



Rational Unified Process

Introdução

Fernando Pedrosa – fpedrosa@gmail.com

Bibliografia

- ▶ Phillip Kruchten – Rational Unified Process Made Easy. Addison Wesley
- ▶ www.ibm.com (RMC)
- ▶ www.wthreex.com/rup

Des. de SW – Principais Problemas

Segundo Kruchten:

- ▶ Necessidades do usuário mal compreendidas
- ▶ Falta de habilidade para tratar mudanças de requisitos
- ▶ Descoberta tardia de problemas sérios
- ▶ Baixa qualidade de software
- ▶ Problemas com papéis e responsabilidades

Rational Unified Process (RUP)

- ▶ Criado por Booch, Jacobson e Rumbaugh, e implementado pela Rational
- ▶ Em seu livro, os amigos se referem a ele como Unified Process. RUP é o nome comercial dado pela Rational
- ▶ Em 2003 a IBM compra a Rational. RUP continua sendo, até hoje, o principal *framework* de processos no qual as metodologias se baseiam

O que é ?

- ▶ É uma plataforma de processos
 - Adaptável
 - Deve ser configurada para selecionar os elementos apropriados às necessidades da organização
- ▶ Fornece atividades, artefatos e guias ligados
 - Às ferramentas IBM/Rational
 - À linguagem UML

Características

▶ Iterativo e Incremental

- O ciclo de vida do produto é dividido em iterações, cada uma entregando incrementos (partes acabadas) do software

▶ Guiado por casos de uso

- Os casos de uso conectam todas as fases e visões, sendo utilizados por todos os stakeholders

▶ Centrado na arquitetura

- Envolve aspectos estáticos e dinâmicos
- Evolui a partir das necessidades do produto

Características

▶ Orientado a Objetos

- Componentes são construídos através de Objetos e estes colaboram entre si para realizar os casos de uso

▶ Planejado por riscos

- Os riscos são analisados continuamente e os de maior criticidade são tratados prioritariamente

Exercícios [1]

(CAIXA – CESPE 2010)

[45] Não se utilizam diagramas de caso de uso em projetos desenvolvidos de acordo com o RUP (rational unified process).

(CEHAP – CESPE 2009)

[27A] O RUP foi projetado em conjunto com a UML e os processos de negócios são modelados usando casos de uso que, posteriormente, serão desenvolvidos para modelar os requisitos de sistema.

(TCU – CESPE 2010)

[109] O processo unificado de software é centrado na arquitetura e orientado por casos de uso, o que sugere um fluxo de processo iterativo e incremental.

Exercícios [1]

(PETROBRAS – CESGRANRIO 2008)

[48] Um princípio fundamental do Processo Unificado é

- (A) ser centrado em arquitetura.
- (B) empregar times auto-dirigidos e auto-organizados.
- (C) o desenvolvimento em cascata.
- (D) a programação em pares.
- (E) a propriedade coletiva do código fonte.

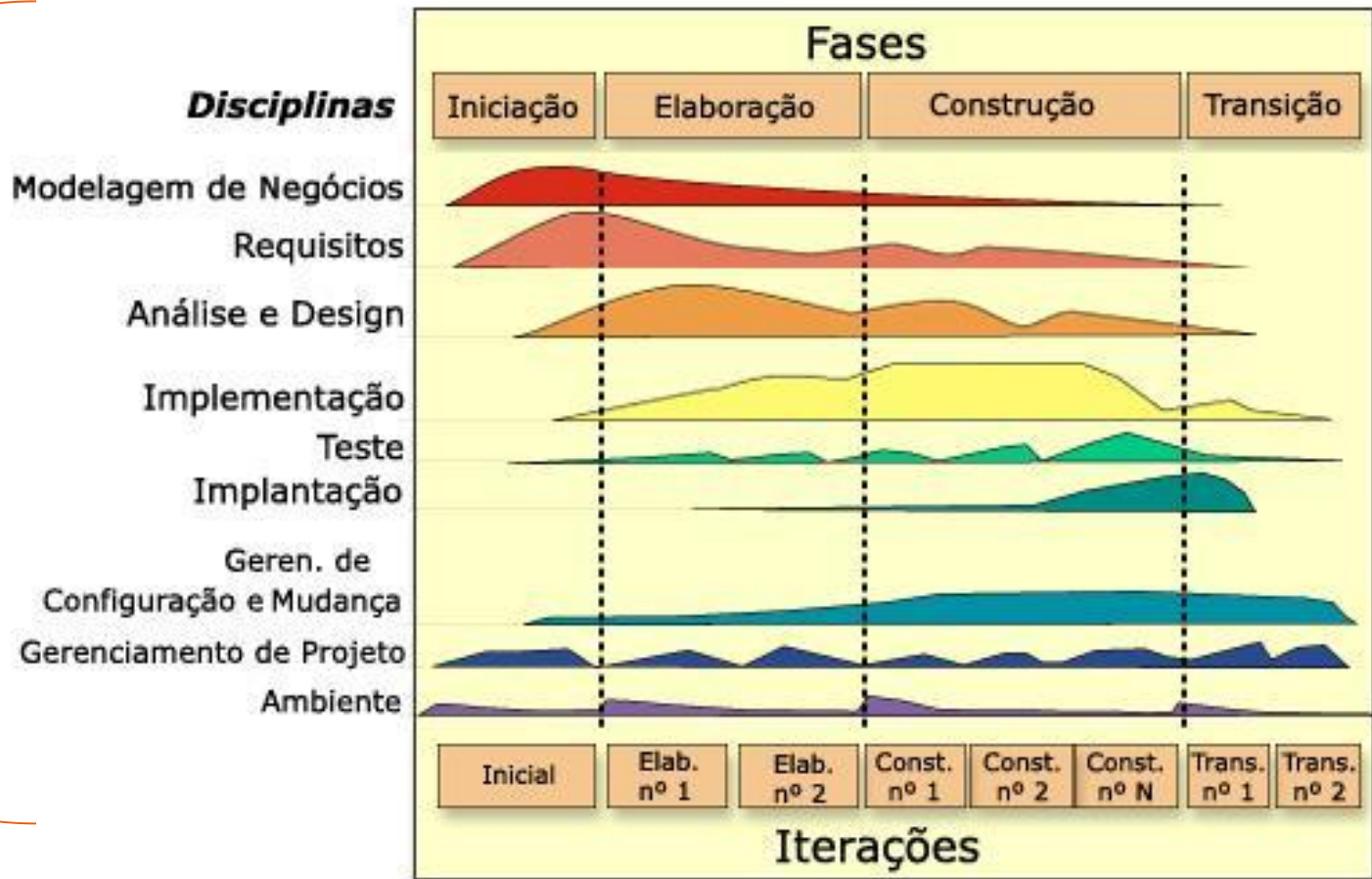
(BASA – CESPE 2010)

[80] A metodologia RUP, que consiste no desenvolvimento interativo com foco na redução dos riscos do projeto, agrega um valor real à organização que necessita manter padrões relativos às comunicações externas e à comunicação com a equipe de desenvolvimento.

Gráfico das Baleias

Eixo
dinâmico

Eixo
estático



Dimensões

O RUP tem duas dimensões

- ▶ A primeira dimensão representa o aspecto dinâmico do processo
 - Eixo horizontal
 - Expresso em termos de fases, marcos e iterações
- ▶ A segunda dimensão representa o aspecto estático do processo
 - Eixo vertical
 - Expresso em termos de componentes, disciplinas, atividades, artefatos, papéis...

Exercícios [2]

(SECONT/ES – CESPE 2010)

[76] O processo unificado é estruturado em duas dimensões. A dimensão horizontal representa o aspecto dinâmico do processo, onde estão representadas suas fases, às quais estão associados marcos que determinam sua finalização. Na outra dimensão estão representadas as disciplinas, que agrupam logicamente as atividades. É possível haver disciplina que não esteja presente em todas as fases.

(EMBASA – CESPE 2009)

[70] A primeira dimensão do RUP representa o aspecto dinâmico do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos.

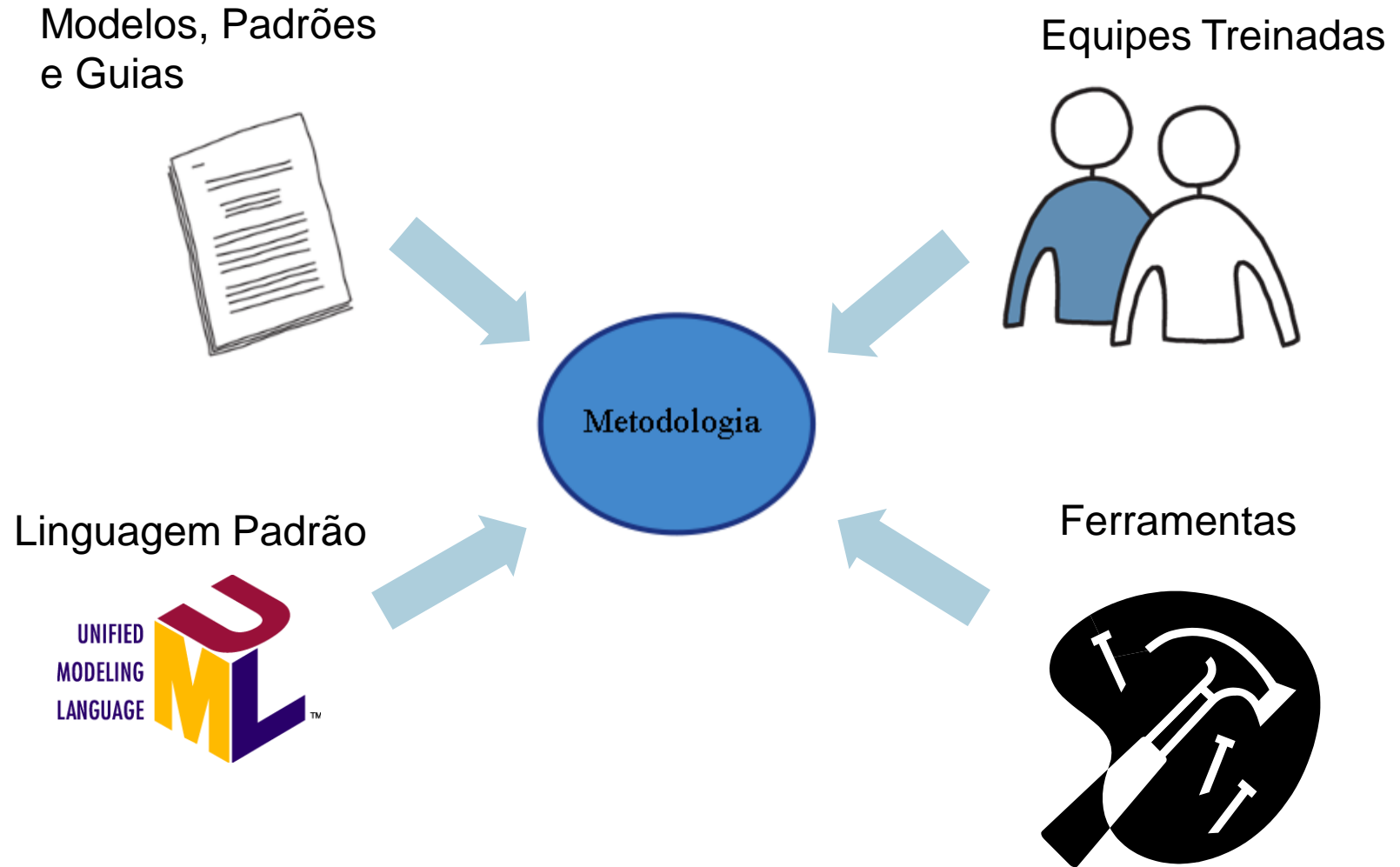
Metodologia baseada no RUP

O que compõe a Metodologia?

