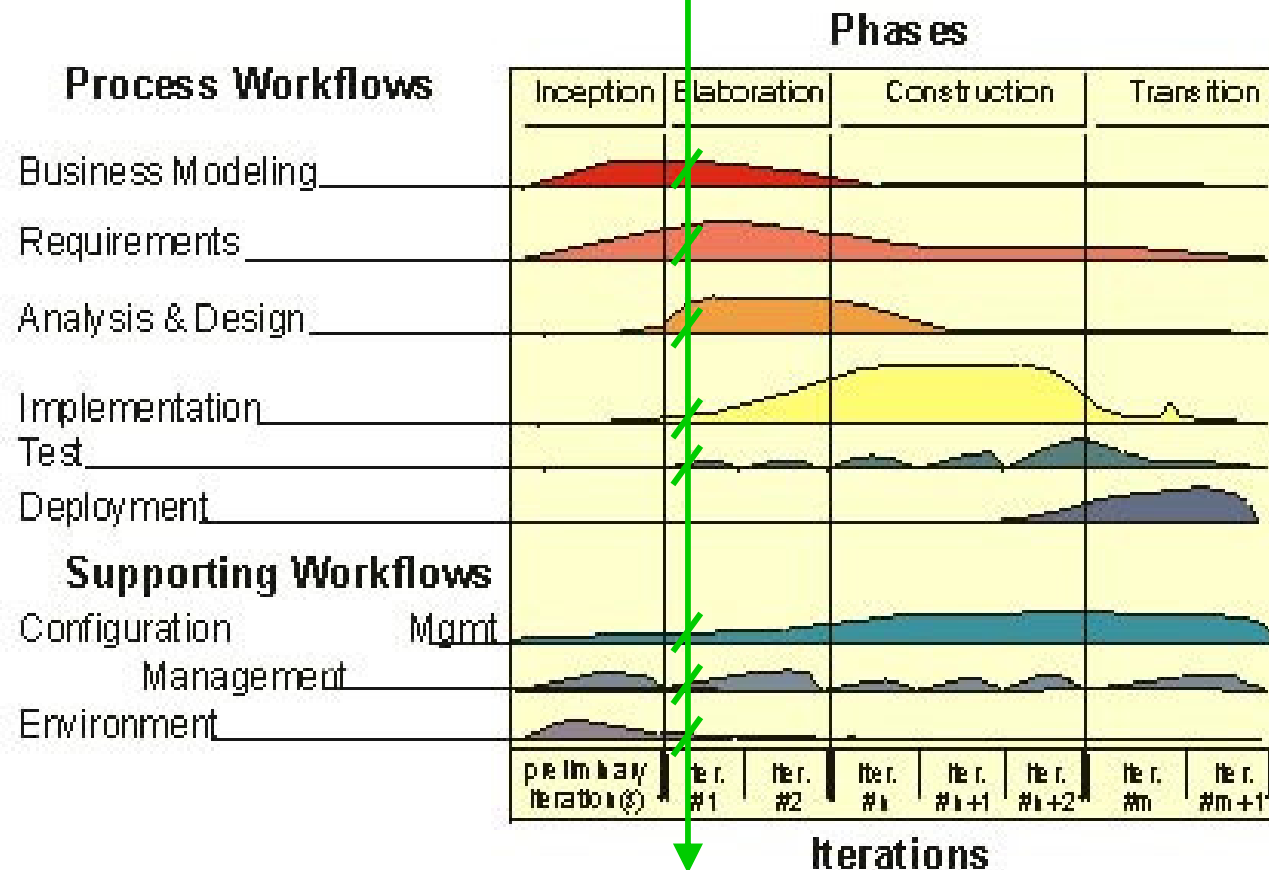
Mas **cada fase** pode ainda ser dividida em **iterações**

Em cada **iteração** aplicam-se sucessivamente as **actividades** previstas nos vários **workflows** com graus de intensidade diferentes, consoante a **fase** e a **iteração**