- ▶ Processo de Desenvolvimento
- ▶ Conjunto de métodos e práticas bem definidas
 - Com responsáveis
 - Entradas/Saídas
 - Ordem de precedência
- ▶ Inclui:
 - Ferramentas, Tecnologias, Pessoas, Padrões e guias

Quem? O quê? Como? Quando?

Componentes da Metodologia



Benefícios da Metodologia

- ▶ Qualidade de software
- ▶ Maior produtividade
- ▶ Maior previsibilidade
- ▶ Maior controle sobre custos e prazos

Conceitos Chave do RUP

- ▶ Fases e Iterações
- ▶ Disciplinas/Fluxo de Atividades
- ▶ Atividades/Tarefas
- ▶ Artefatos/Produtos de Trabalho
- ▶ Papéis

Fases

Concepção

Estabelecer o escopo, e estimar custos e riscos

Elaboração

Assegurar que os principais riscos foram diminuídos e definir uma arquitetura executável

Construção

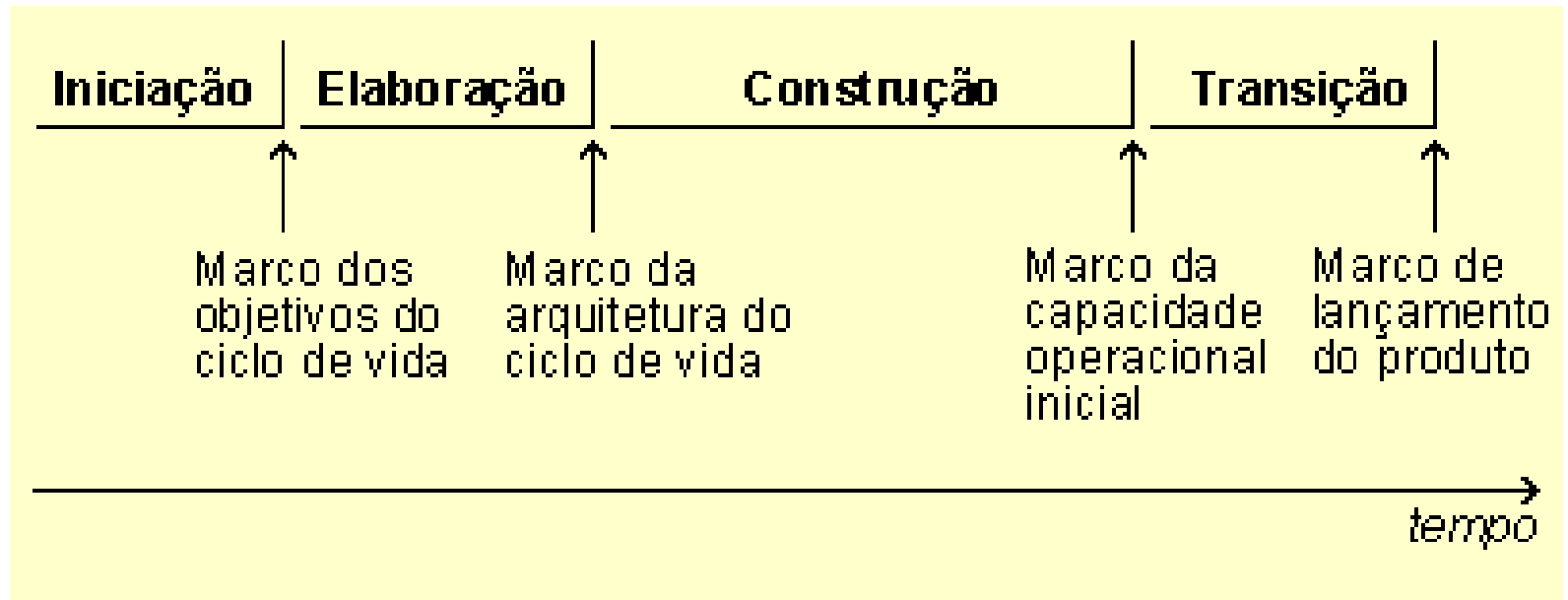
Desenvolver de modo iterativo e incremental um produto completo para a Transição

Transição

Disponibilizar o Software para seus usuários finais

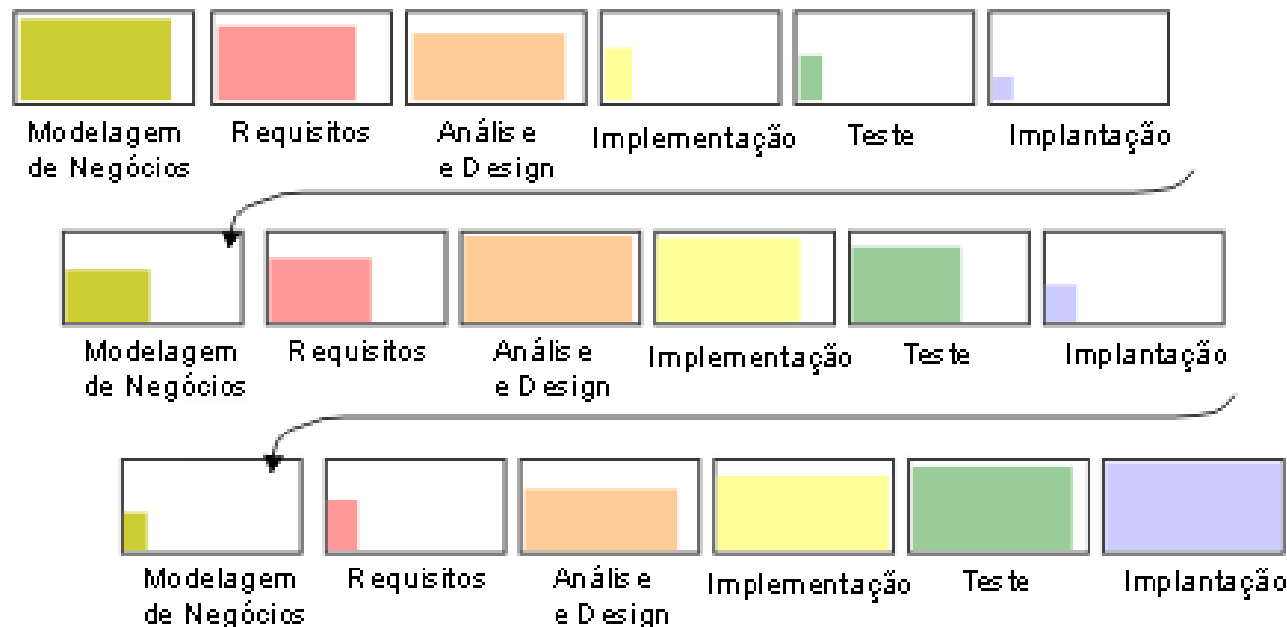
Fases

Cada fase termina com um Marco



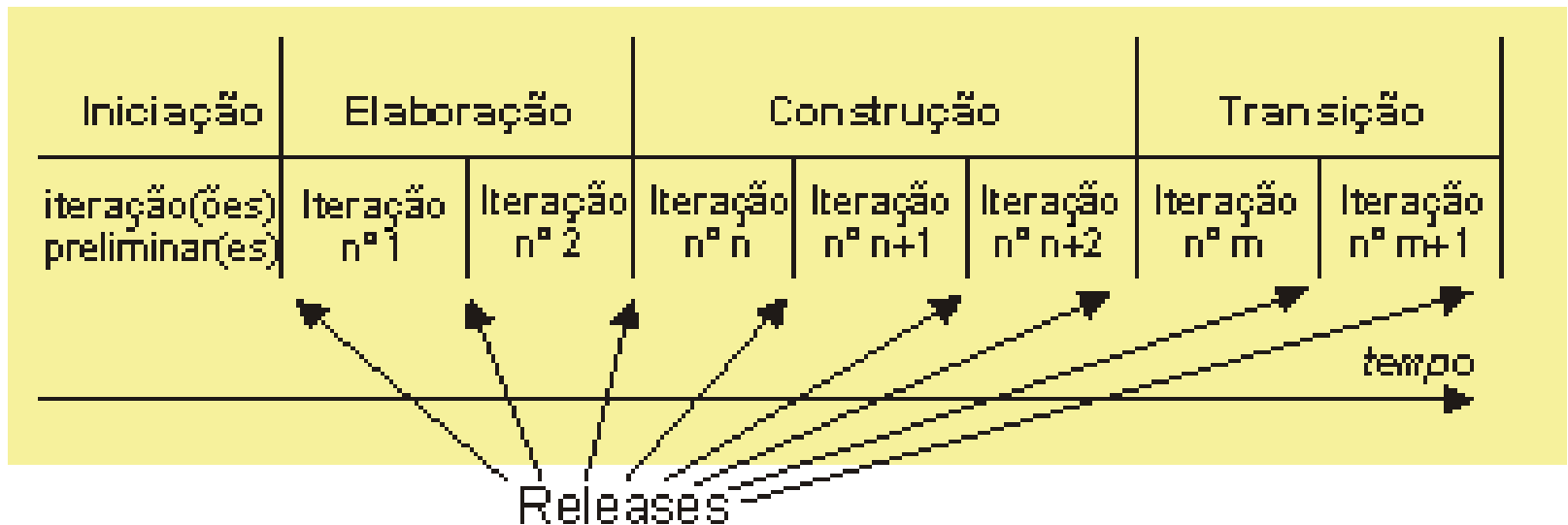
Iterações

Cada passagem pela sequência de disciplinas do projeto se chama **iteração**



Fases e Iterações

Cada fase pode ser dividida em iterações



Exercícios [3]

(SERPRO – CESPE 2010)

[72] O framework de processo RUP (rational unified process) organiza o ciclo de vida de um produto de software desde o início de sua concepção até a sua aposentadoria, na seguinte sequência de etapas: concepção, elaboração, construção e transição

(CEHAP – CESPE 2009)

[27 C] Ao contrário do modelo em cascata, no qual as fases coincidem com as atividades do processo, o RUP compreende as fases de concepção, elaboração, construção e transição.

Exercícios [3]

(MPE/RR – CESPE 2008)

[80] No Processo Unificado, a vida de um sistema é dividida em ciclos; cada ciclo, por sua vez, é dividido em fases e, entre as fases, tem-se a fase Construção, na qual as atividades visam capturar requisitos ainda não capturados na fase anterior e produzir uma arquitetura executável, a ser usada na fase Elaboração.

[81] O Processo Unificado é iterativo e incremental. Ao final de cada iteração, a qual é um miniprojeto, os modelos que representam o sistema encontram-se em um determinado estado, denominado baseline. As atividades de cada fase de um ciclo de vida podem ser distribuídas entre várias iterações.

Exercícios [3]

(ANATEL – CESPE 2006)

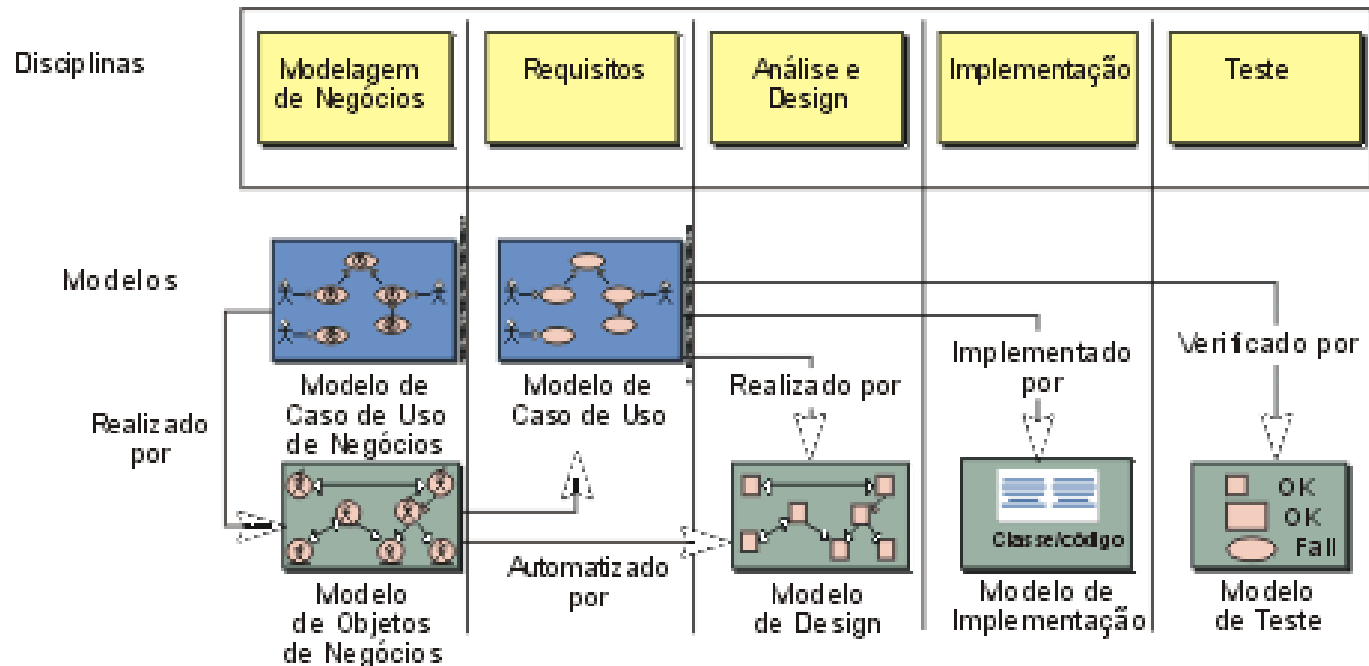
[86] O ciclo apresentado na figura, que compreende uma execução seqüenciada das atividades de modelagem de negócios, requisitos, análise e desenho, implementação, testes, avaliação etc., forma o denominado ciclo de vida de software no modelo RUP.

Disciplinas

- ▶ São um conjunto de atividades (fluxo de trabalho) relacionadas a uma “área de interesse” do projeto
- ▶ Ajudam a compreender o projeto a partir de uma perspectiva em cascata

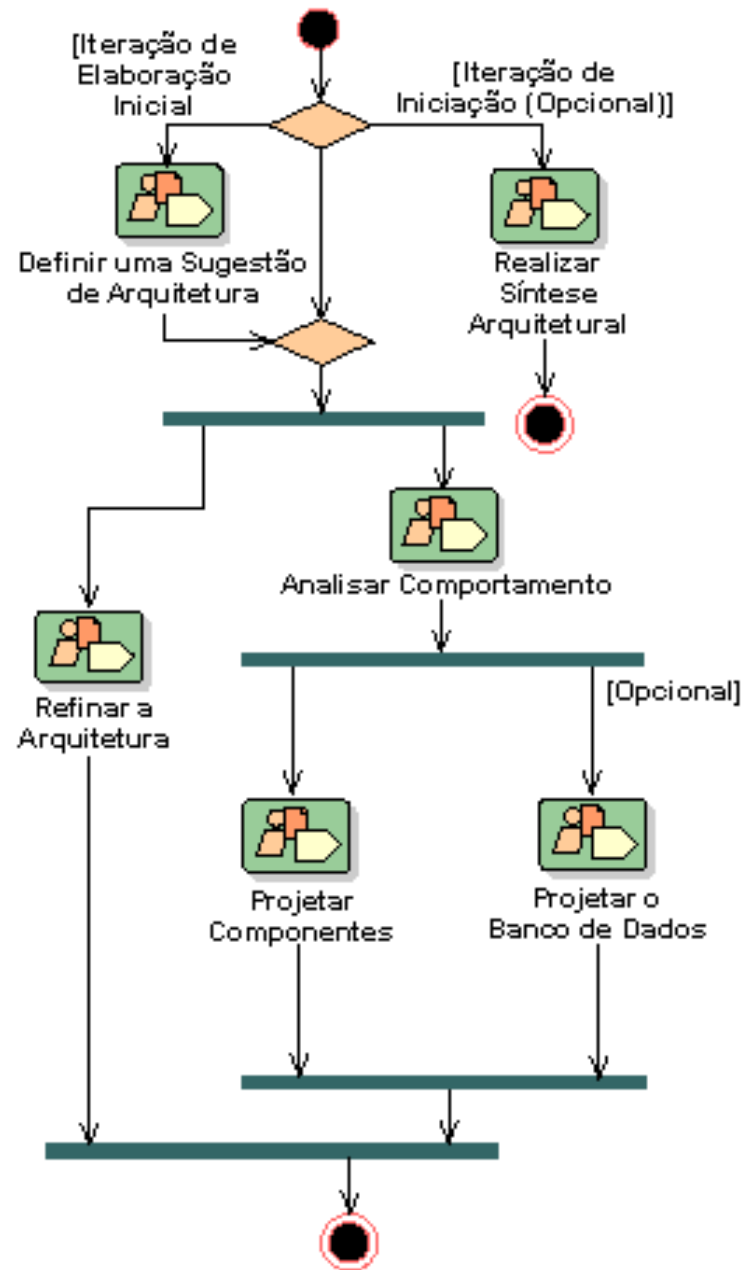
Disciplinas

Algumas disciplinas estão associadas a conjuntos específicos de modelos



Disciplinas

Cada disciplina possui um fluxo de trabalho (ex: Análise e Design)



Disciplinas

▶ Disciplinas básicas

- **M**odelagem de Negócios
- **R**equisitos
- **A**nálise e projeto
- **I**mplementação
- **T**estes
- **I**mplantação

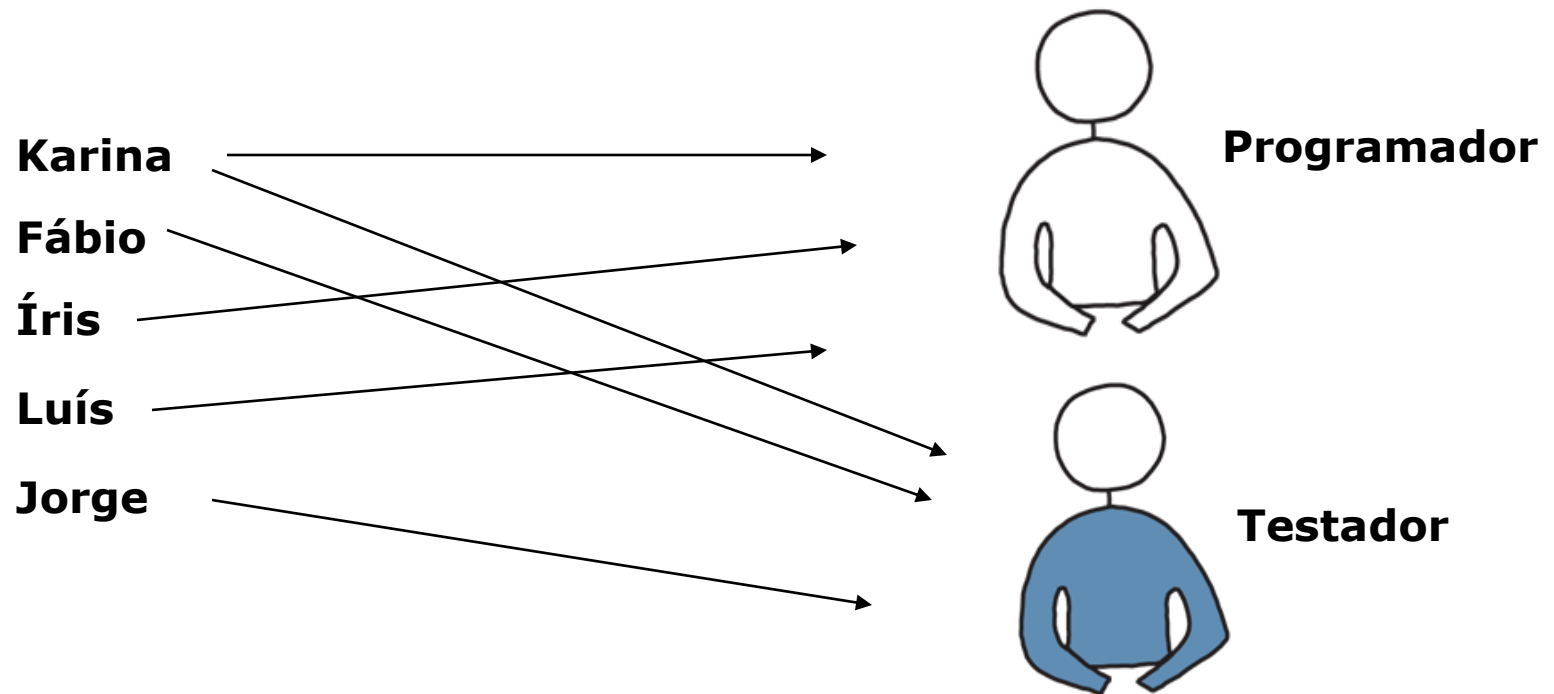
Disciplinas de suporte

- **G**erenciamento de Projeto
- **G**erenc. de configuração e mudanças
- **A**mbiente

MRAITIGGA

Papeis

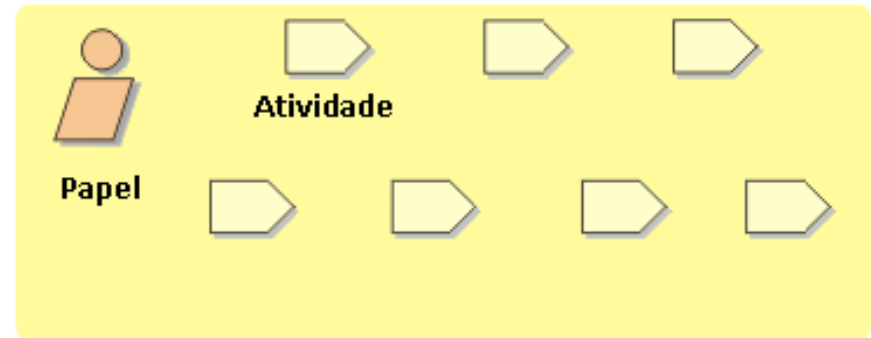
Definem o comportamento e as responsabilidades no processo



Não representam pessoas!