# Rational Unified Process (RUP)

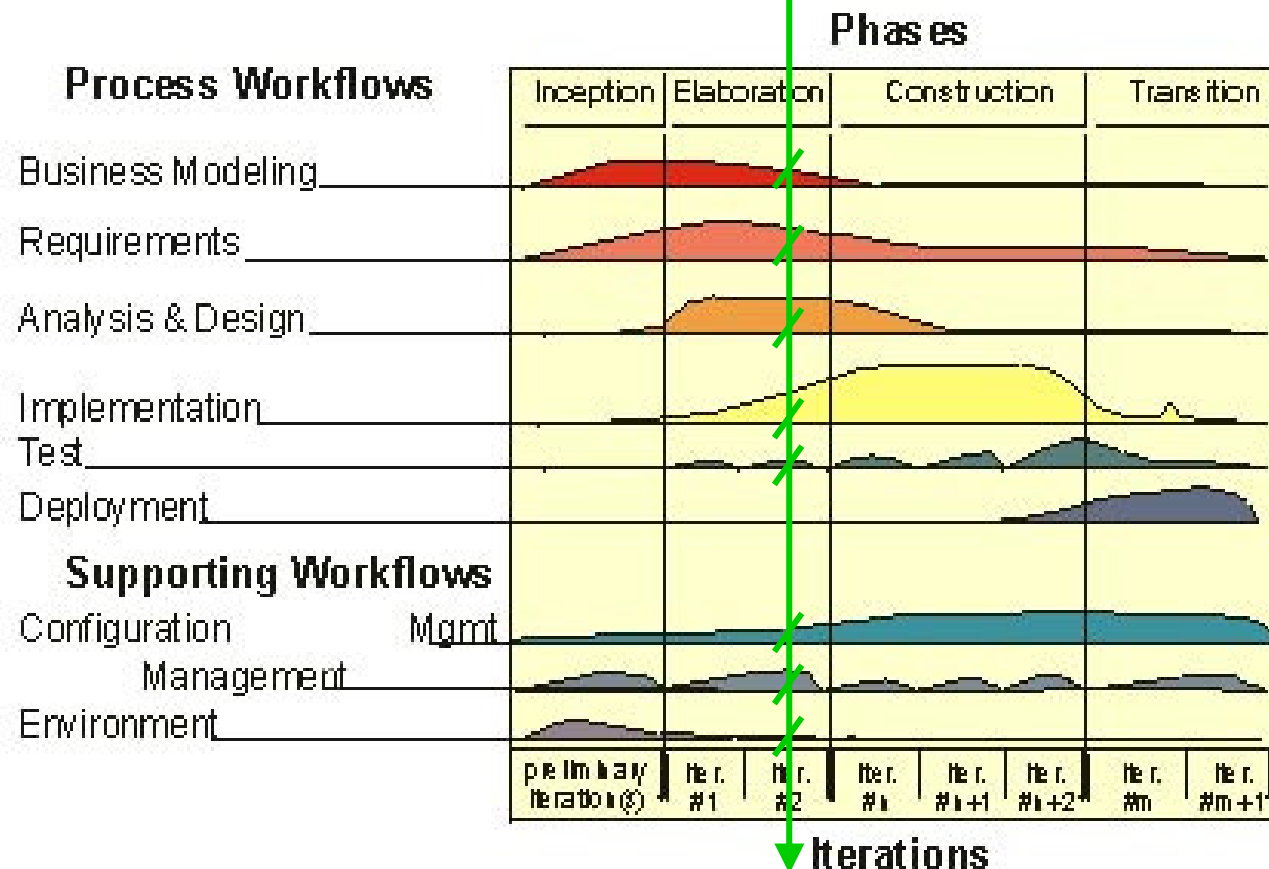


# Rational Unified Process (RUP)

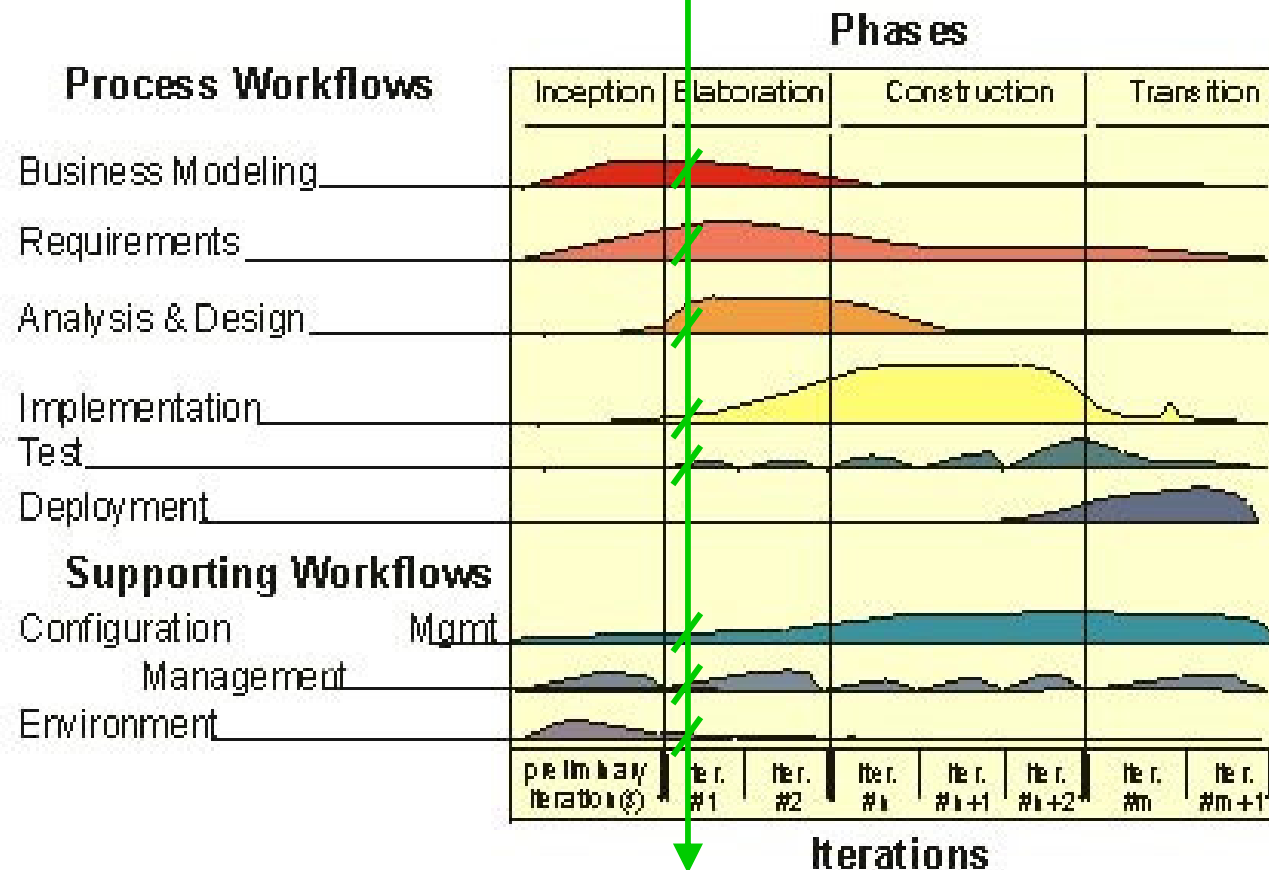




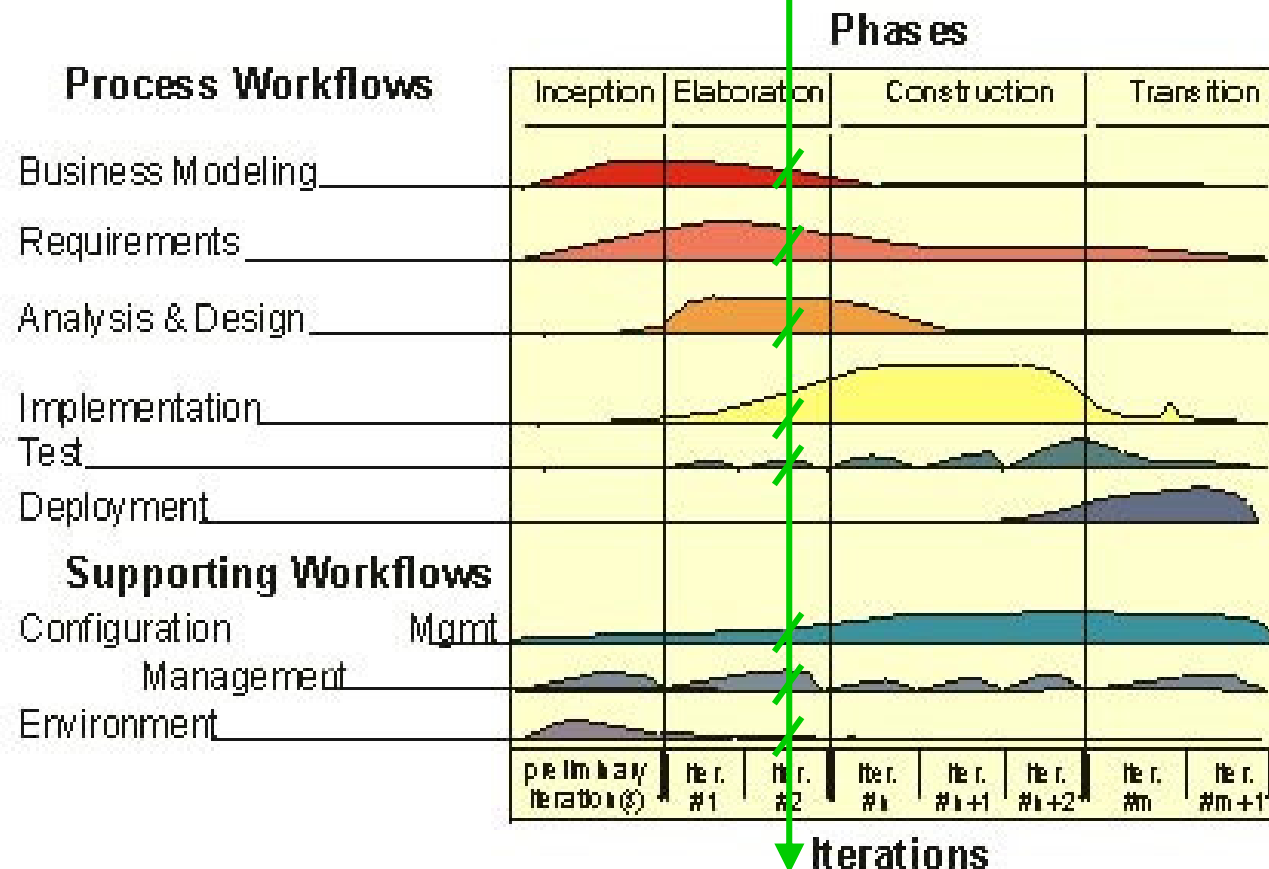
# Rational Unified Process (RUP)



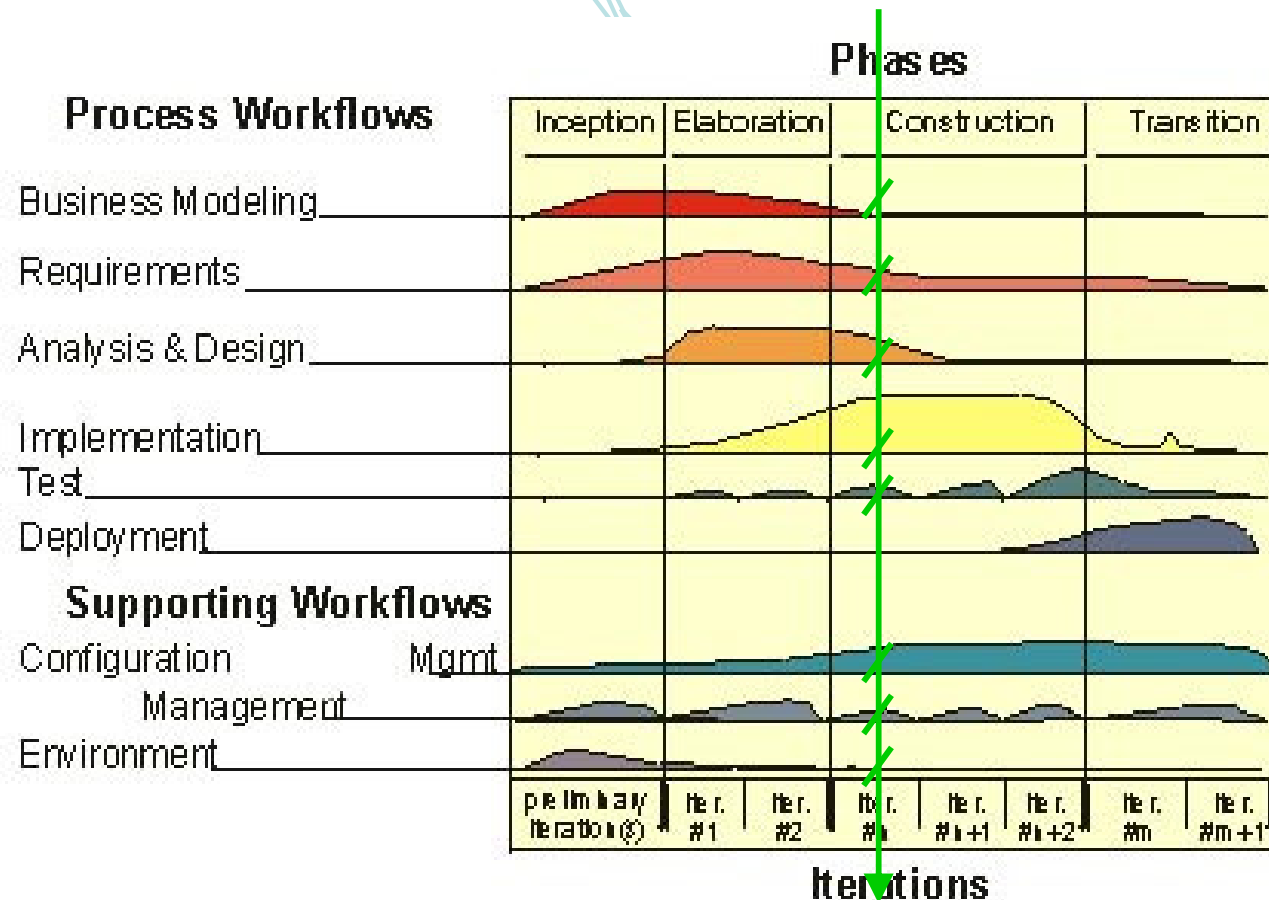
# Rational Unified Process (RUP)



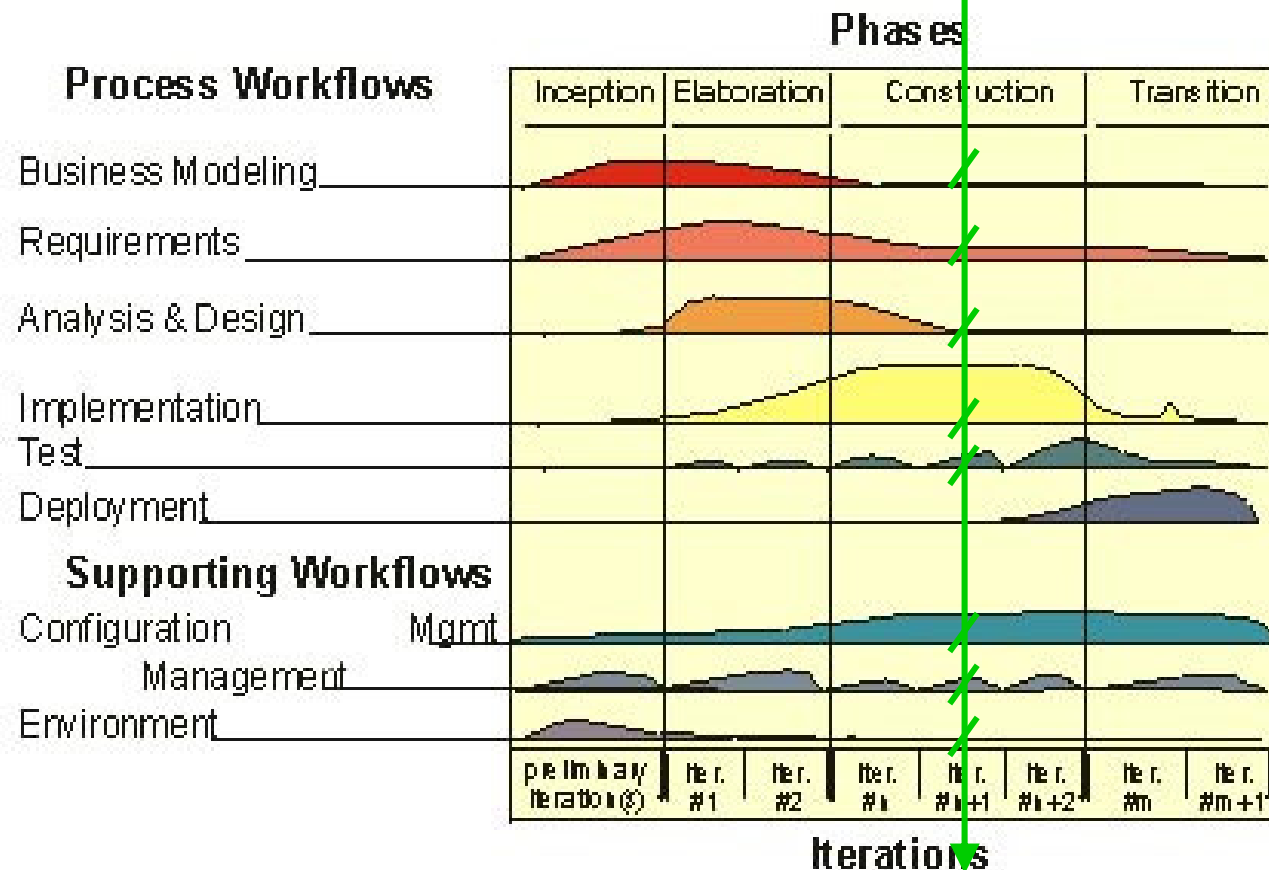
# Rational Unified Process (RUP)