Atividades/Tarefas

- ▶ Unidade de trabalho desempenhada por um papel
- ▶ Inseridas no contexto de uma Disciplina
- ▶ Compostas de:
 - Finalidade
 - Passos
 - Entradas e saídas
 - Papel responsável
 - Guias e padrões



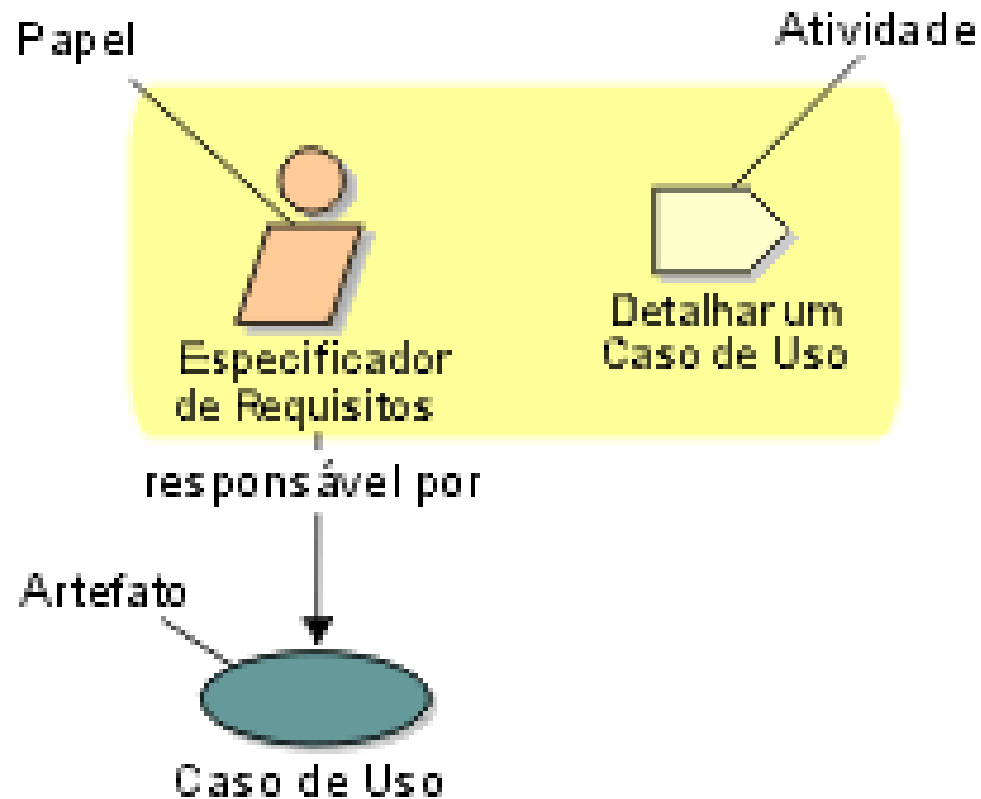
Artefatos / Produtos de Trabalho

- ▶ São o resultado de um processo de trabalho
- ▶ Utilizados como entradas e/ou saídas na execução das atividades
- ▶ Podem ser:
 - Modelos
 - Documentos
 - Código fonte
 - Executáveis, etc...



Em resumo...

- ▶ Papéis executam Atividades que geram Artefatos

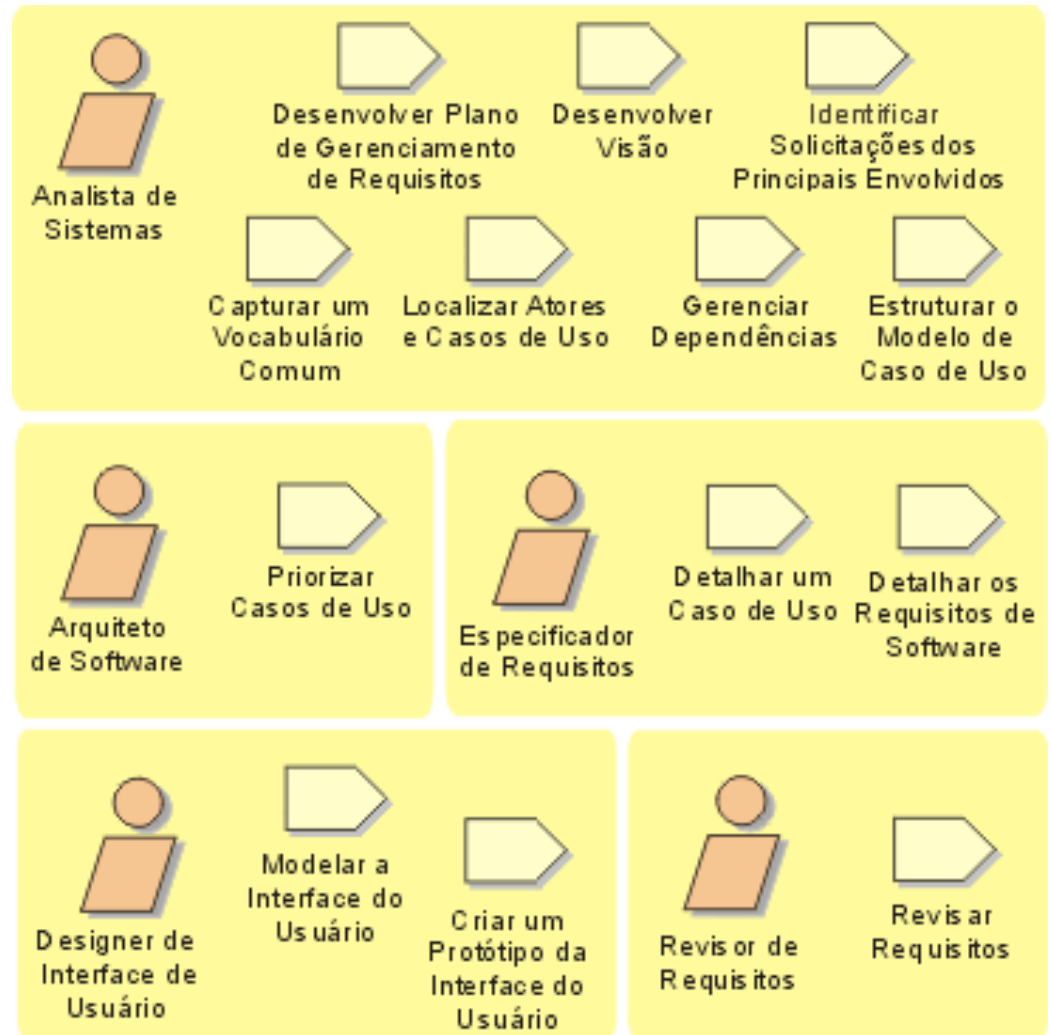


Visão Geral das Atividades

Cada disciplina tem uma visão geral de **atividades** executadas

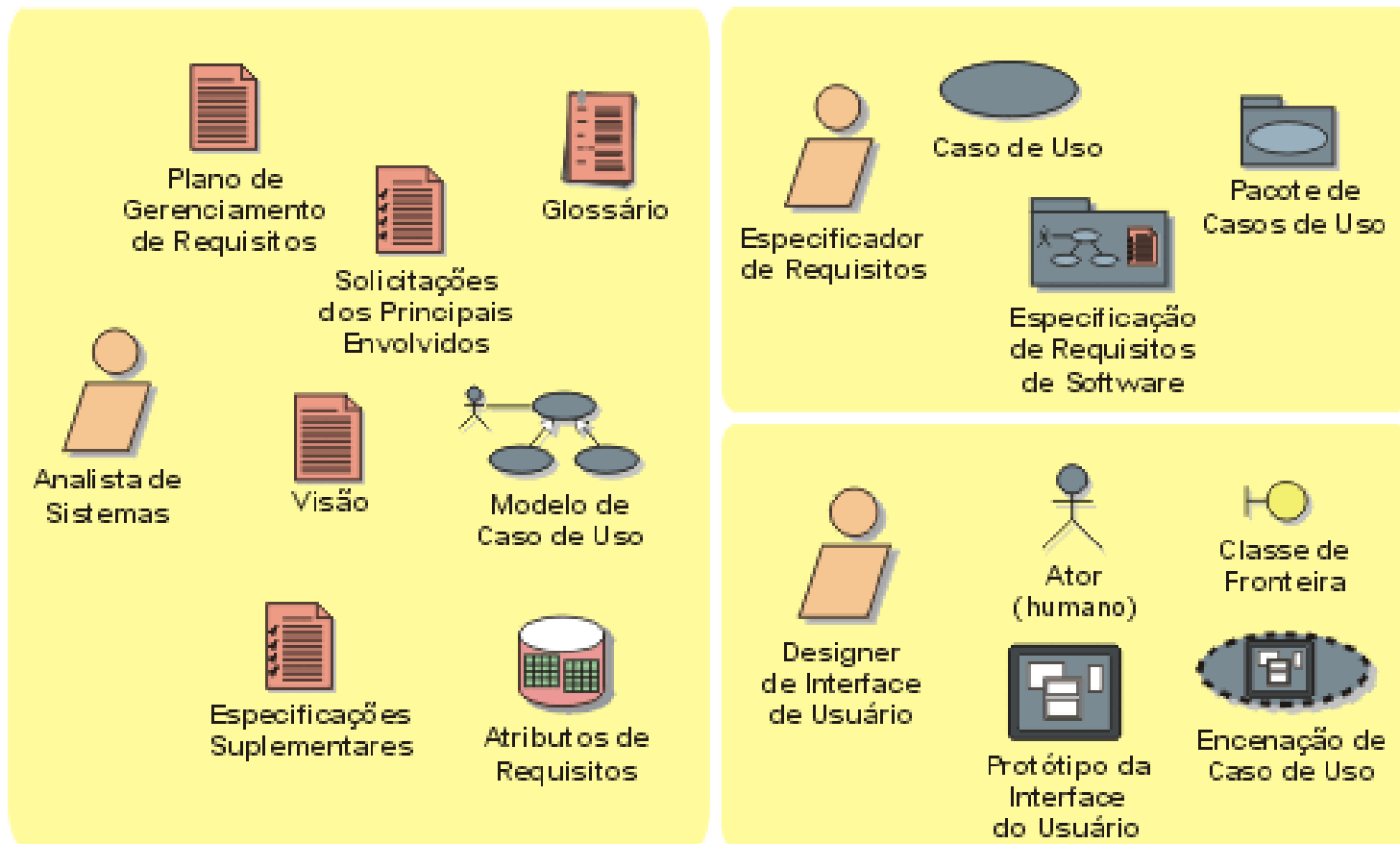
Ex:

Disciplina de Requisitos



Visão Geral de Artefatos

Cada disciplina tem uma visão geral dos **artefatos** relacionados (ex: Requisitos)



Exercícios [4]

(MPE/RR – CESPE 2008)

[79] No Processo Unificado, atividades são organizadas em fluxos de atividades. Algumas atividades produzem artefatos, que podem ser de engenharia ou gerenciais. Entre os artefatos criados, há modelos que visam especificar o sistema a partir de certos pontos de vista e níveis de abstração.

(TRE/MT – CESPE 2010)

[32–C] As disciplinas de suporte (apoio) do RUP são: gerenciamento de classes; gerenciamento de produto; e ambiente.

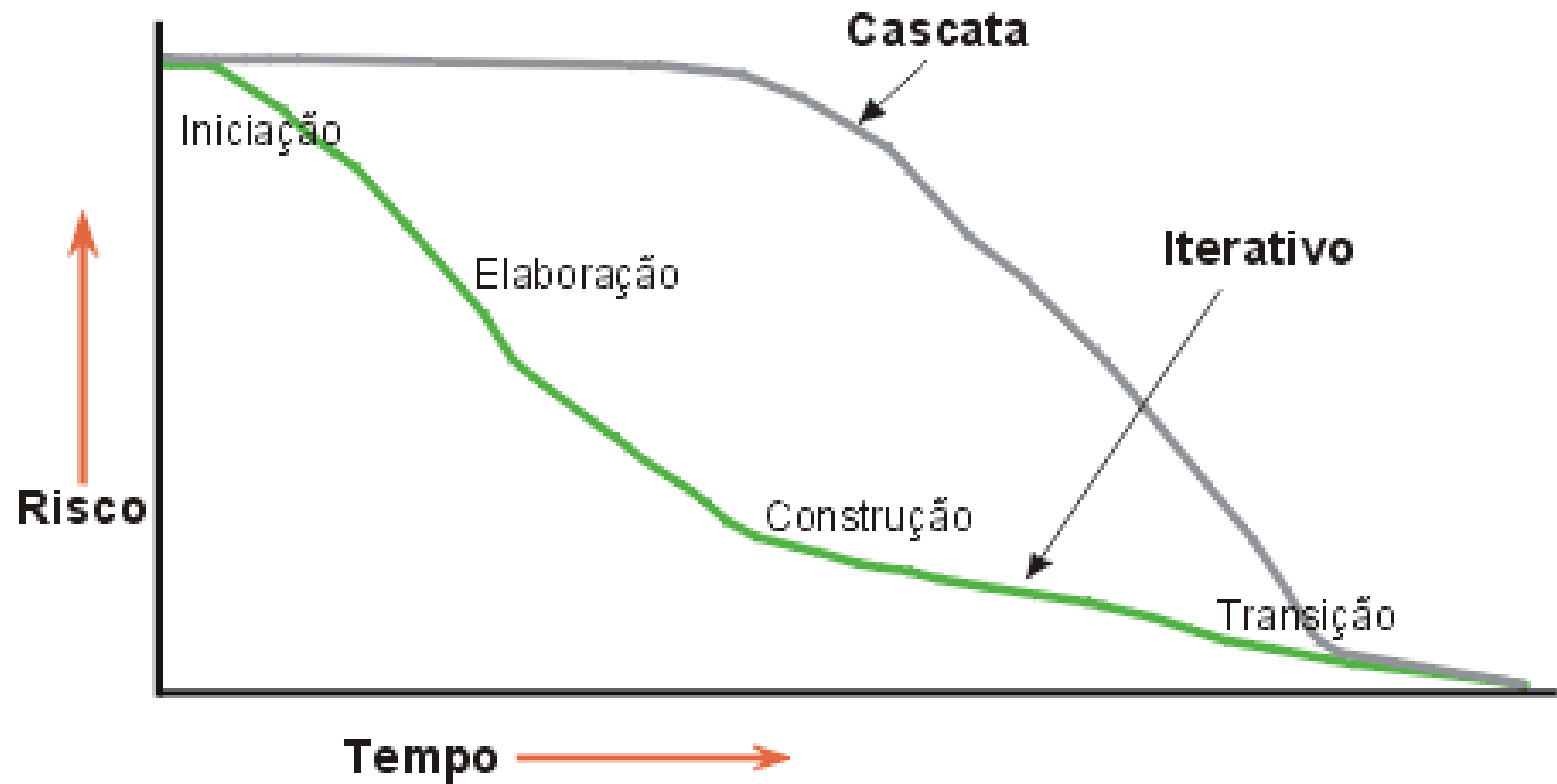
Exercícios [4]

[32-D] Um papel é uma definição abstrata de um conjunto de atividades executadas e dos respectivos artefatos. Exemplos de papéis no RUP são: analistas, desenvolvedores e testadores. Explicitamente, papéis de gerentes não fazem parte dos papéis possíveis no RUP.

[32-E] As disciplinas de engenharia do RUP são: modelagem de negócios; requisitos; análise e projeto; implementação; teste; qualidade; e implantação.

Melhores Práticas: Desenvolvimento Iterativo

Por que desenvolver iterativamente?



Características

- ▶ Desenvolvimento Iterativo lida com mudanças
 - As táticas e os requisitos variáveis são acomodados
- ▶ Diminui riscos
 - Os riscos são reduzidos mais cedo, pois os elementos são integrados progressivamente
- ▶ Busca melhor qualidade
 - A melhoria e o refinamento do produto são facilitados, tornando-o mais robusto

Características

- ▶ Aprende e melhora
 - As organizações podem aprender a partir dessa abordagem e melhorar seus processos
- ▶ Aumenta o reuso
 - Identificar partes comuns quando estão parcialmente projetadas ou implementadas é mais fácil que identificar todas as semelhanças no início

Iterações

- ▶ Incorporam um conjunto **quase** seqüencial de atividades:



Desenvolvimento Iterativo

“Com a abordagem em cascata, tudo parece bem até quase no final do projeto; às vezes, até a metade da integração. Aí, tudo desmorona. Com a abordagem iterativa, é muito difícil esconder a verdade durante muito tempo” – cliente da Rational



Em resumo, esta é a ideia:

“Jogue fora um pouco do trabalho durante o caminho, para não ter que jogar todo o trabalho fora ao final.” – Grady Booch



Exercícios [5]

(TRE/MT – CESPE 2010)

[44–A] Uma das principais características do RUP é o uso da iteração, que, por meio de refinamentos sucessivos, melhora o entendimento do problema.

[44–C] Pelo fato de o RUP ser muito complexo, seu foco evita a redução dos riscos do projeto. Essa fase é tratada diretamente na UML.

Melhores Práticas: Gerência de Requisitos

Gerenciamento de Requisitos

- ▶ Um **requisito** é uma condição ou uma restrição com a qual o sistema deverá estar em conformidade
- ▶ O gerenciamento de requisitos é uma abordagem sistemática para:
 - Localizar,
 - Documentar,
 - Organizar e Controlar os requisitos variáveis em um sistema.

Problemas com Requisitos

- ▶ Nem sempre os requisitos são óbvios e podem vir de várias fontes.
- ▶ Requisitos relacionam-se entre si, mas deve haver consistência nos relacionamentos
- ▶ Cada requisito é diferente, eles não são igualmente importantes ou fáceis de entender

Problemas com Requisitos

- ▶ Há várias partes interessadas
- ▶ O número de requisitos pode se tornar impossível de gerenciar se eles não forem controlados
- ▶ Os requisitos são alterados

Como gerenciar as dificuldades?

▶ Analise o problema

- Entenda o “problema por trás do problema”
- Estabeleça um vocabulário comum
- Proponha soluções em alto nível

Como gerenciar as dificuldades?

- ▶ **Entenda a necessidade das partes interessadas**
 - Todos querem algo. Determine qual é a melhor “fonte”
 - Utilize técnicas de elicitação de requisitos
 - Faça acordos, balanceie prioridades

Como gerenciar as dificuldades?

▶ Defina o sistema

- Defina o que sistema deve fazer, em termos gerais, utilizando linguagem natural e gráfica

▶ Gerencie o escopo do sistema

- O que está dentro? O que está fora?
- Estabeleça prioridades!

Como gerenciar as dificuldades?

▶ **Gerencie requisitos variáveis**

- Garanta que os requisitos tenham uma estrutura que os tornem facilmente atualizáveis
- Rastreie os requisitos

Casos de Uso guiam o desenvolvimento

- ▶ O RUP recomenda a utilização de Casos de Uso como método para a organização dos requisitos funcionais
- ▶ Em vez de fazer uma lista de requisitos, organize-os na visão de como o usuário poderá utilizar o sistema
 - Utilize Casos de Uso!

Casos de Uso

- ▶ Um caso de uso define um conjunto de cenários
- ▶ Cada cenário descreve o comportamento do sistema em termos de sequências de ações
- ▶ Um caso de uso deve produzir um resultado de valor observável para um **ator**
 - Atores são as entidades que interagem com o sistema

Quem utiliza Casos de Uso?

- ▶ Clientes

- Para entenderem o comportamento do sistema e aprovar o fluxo de eventos

- ▶ Arquitetos de Software

- Para identificar características da arquitetura

- ▶ Analistas, Projetistas, Desenvolvedores

- Para entender o comportamento do sistema e refiná-lo

Quem utiliza Casos de Uso?

- ▶ Testadores
 - Utilizam os casos de uso como base para gerar casos de teste
- ▶ Gerentes
 - Para planejar e acompanhar o progresso do projeto
- ▶ E outras partes interessadas...
- ▶ Por isso se diz que o RUP é guiado por Casos de Uso

Exercícios [6]

(EMBASA – CESPE 2010)

[72] No RUP, os manuais dos sistemas e as rotinas de teste são definidos a partir dos casos de uso. Entretanto, os elementos da arquitetura e a estratégia de implantação do sistema, por se relacionarem com a infraestrutura e não com os requisitos funcionais, não são definidos com base nos casos de uso.

Melhores Práticas: Arquitetura de Componentes

Arquitetura

Segundo a IEEE:

- ▶ “O conceito de mais alto nível de um sistema em seu ambiente”
- ▶ A arquitetura de um sistema é a sua organização ou estrutura de componentes significativos que interagem através de interfaces

Arquitetura

Arquitetura de Software inclui:

- ▶ As decisões significativas sobre a organização de um sistema
- ▶ A seleção de elementos estruturadores e suas interfaces
- ▶ A especificação do comportamento dos elementos do sistema e como eles colaboram entre si