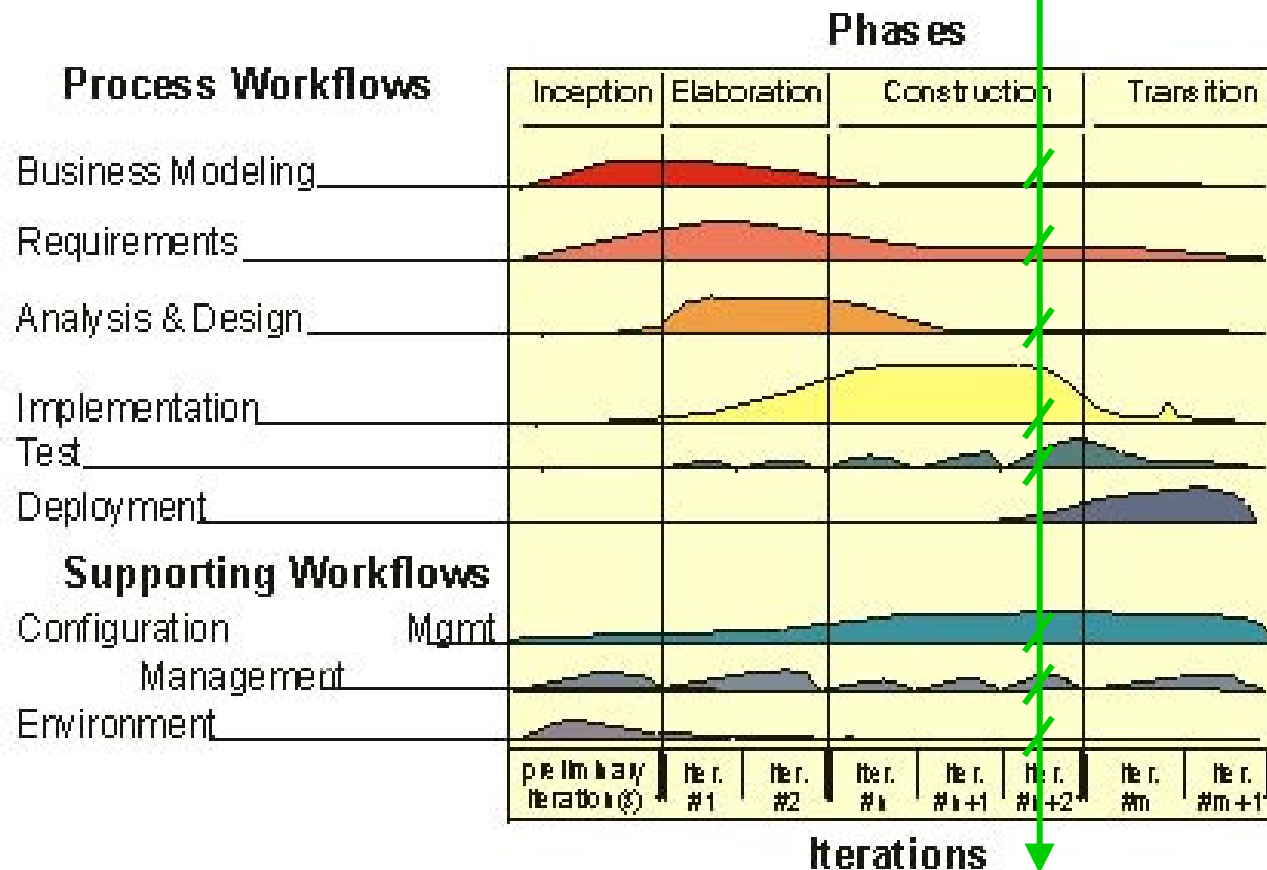
# Rational Unified Process (RUP)



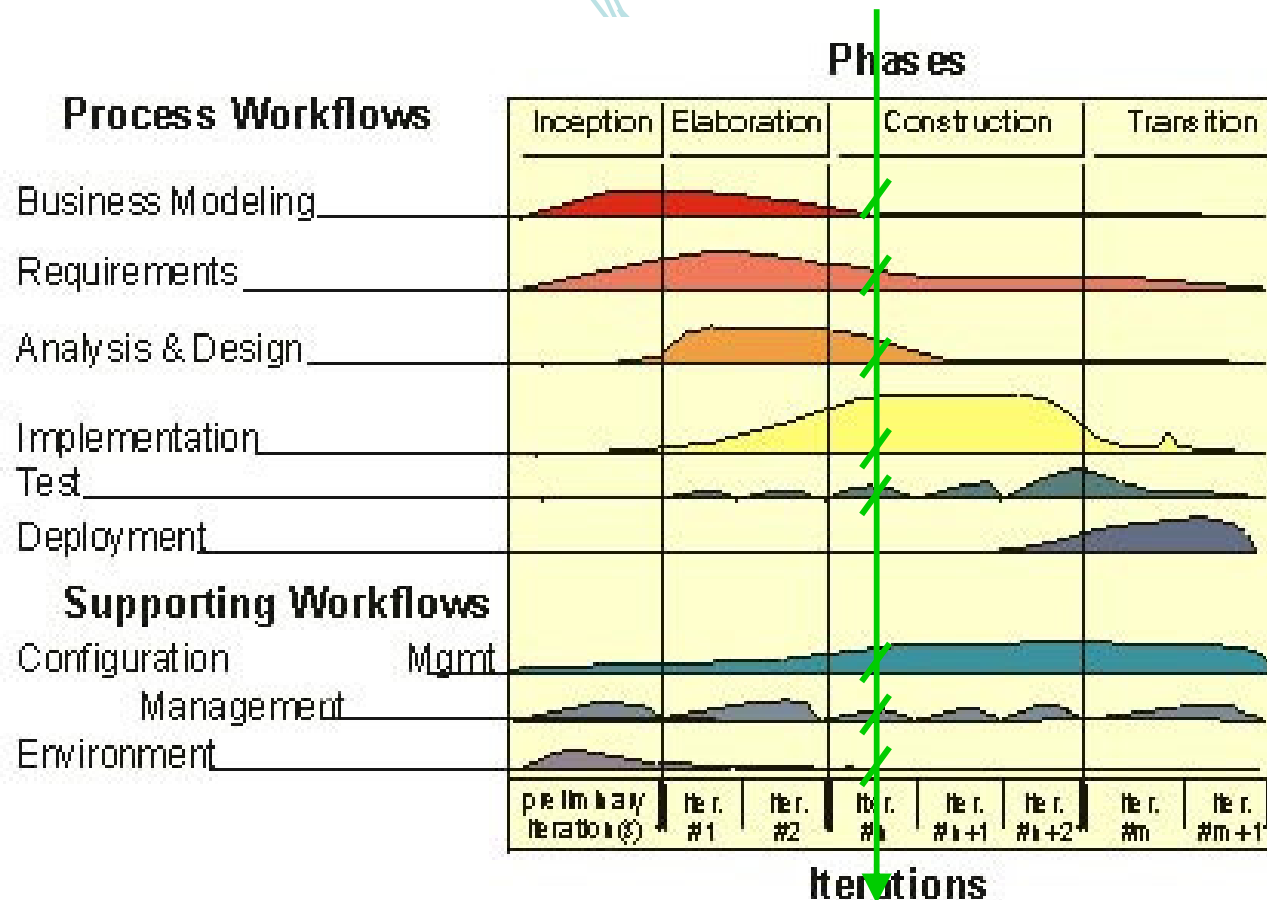
# Rational Unified Process (RUP)



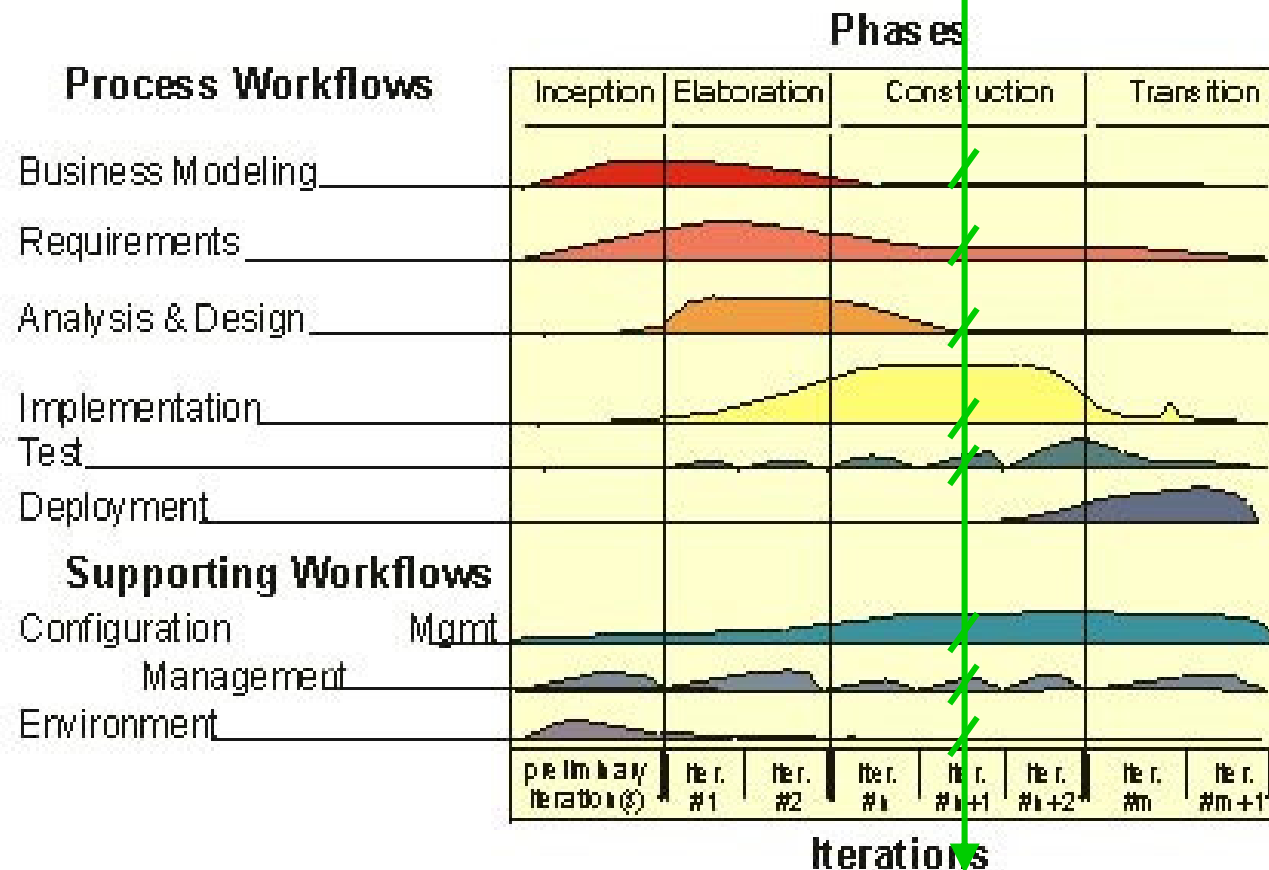
# Rational Unified Process (RUP)