Arquitetura

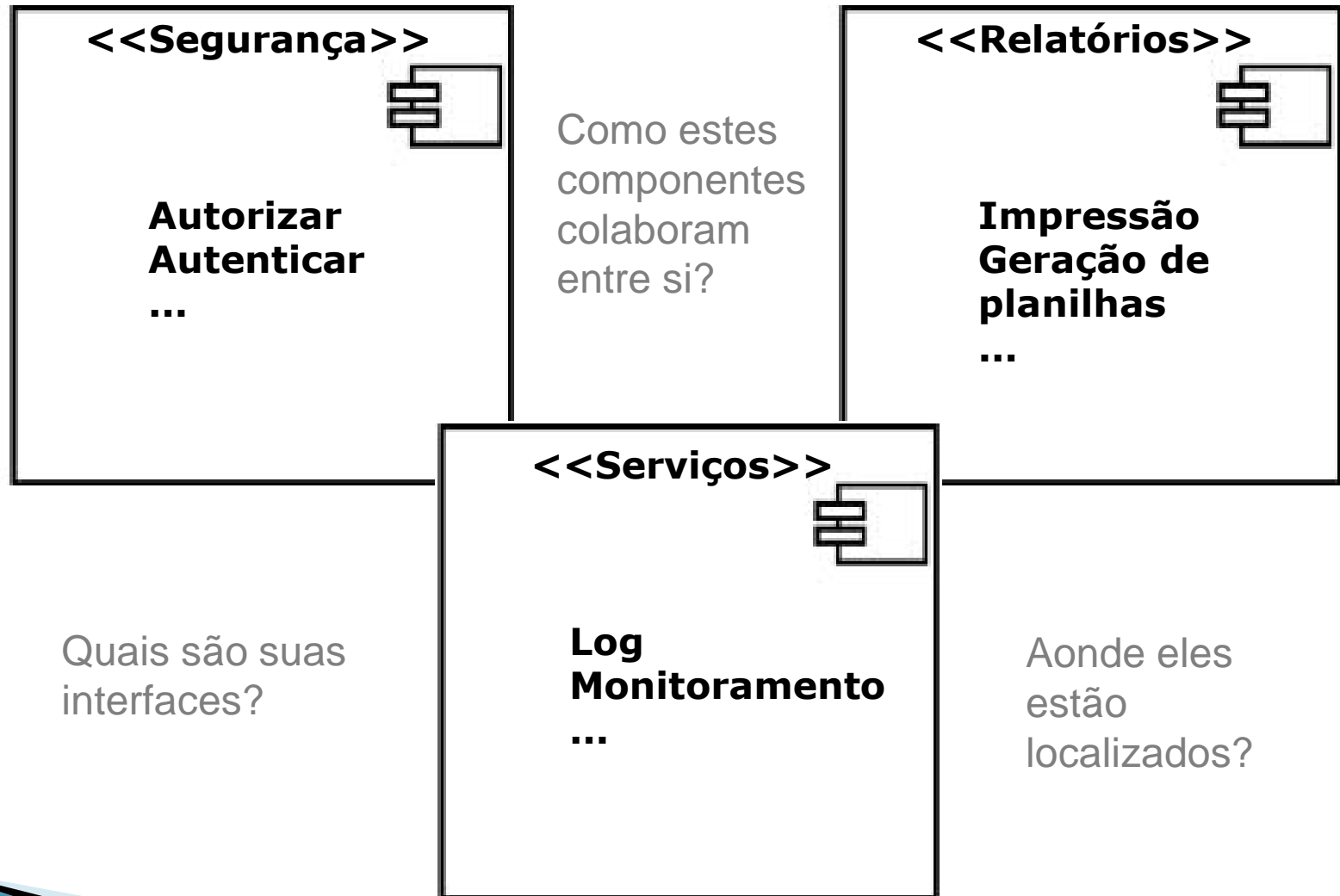
- ▶ Não se preocupa apenas com estrutura e comportamento, mas também com:
 - Funcionalidade
 - Desempenho
 - Segurança
 - Reuso
 - Manutenibilidade
 - Decisões tecnológicas e econômicas, ...

Componentes

▶ O que são?

- Grupos coesos de código fonte ou executável com interfaces e comportamentos bem definidos
- Fornecem forte encapsulamento de conteúdo
- Substituíveis
- Exemplos: módulos, pacotes, subsistemas, componentes OTS

Componentes



Vantagens de uma Arquitetura baseada em Componentes

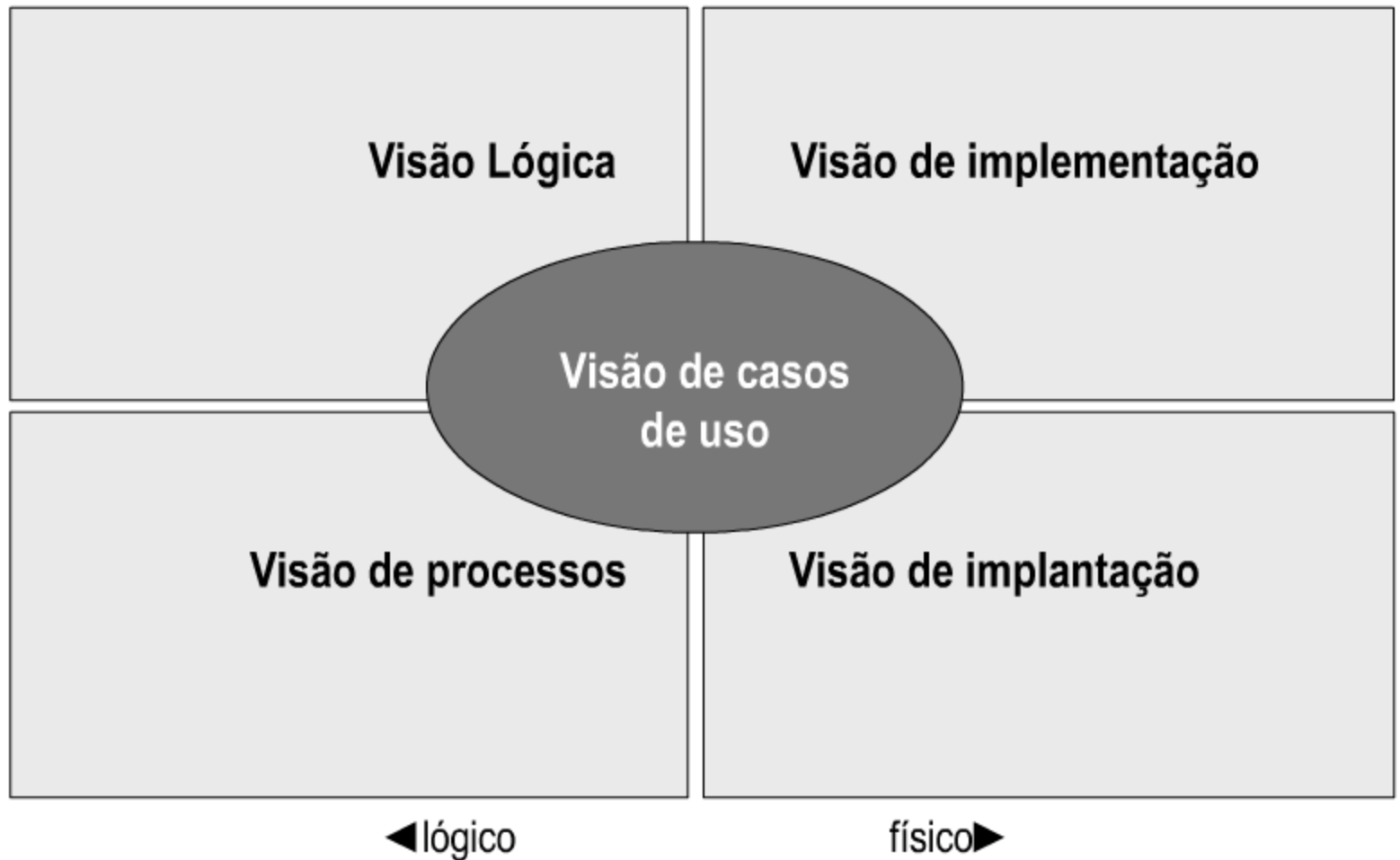
- ▶ Permite reuso em grande escala
- ▶ Permite gerenciar a complexidade do projeto e manter a integridade do sistema
- ▶ Identifica, isola, projeta, desenvolve e testa “pedaços” bem formados do sistema
- ▶ Possibilita usar componentes de prateleira (*off the shelf*)



Visões Arquiteturais

- ▶ No RUP a arquitetura é representada por uma série de visões de arquitetura diferentes
- ▶ Em sua essência, as Visões são fragmentos que ilustram os elementos “significativos em termos de arquitetura”
- ▶ É conhecido como o **modelo de visão 4+1**

4+1 Visões



Visão de Casos de Uso

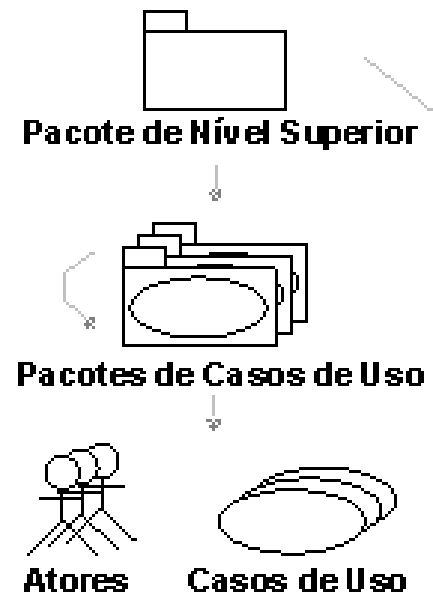
- ▶ Contém Casos de Uso e cenários que abrangem comportamentos significativos em termos de arquitetura, classes ou riscos técnicos
- ▶ É uma visão **obrigatória** do documento de arquitetura de software

Visão de Casos de Uso

Visão de Casos de Uso

mostra um subconjunto significativo
do ponto de vista da arquitetura de

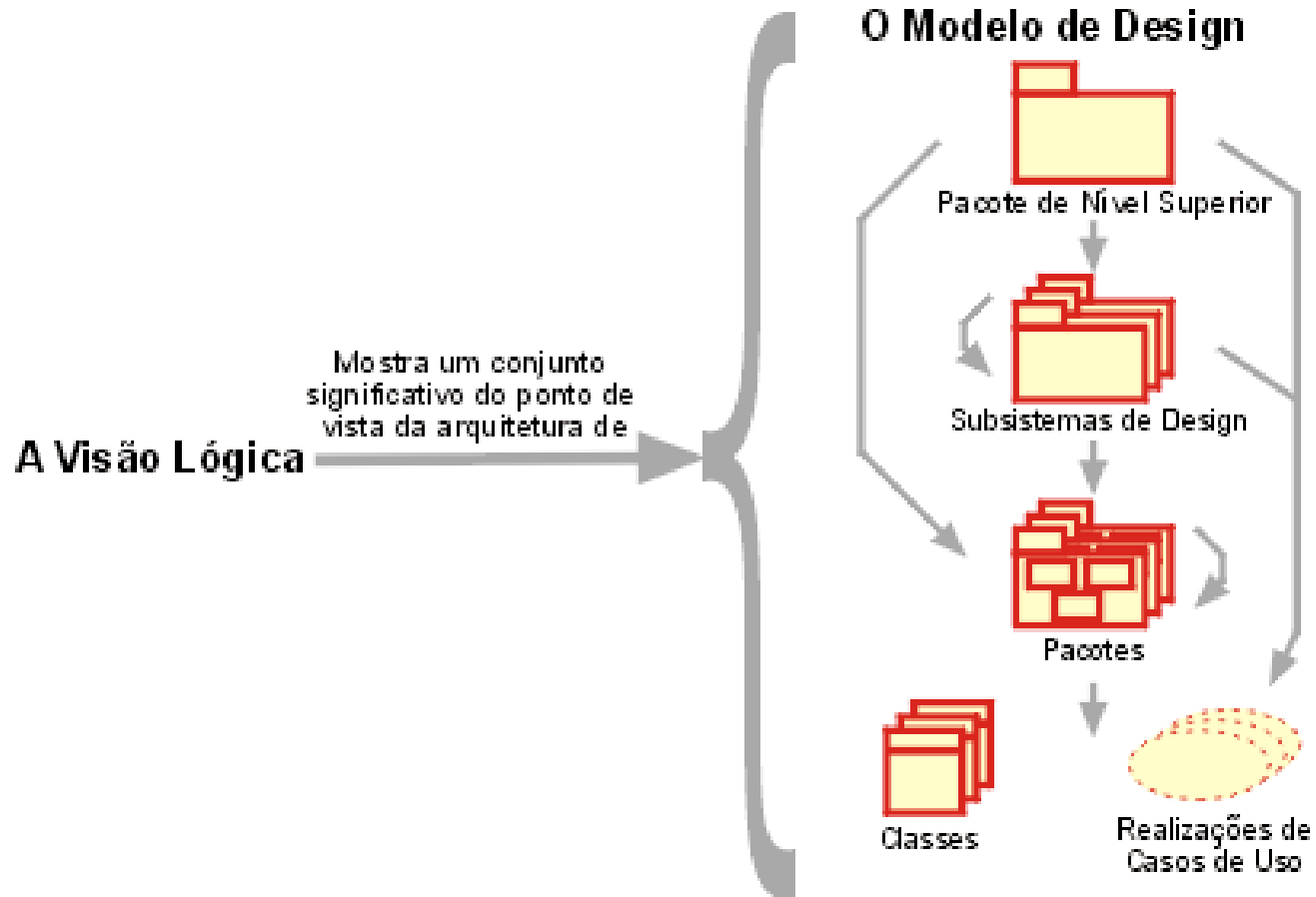
O Modelo de Casos de Uso



Visão Lógica

- ▶ Contém as classes de projeto mais importantes e sua organização em pacotes e subsistemas, e a organização desses pacotes em camadas
- ▶ É uma visão **obrigatória** do documento de arquitetura de software

Visão Lógica



Visão de Implementação

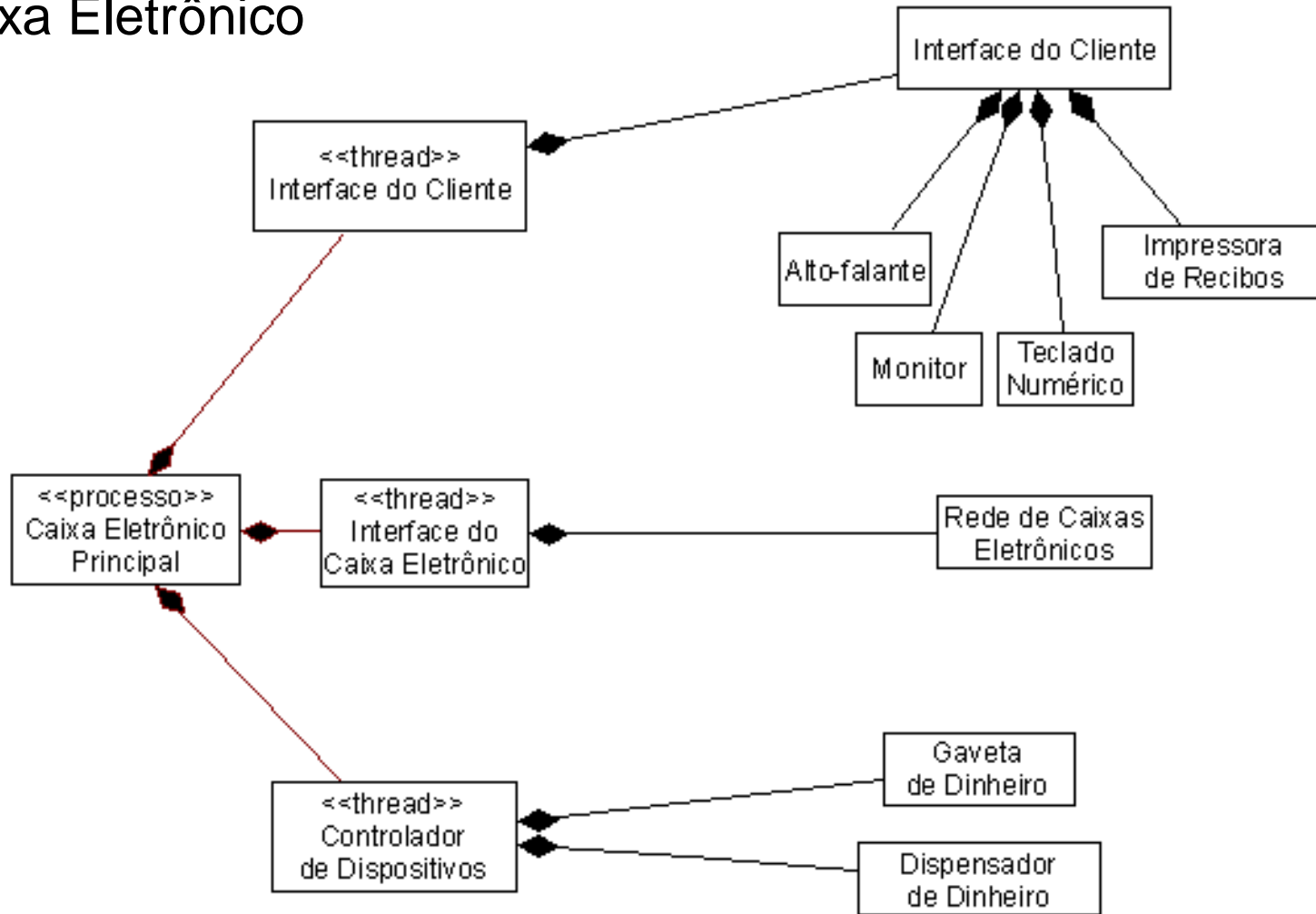
- ▶ Contém uma visão geral do Modelo de Implementação e sua organização em termos de módulos em pacotes e camadas
- ▶ Detalha os pacotes e módulos da Visão Lógica (detalhes “físicos”)
- ▶ É uma visão **opcional** do documento de arquitetura de software
 - Deve ser usada apenas se a implementação não for derivada diretamente do modelo de projeto (isto é, há detalhes adicionais)

Visão de Processos

- ▶ Contém a descrição das tarefas (processos e threads) envolvidas, suas interações e configurações e a alocação dos objetos e classes de projeto em tarefas
- ▶ É uma visão **opcional** do documento de arquitetura de software
 - Só precisa ser usada se o sistema tiver alto grau de paralelismo

Visão de Processos

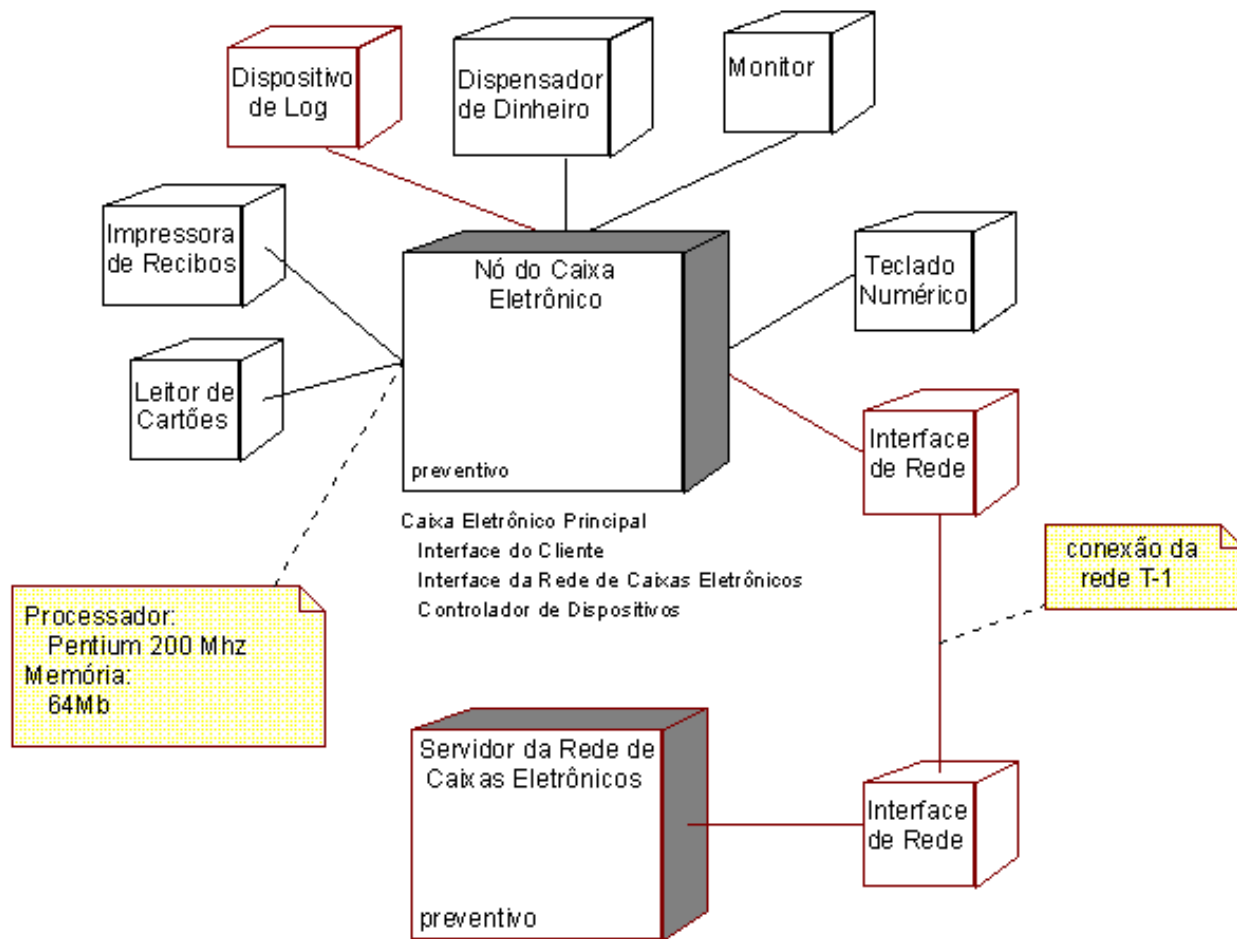
Caixa Eletrônico



Visão de Implantação

- ▶ Contém a descrição dos vários nós físicos do sistema e a alocação de tarefas atribuídas a eles
- ▶ É uma visão **opcional** do documento de arquitetura de software
 - Só precisará ser usada se o sistema estiver distribuído em vários nós físicos

Visão de Implantação



Exercícios [7]

(TRE/BA – CESPE 2010)

[62] Na engenharia de software baseada em componentes, na qual se supõe que partes do sistema já existam, o processo de desenvolvimento concentra-se mais na integração dessas partes que no seu desenvolvimento a partir do início. Essa abordagem é baseada em reuso para o desenvolvimento de sistemas de software.

(SAD/PE – CESPE 2010)

[35-D] É desejável que o valor da coesão e o do acoplamento, duas importantes propriedades da arquitetura de um software, sejam maximizados durante a engenharia de software.

Melhores Práticas: Modelagem Visual (UML)

O que é modelar visualmente?