# Rational Unified Process (RUP)

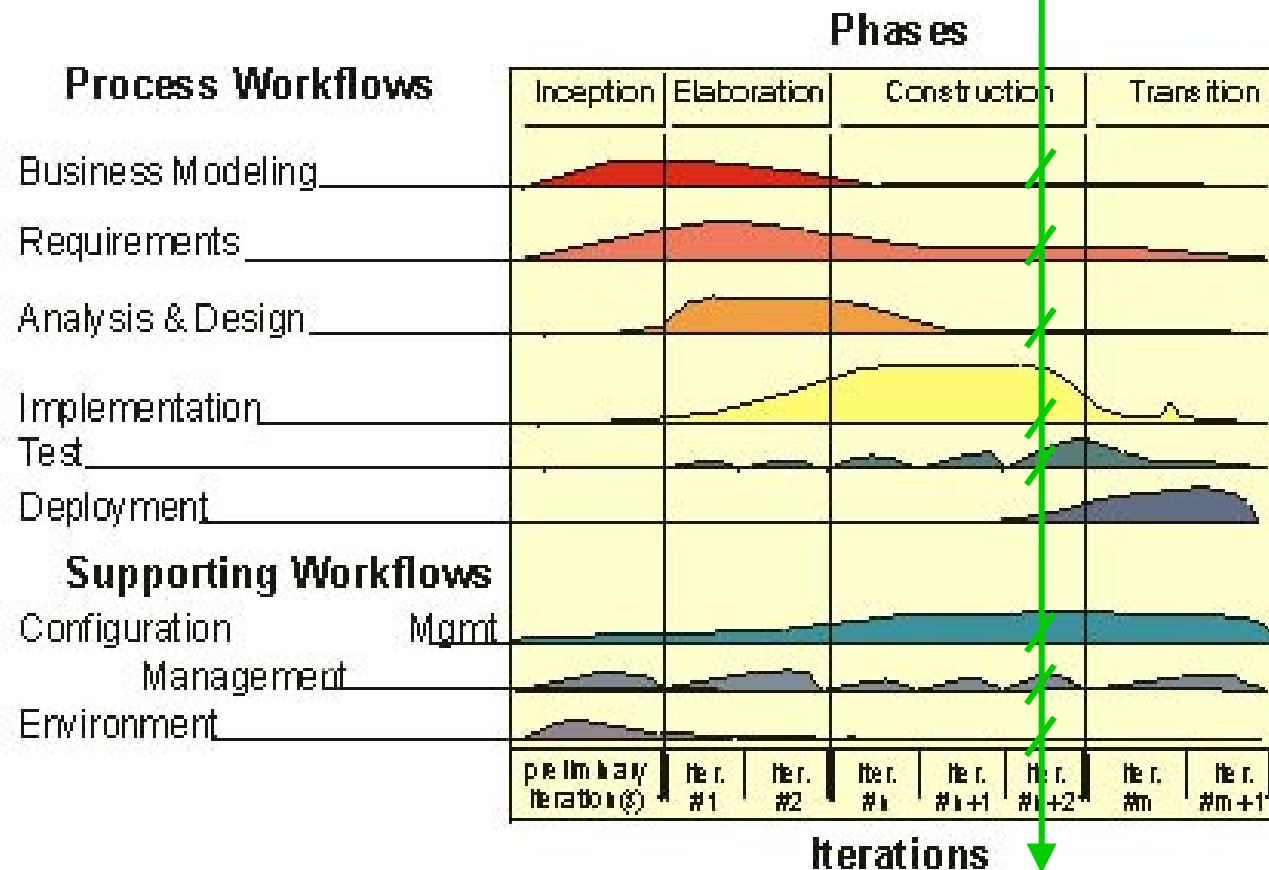


# Rational Unified Process (RUP)

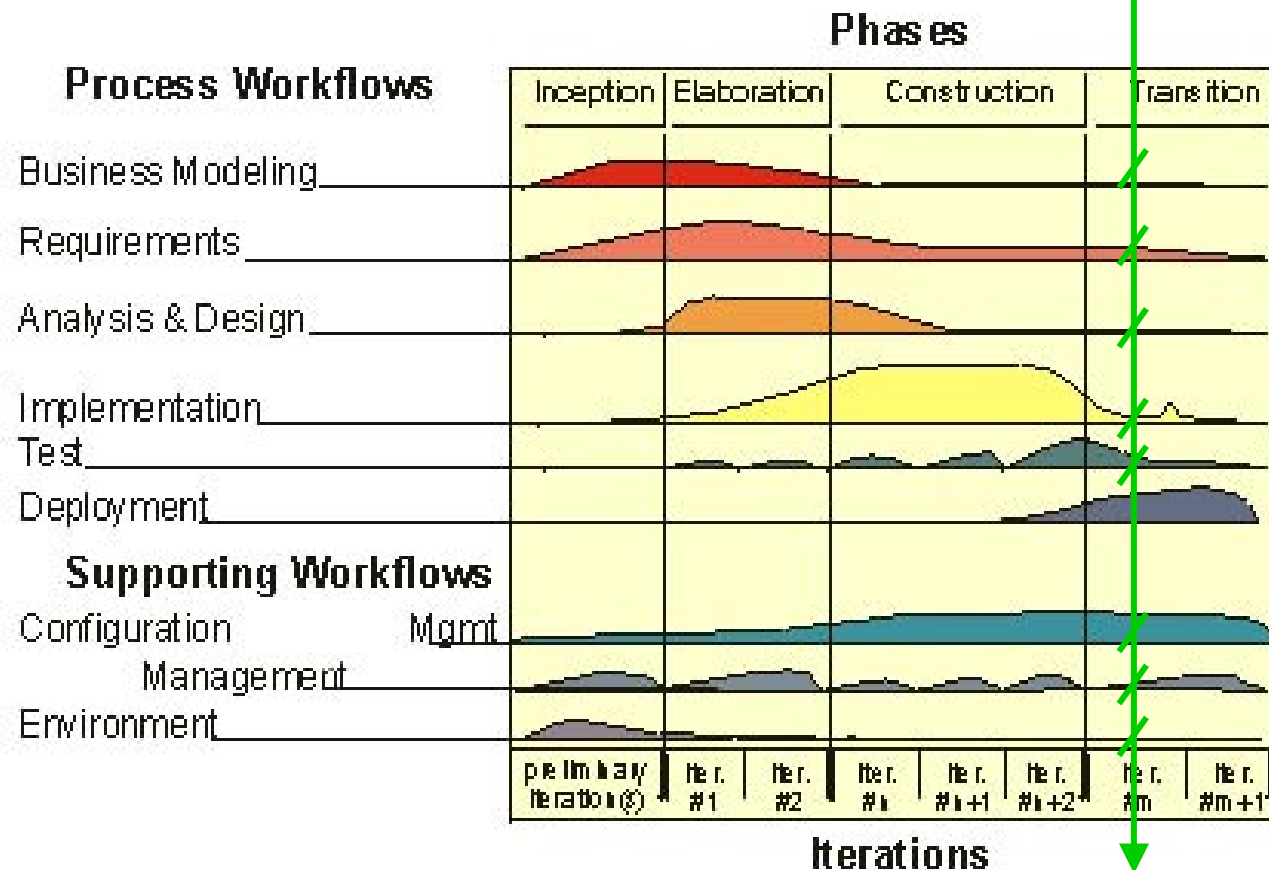




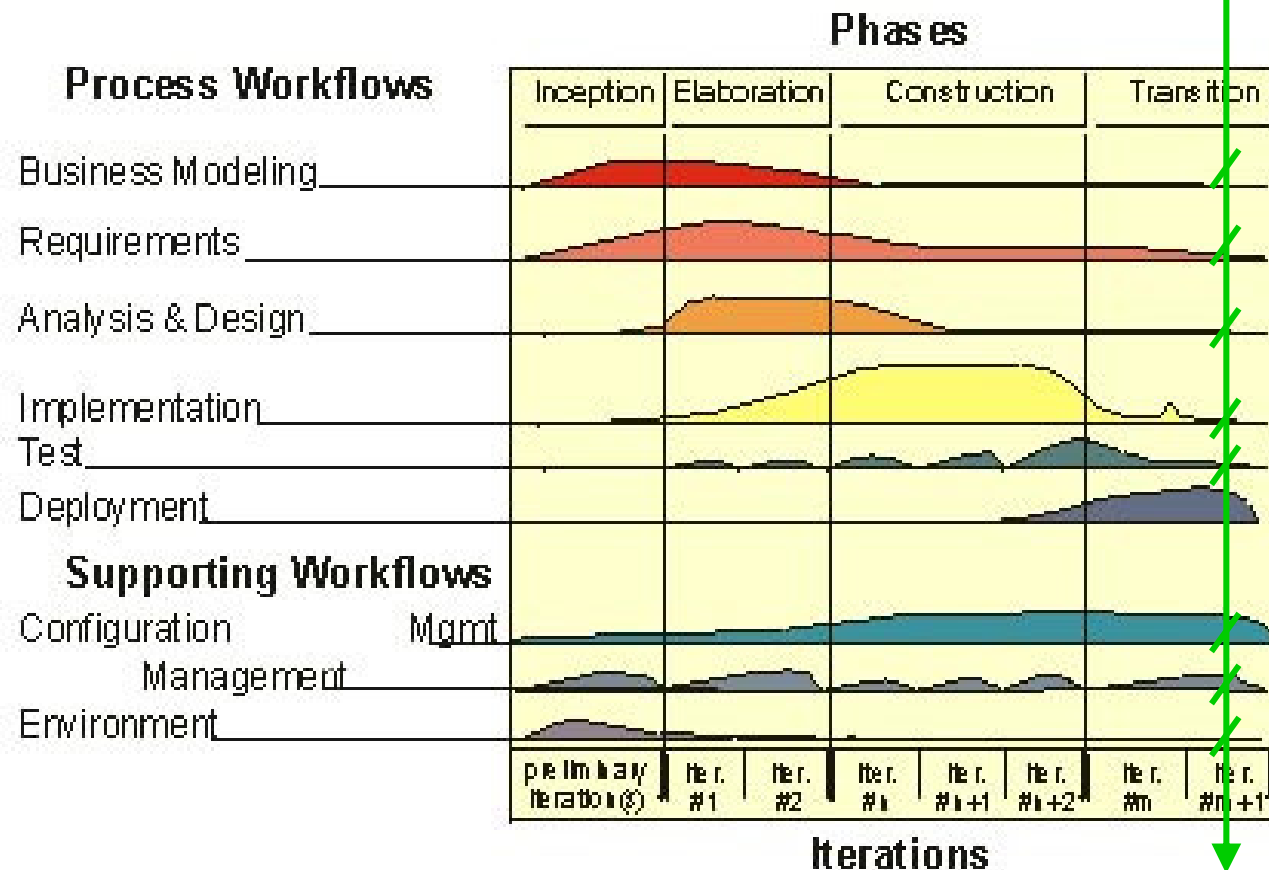
# Rational Unified Process (RUP)



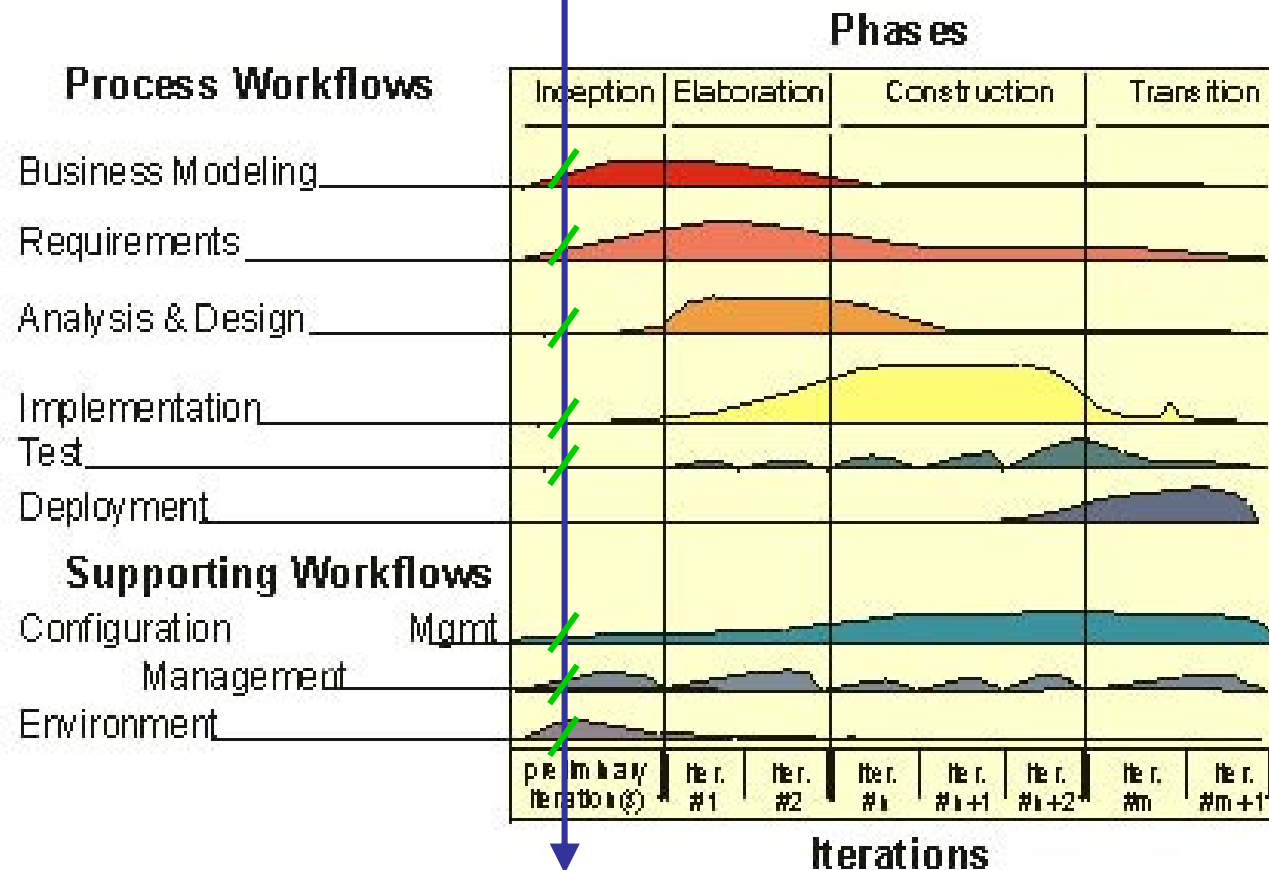
# Rational Unified Process (RUP)



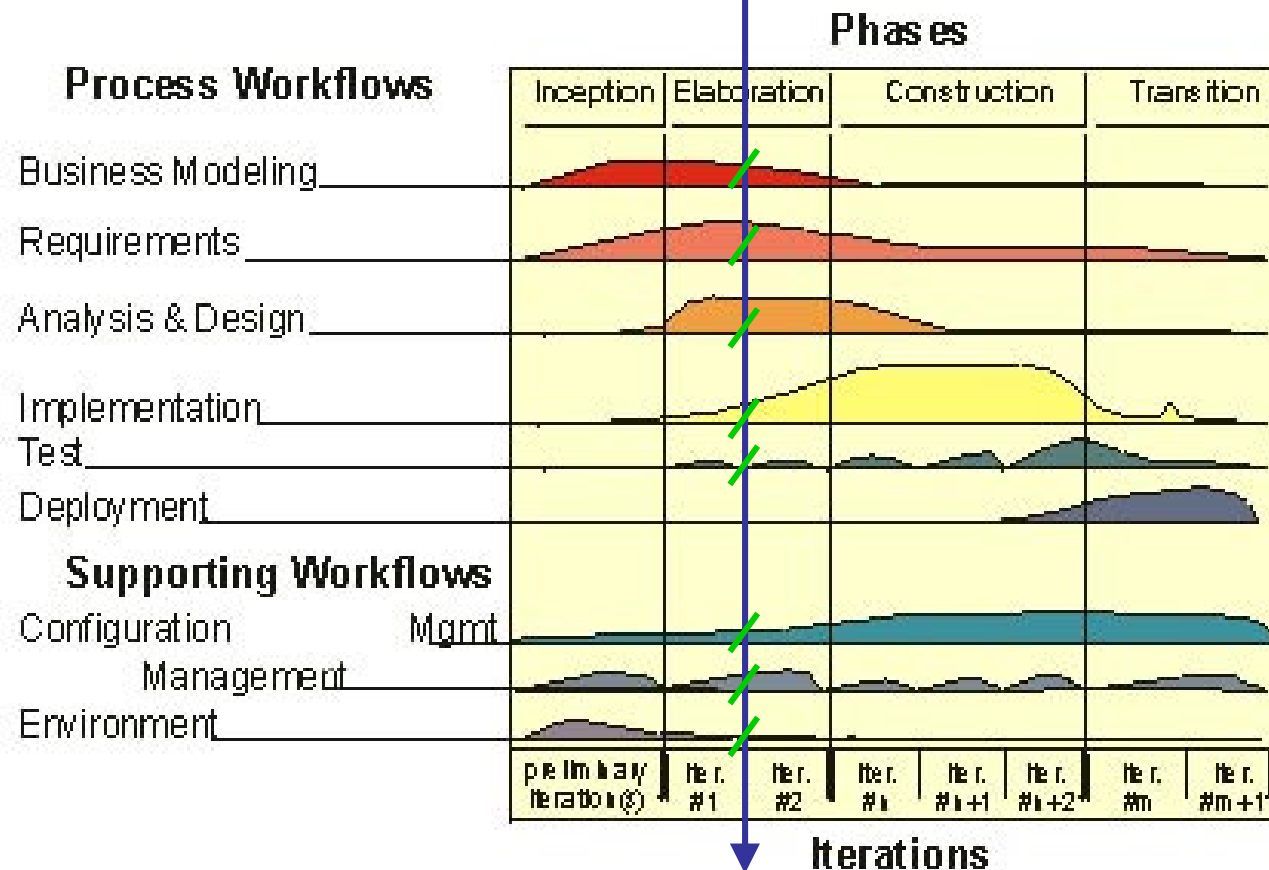
# Rational Unified Process (RUP)



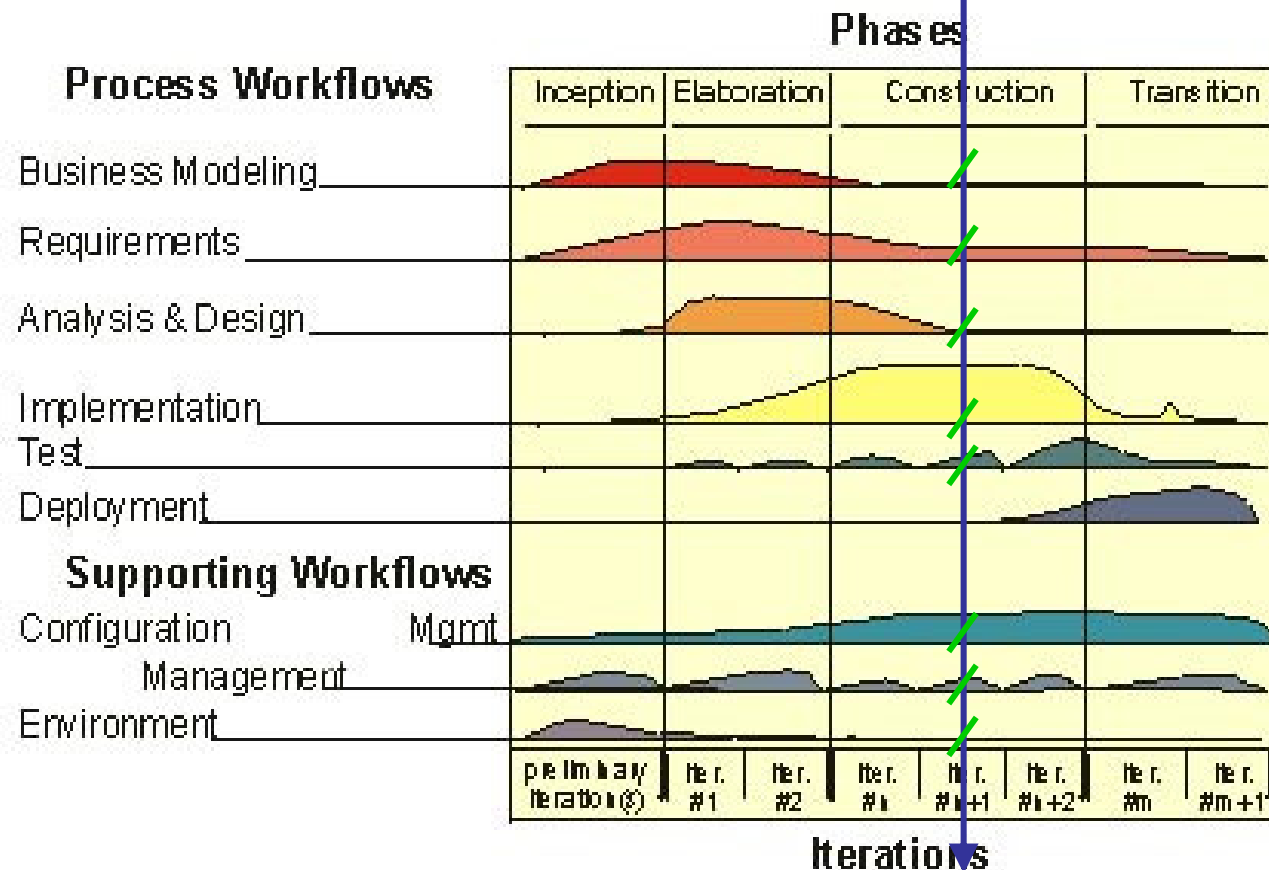
# Rational Unified Process (RUP)



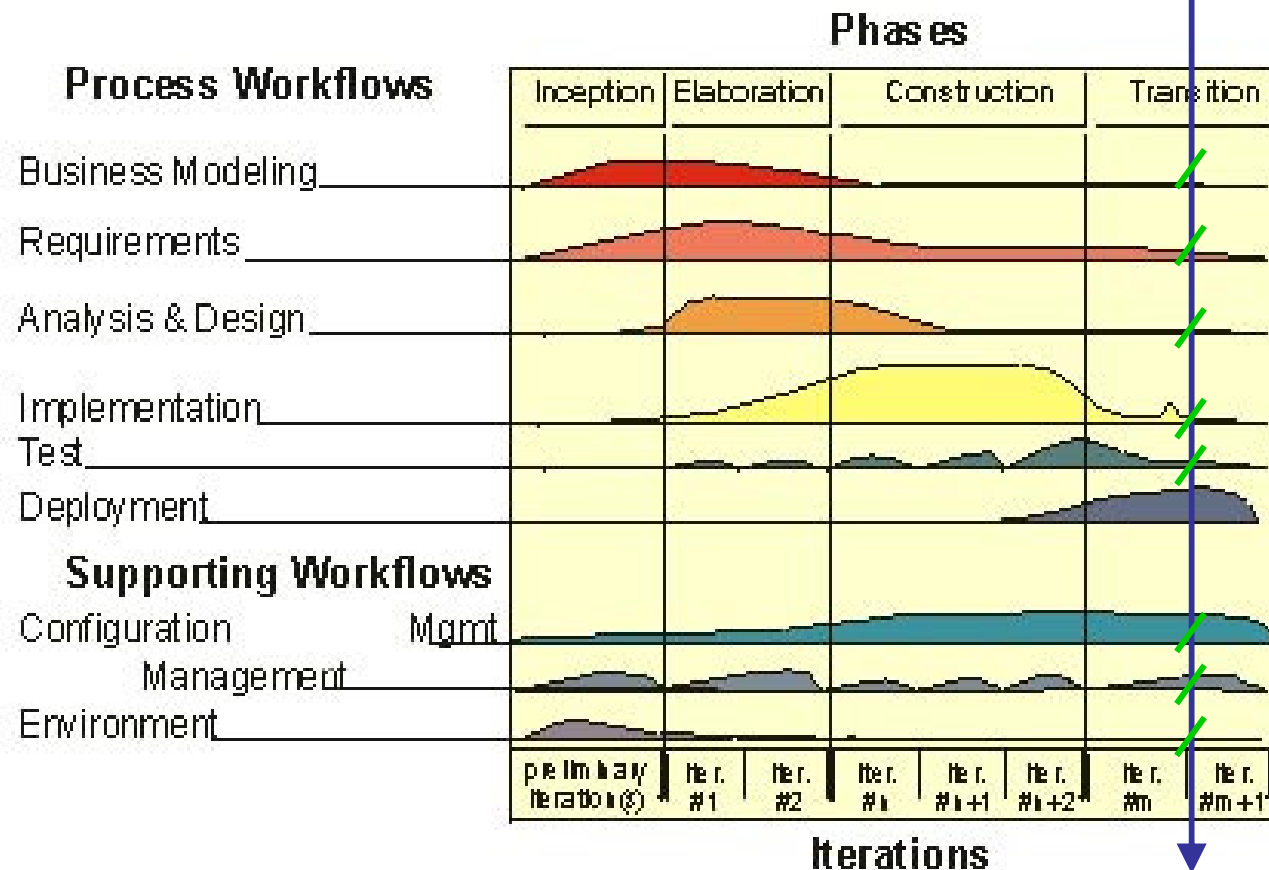
# Rational Unified Process (RUP)



# Rational Unified Process (RUP)

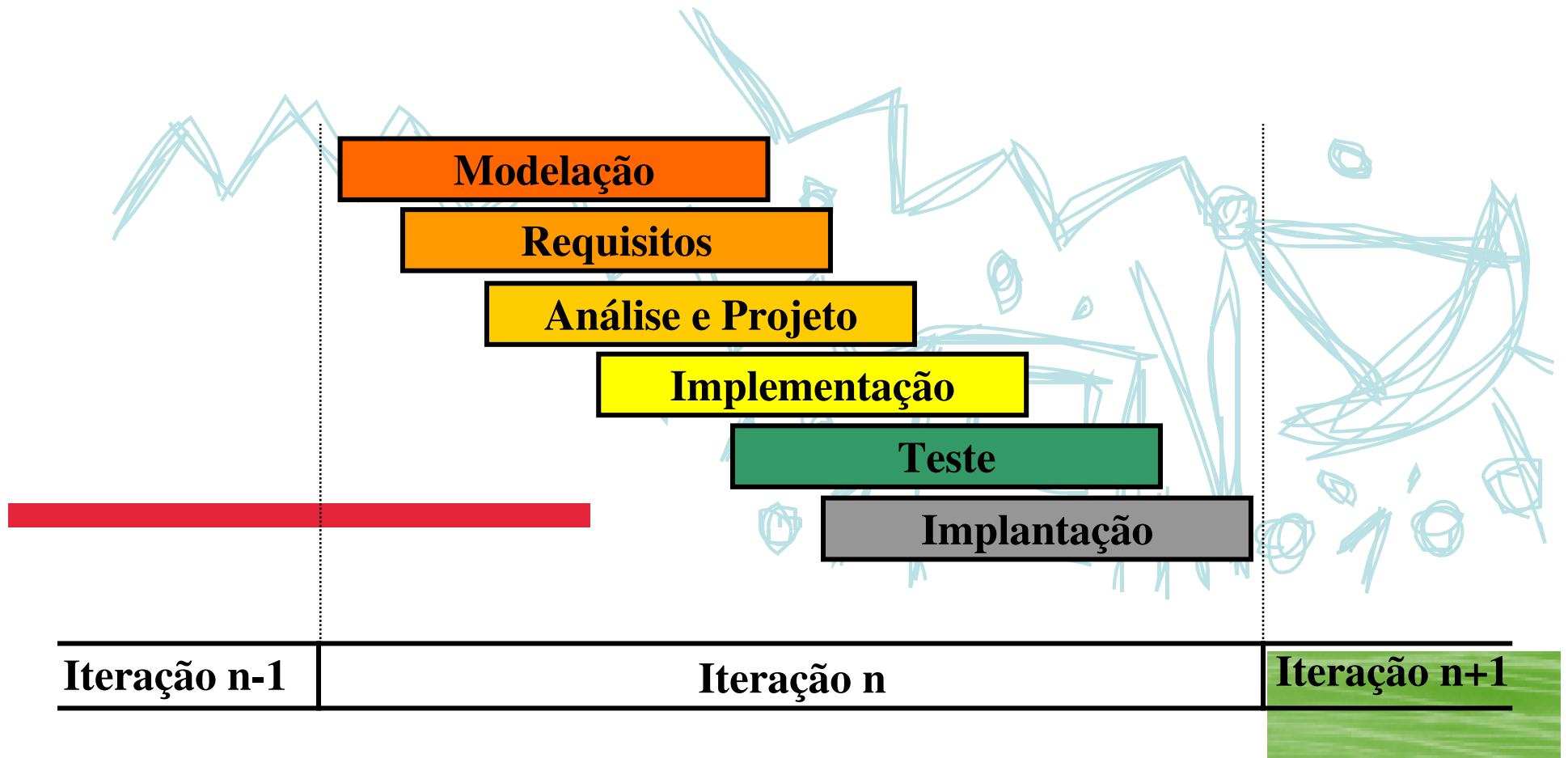


# Rational Unified Process (RUP)



# Rational Unified Process (RUP)

Iterações como mini-cascatas...





# Rational Unified Process (RUP)

Duração influenciada por:

- tamanho da organização (**maior duração**)
- tamanho do projecto (**maior duração**)
- familiaridade com o projecto e maturidade (**menor duração**)
- simplicidade técnica (**menor duração**)

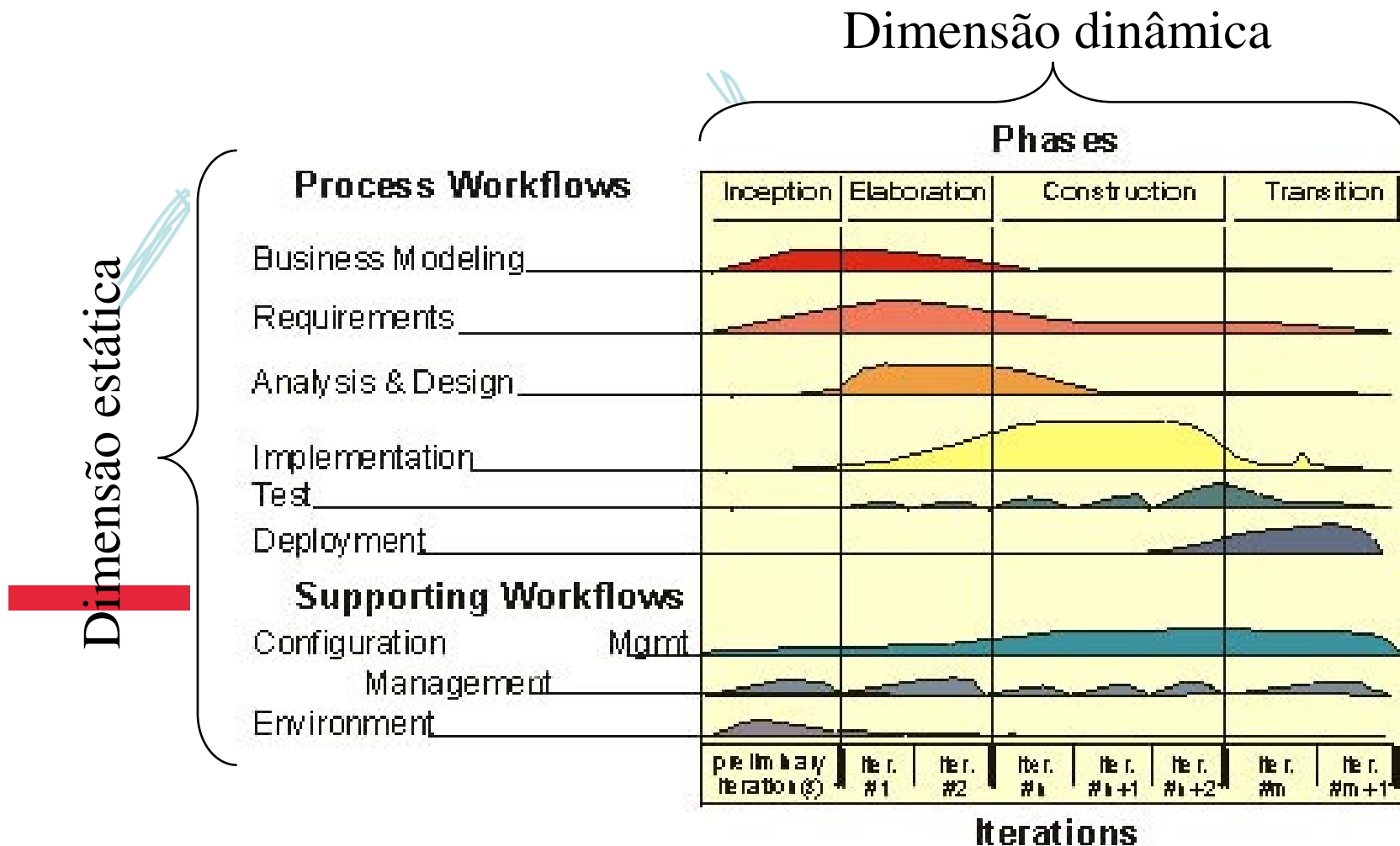
6 mais ou menos 3 iterações:

- Concepção: 0 a 1
- Elaboração: 1 a 3
- Construção: 1 a 3
- Transição: 1 a 2

# Rational Unified Process (RUP)

- Duas dimensões
  - Dinâmica
    - Desenrolar do processo com o tempo – ciclos, fases, iterações e marcos
  - Estática
    - Organização do processo em termos de actividades, artefactos, intervenientes e fluxos de trabalho

# Rational Unified Process (RUP)

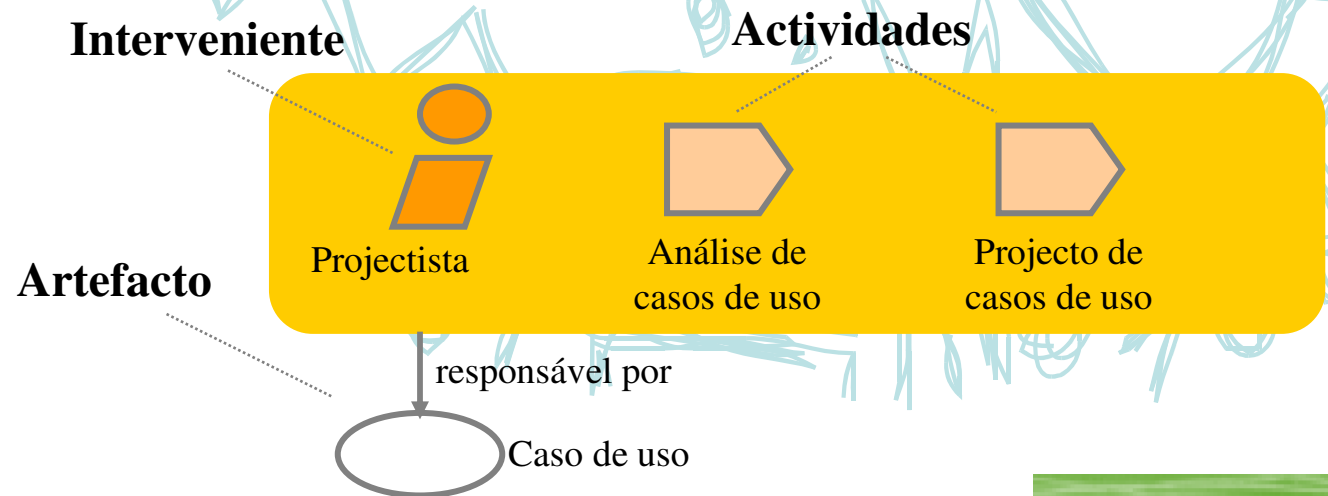


# Rational Unified Process (RUP)

***Quem está a fazer o quê, como e quando?***

# Rational Unified Process (RUP)

- Intervenientes (workers) - quem
- Actividades (activities) - como
- Artefactos (artifacts) - o quê
- Fluxos de trabalho (workflows) - quando



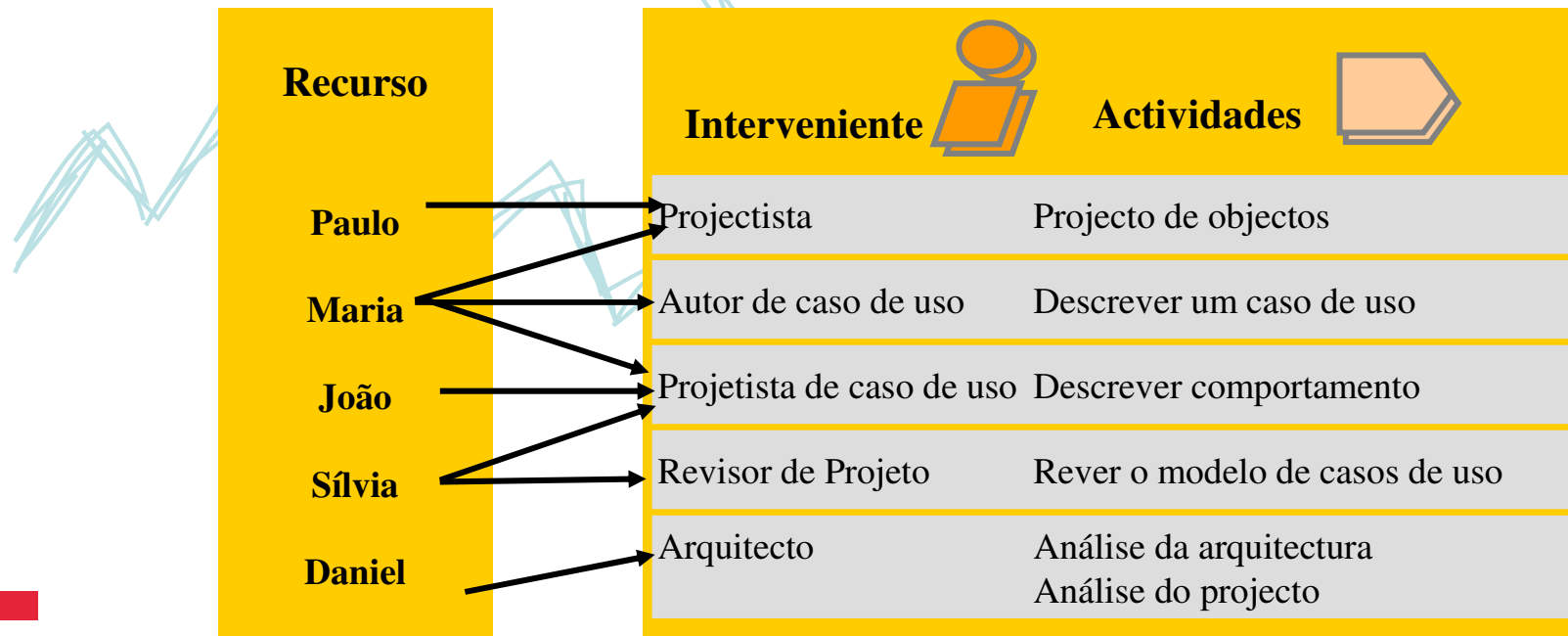
# Rational Unified Process (RUP)

## Intervenientes (*workers*)

- Um interveniente (*worker*) define o **comportamento e responsabilidades** de um **indivíduo**, ou um grupo que trabalha como uma **equipa**.
- Um interveniente é um “chapéu” que um indivíduo usa no projecto. Um indivíduo pode usar mais do que um chapéu (ou desempenhar mais que um papel)
- As **responsabilidades** atribuídas a um interveniente incluem tanto a **execução** de um conjunto de **actividades** como a **titularidade** de certos **artefactos**

# Rational Unified Process (RUP)

## Intervenientes (*workers*)



Pessoas e intervenientes

# Rational Unified Process (RUP)

## Actividades (*activities*)

- É uma **unidade de trabalho** que pode ser solicitada a um indivíduo num dado **papel**
- Tem um **objectivo claro**, geralmente expresso em termos de **produção** ou **actualização** de um **artefacto** (modelo, classe, plano...)
- Todas as **actividades** têm atribuído um **interveniente específico**
- A granularidade pode ser de horas a dias
- Normalmente envolve um interveniente e afecta um ou artefacto ou um conjunto reduzido



# Rational Unified Process (RUP)

## Actividades (*activities*)

- Deve ser utilizável como elemento de planeamento e aferição de progresso (demasiado pequena – negligenciada; demasiado grande – implica progresso avaliado em partes)
- Exemplos:
  - ***Planear uma iteração*** (interveniente: gestor de projecto)
  - ***Identificar casos uso e actores*** (interveniente: analista)
  - ***Rever o projecto*** (interveniente: revisor de projecto)
  - ***Executar testes de desempenho*** (interveniente: testador)

# Rational Unified Process (RUP)

## Artefactos (*artifacts*)

- É um elemento (de informação) produzido, modificado ou utilizado por um processo
- São os elementos tangíveis de um projecto (as coisas produzidas ou utilizadas ao trabalhar para o objectivo último)
- São utilizados como “entrada” pelos ***intervenientes*** para desempenhar uma ***actividade*** e são o resultado ou “saída” de tais actividades

# Rational Unified Process (RUP)

## Artefactos (*artifacts*)

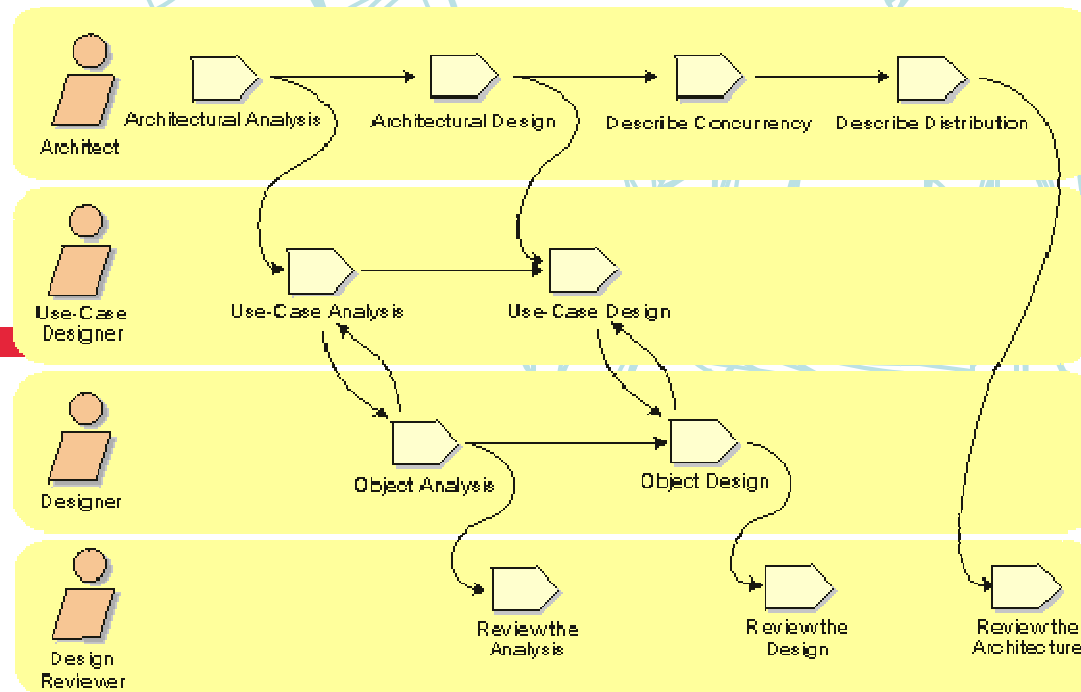
### – Exemplos:

- Um modelo de casos de uso
- Um elemento de um modelo – uma classe
- Um documento – *business case* ou arquitectura do s/w
- Código fonte
- Executáveis (binários)

# Rational Unified Process (RUP)

## Fluxos de trabalho (*workflows*)

- ***Sequências de actividades*** que conduzem a um resultado válido (e observável) e que demonstram as ***interacções*** entre ***intervenientes***



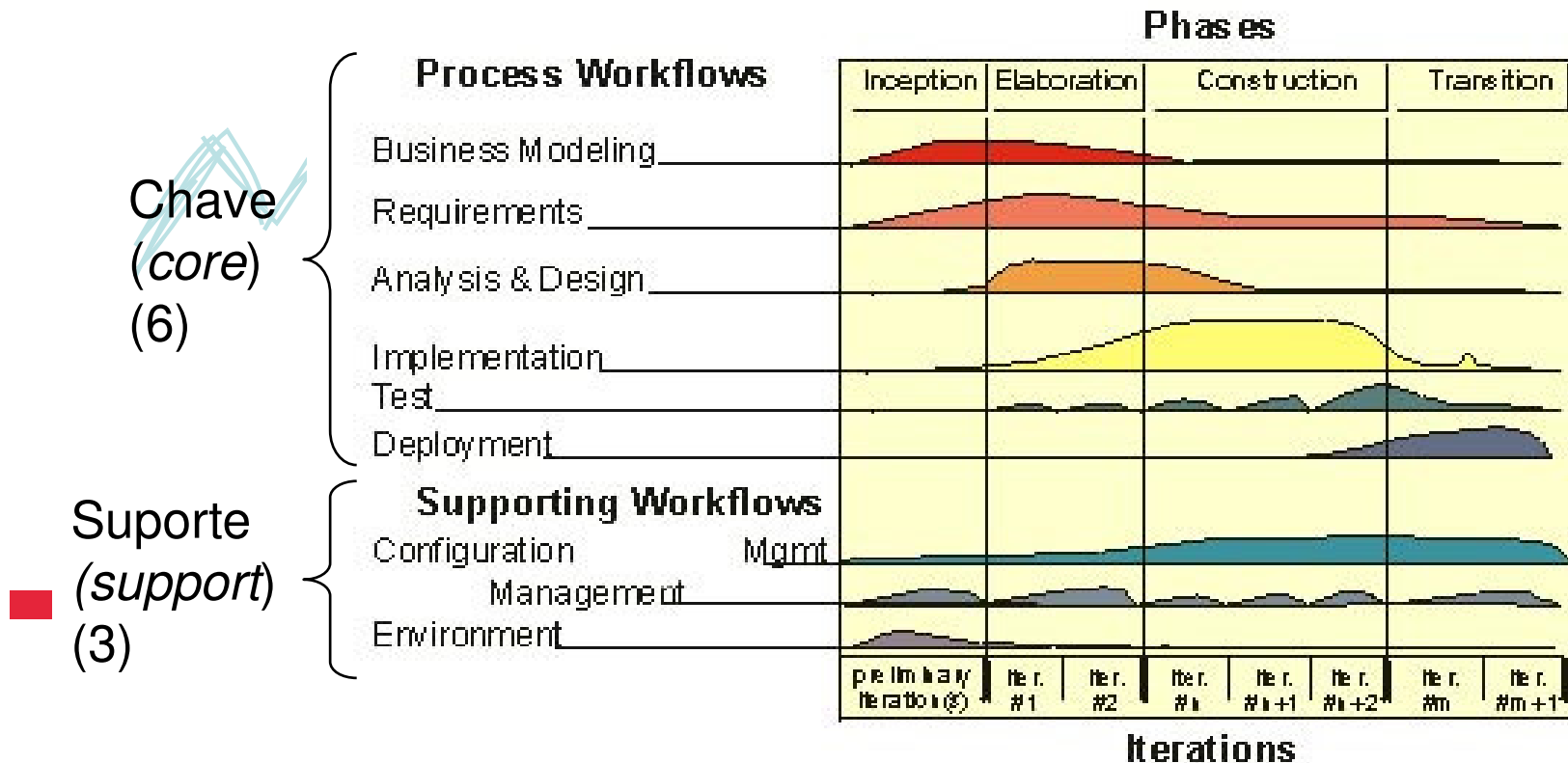
# Rational Unified Process (RUP)

## Fluxos de trabalho (*workflows*)

- Nem sempre é possível ou prático representar todas as dependências entre actividades. Frequentemente duas ou mais actividades estão mais interligadas do que é mostrado, especialmente quando envolvem o mesmo interveniente ou o mesmo indivíduo.
- ***Um workflow não é um “programa” para humanos “correrem”***

# Rational Unified Process (RUP)

- **Fluxos de trabalho *chave* e de *suporte***



# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Modelação do negócio (*business modeling*)
  - Tentativa de fazer as comunidades de modelação (engenharia) do negócio e de modelação (engenharia) de software falarem a mesma linguagem
  - Documentação (descrição) dos **processos de negócio** recorrendo a **casos de uso** para compreensão comum...
  - Muitos processos não fazem modelação do negócio

# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Requirements (*requisitos*)
  - O objectivo é descrever **o que** o sistema deve fazer
  - Que deverá ter o acordo do cliente e produtor
    - É criado um documento de **visão** e as necessidades dos *stakeholders* identificadas
    - Os actores são identificados
    - Casos de uso são identificados – comportamento do sistema – e descritos em detalhe



# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Análise & projecto (*analysis & design*)
  - O objectivo é mostrar **como** to sistema vai ser criado na fase de **implementação**
  - Resulta num **modelo de projecto** (*design model*) e, opcionalmente, num **modelo de análise** (*analysis model*)
  - O **modelo de projecto** (*design model*) serve como abstracção do código fonte – **mapa do código fonte** a escrever (classes, packages...)

# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Implementação (*implementation*)
  - Objectivos:
    - Definir a organização do código fonte
    - Implementar classes em termos de componentes (fontes, binários, executáveis, outros)
    - Testar cada componente desenvolvido
    - Integrar as várias contribuições formando um sistema executável
  - ***O RUP enfatiza a implementação com recurso a componentes***

# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Teste (*test*)
  - Objectivos:
    - Verificar a interacção entre objectos
    - Verificar a correcta integração de todos os componentes de *software*
    - Verificar que todos os requisitos foram correctamente implementados
    - Identificar e garantir que os defeitos são tratados antes da implantação do *software*

# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Teste (*test*)
  - A abordagem iterativa do RUP significa que os testes são feitos ao longo de todo o projecto (o que permite detectar erros mais cedo e reduzir o custo de os corrigir)
  - Os testes são realizados tendo em conta 3 dimensões:
    - Fiabilidade
    - Funcionalidade
    - Desempenho (aplicacional e do sistema)

# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Implantação (*deployment*)
  - O objectivo desta fase é realizar disponibilizações do software aos utilizadores finais
  - Envolve actividades como:
    - Efectuar lançamentos externos do software
    - Embalar o software
    - Distribuir o software
    - Instalar o software
    - Proporcionar ajuda e assistência aos utilizadores

# Rational Unified Process (RUP)

- Fluxos de trabalho chave – Implantação (*deployment*)
  - Pode também incluir:
    - Planeamento e orientação de testes Beta
    - Migração de software ou dados existentes
    - Aceitação formal
  - Não obstante muitas actividades de implantação estarem centradas na fase de transição, muitas delas têm que ser incluídas em fases anteriores para preparar para a implantação depois da fase de construção

# Rational Unified Process (RUP)

- Fluxos de trabalho de suporte – Gestão de configuração (e mudança) (*configuration (& change) management*)
  - Este fluxo de trabalho descreve como controlar os numerosos artefactos produzidos pelos vários indivíduos que trabalham num mesmo projecto. Ajuda a evitar confusões dispendiosas e assegura que os artefactos resultantes não sofrem de conflitos
  - Que se podem dever a...

# Rational Unified Process (RUP)

- Fluxos de trabalho de suporte – Gestão de configuração (e mudança) (*configuration (& change) management*)
  - **Actualizações simultâneas**: quando duas ou mais pessoas trabalham sobre um mesmo artefacto e as alterações feitas por uma se sobrepõem às realizadas por outra
  - **Notificações limitadas**: quando é resolvido um problema num artefacto partilhado e nem todos os indivíduos são notificados
  - **Múltiplas versões**: uma vez que grandes projectos são desenvolvidos de forma evolutiva, é necessário garantir que correcções numa versão são propagadas às restantes (ex.: vCliente, vTeste, vDesenvolvimento)



# Rational Unified Process (RUP)

- Fluxos de trabalho de suporte – Gestão de configuração (e mudança) (*configuration (& change) management*)
  - Também abrange gestão de pedidos de mudança – como reportar defeitos, geri-los ao longo do ciclo de vida e como usar informação sobre os defeitos para acompanhar o progresso e tendências

# Rational Unified Process (RUP)

- Fluxos de trabalho de suporte – Gestão (de projecto) ( *(project) management*)
  - Gestão de projectos de software é a arte de equilibrar objectivos contraditórios, gerir o risco e superar as dificuldades para disponibilizar um produto que satisfaz as necessidades dos utilizadores
  - Este fluxo de trabalho centra-se especificamente no aspecto do desenvolvimento iterativo, proporcionando referenciais de gestão, orientações práticas para planeamento, contratação, execução e monitorização de projectos e ainda um referencial para gestão de risco

# Rational Unified Process (RUP)

- Fluxos de trabalho de suporte – Ambiente (*environment*)
  - Tem como objectivo proporcionar à organização o ***ambiente de desenvolvimento do software*** (tanto processo como ferramentas)
  - Centra-se nas actividades de configuração do processo no contexto do projecto, bem como nas actividades para desenvolvimento das linhas de orientação para suporte ao projecto

# Rational Unified Process (RUP)

- Fluxos de trabalho de suporte – Ambiente (*environment*)
  - É disponibilizado um procedimento passo-a-passo descrevendo como implementar um processo numa organização
  - Contém o **development kit** que disponibiliza linhas de orientação, documentos padrão e ferramentas necessários à personalização do processo