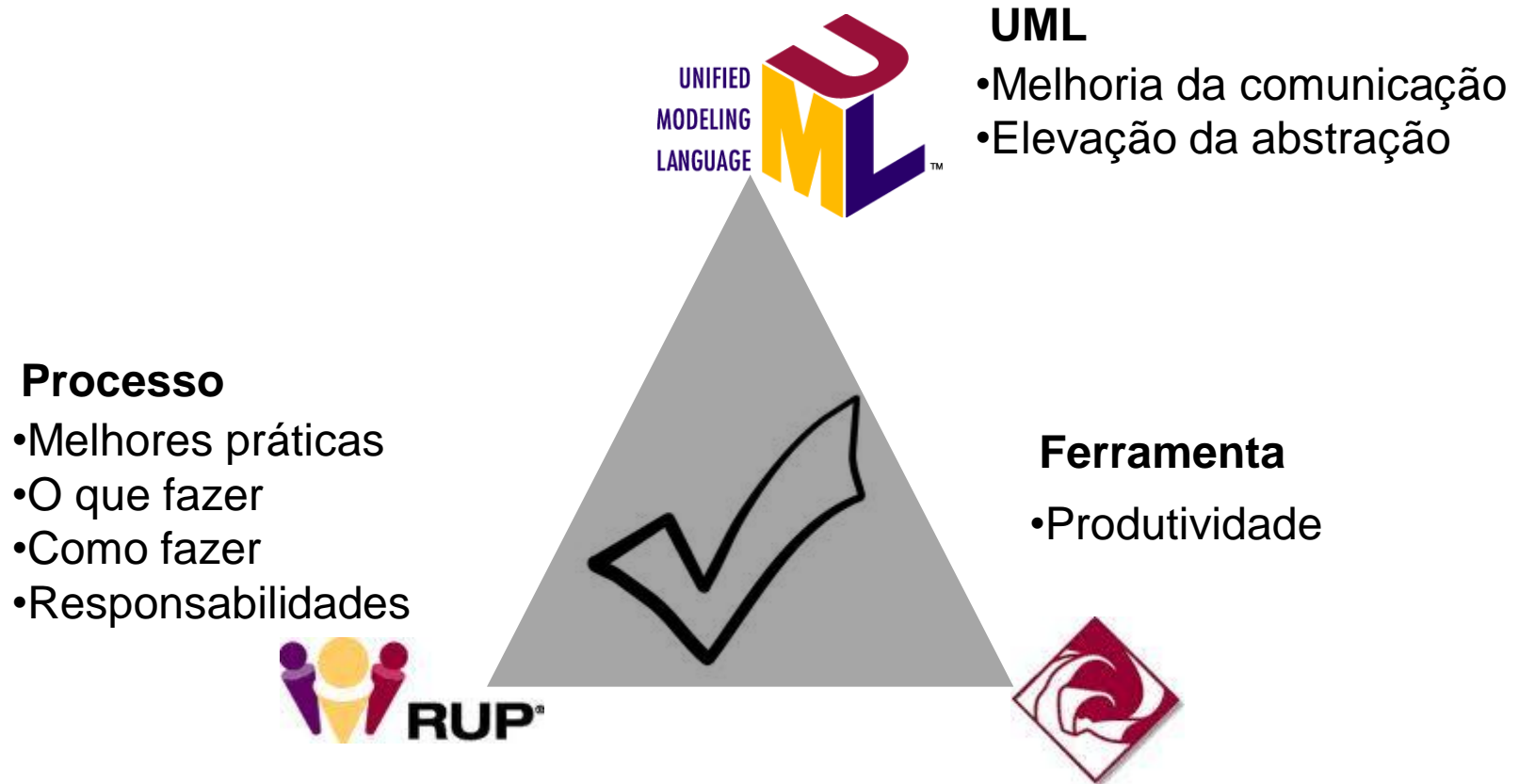
- ▶ Fazer uso de notação de design gráfica e visual para capturar o projeto do software
- ▶ Permitir que o nível de abstração seja aumentado
- ▶ Capturar requisitos com precisão
- ▶ Melhorar a comunicação da equipe (acabando com a ambiguidade)

RUP preconiza o uso de UML

- ▶ **UML – Unified Modeling Language**
 - Notação padrão para modelagem de Software
 - Oferece múltiplas perspectivas de um problema
 - Mantém projeto e implementação consistentes



UML x Processo x Ferramenta



Exercícios [8]

(TCU – CESPE 2010)

[108] UML (unified modeling language) é uma tecnologia concorrente com o processo unificado, no que diz respeito ao apoio à prática de engenharia de software orientada a objetos.

(TJ/CE – CESPE 2008)

[90] O modelo de ciclo de vida prescrito pela metodologia RUP é iterativo, incremental, direcionado por riscos, adota as áreas de processos de gerência de processos prescritas pelo modelo CMMI e é baseado em modelagem visual com UML e ferramentas CASE.

Melhores Práticas: Verificação da qualidade

Qualidade

- ▶ Para as finalidades do RUP, é definida como:

“...a característica de ter demonstrado a realização da criação de um produto que atende ou excede os requisitos acordados, conforme avaliado por medidas e critérios acordados, e que é criado em um processo acordado”

Controle x Garantia da Qualidade

▶ Controle da Qualidade

- Tem foco no **produto**, e em encontrar defeitos específicos
- Preza pelos **resultados** do seu trabalho

▶ Garantia da Qualidade

- Tem foco nos **processos** e como eles estão sendo executados
- Garante que você está fazendo as coisas de maneira correta

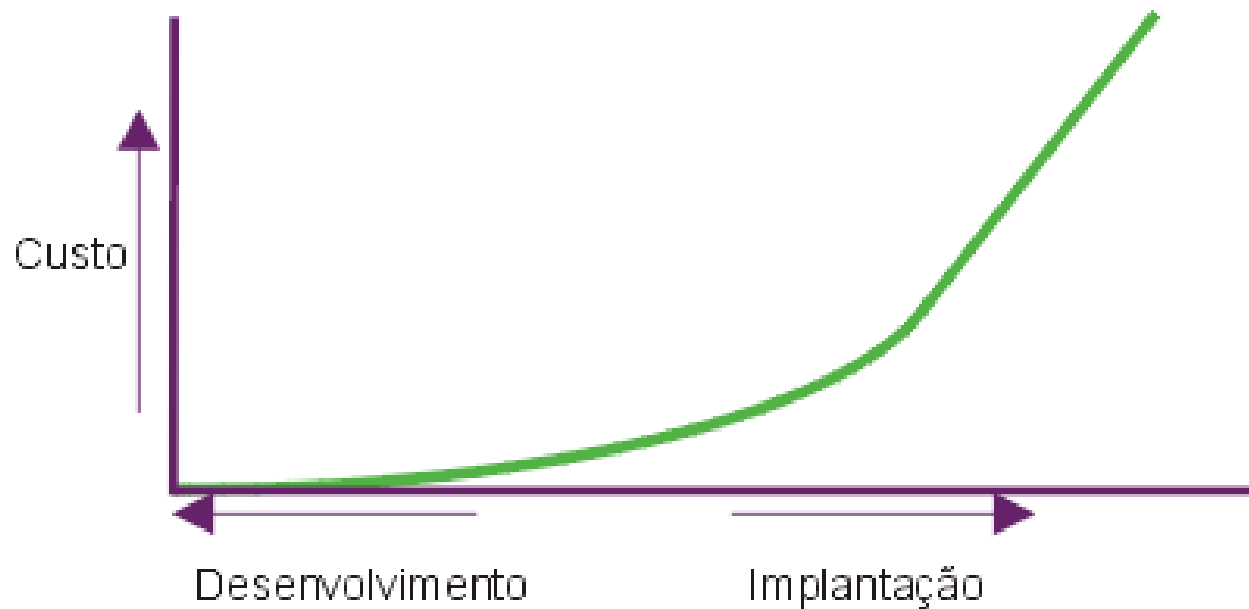
Dimensões da Qualidade

Qualidade é multidimensional

- ▶ Andamento: progresso do projeto
- ▶ Variação: diferença entre planejado e executado
- ▶ Confiabilidade: robustez
- ▶ Funcionalidade: casos de uso implementados
- ▶ Desempenho: tempo de execução em diversas situações reais

Mas afinal de contas: por que gerenciar qualidade?

- ▶ O custo sobe exponencialmente...
- ▶ Verificar a qualidade durante o ciclo de vida é **essencial!**



Exercícios [9]

(TRE/BA – CESPE 2010)

[53] Uma falha comum em projetos de sistemas computacionais é não assegurar a qualidade do software. Normalmente, essa questão é discutida após o término dos projetos, ou a qualidade fica sob a responsabilidade de equipe diferente da equipe de desenvolvimento. O RUP, proposto pela IBM, é um processo que provê uma solução disciplinada sobre como assinalar tarefas e responsabilidades dentro de uma organização de desenvolvimento de software, porém, não auxilia no controle do planejamento e verificação da qualidade.

Melhores Práticas: Gerenciamento de Mudanças

O ambiente do projeto...

- ▶ Vários desenvolvedores
- ▶ Diferentes equipes
- ▶ Diferentes locais
- ▶ Várias iterações, releases, produtos, plataformas

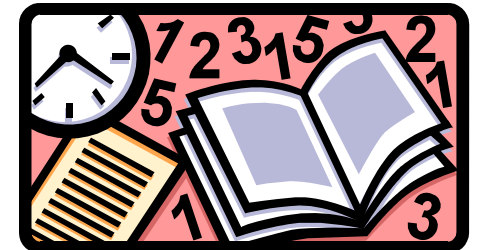
**Na ausência de controle disciplinado,
o processo de desenvolvimento
rapidamente se transforma no caos!**

Item de Configuração

- ▶ Uma entidade que satisfaz algum propósito para o usuário final e que pode ser unicamente identificada
- ▶ Podem ser
 - Arquivos–fonte, Executáveis, DLLs, etc.
 - Planos, especificações, modelos, etc.
 - Casos de teste, manuais, documentação de apoio, etc.
- ▶ Estão sob a Gestão da Configuração

Gerenciamento de Mudanças

- ▶ Gerência de Mudanças é o processo de avaliar, coordenar e decidir sobre a realização de mudanças propostas a itens de configuração (IC's)
- ▶ Apenas mudanças aprovadas são implementadas nos IC's e nos dados e documentos relacionados



Gerenciamento de Mudanças

No RUP, abrange as atividades de:

- ▶ Coordenação de atividades e artefatos
 - Procedimentos repetíveis para gerenciar mudanças sobre os artefatos
- ▶ Coordenação de iterações e releases
 - Mantém o controle sobre os releases ao final de cada iteração (baselines)
- ▶ Controle de mudanças no software
 - Mantém uma estrutura bem definida para gerenciar mudanças no software

Exercícios [10]

(TRE/MT – CESPE 2010)

[32-B] O RUP promove o uso de seis melhores práticas: desenvolva iterativamente; gerencie requisitos; use arquiteturas de componentes; modele visualmente (UML); verifique qualidade de software continuamente; e gerencie mudanças.

Exercícios [10]

(BNDES – CESGRANRIO 2009)

[51] A gerência de desenvolvimento de sistemas de uma empresa está reformulando seu processo de software. Para isso, deseja criar uma metodologia de desenvolvimento baseada no Processo Unificado. A respeito desse processo, é INCORRETO afirmar que o(a)

- (A) desenvolvimento é iterativo, incremental e orientado por casos de uso.
- (B) caso de uso mais crítico deve ser atacado, preferencialmente, no final.
- (C) fase de transição envolve treinamento de usuários e assistência no uso do produto.

Exercícios [10]

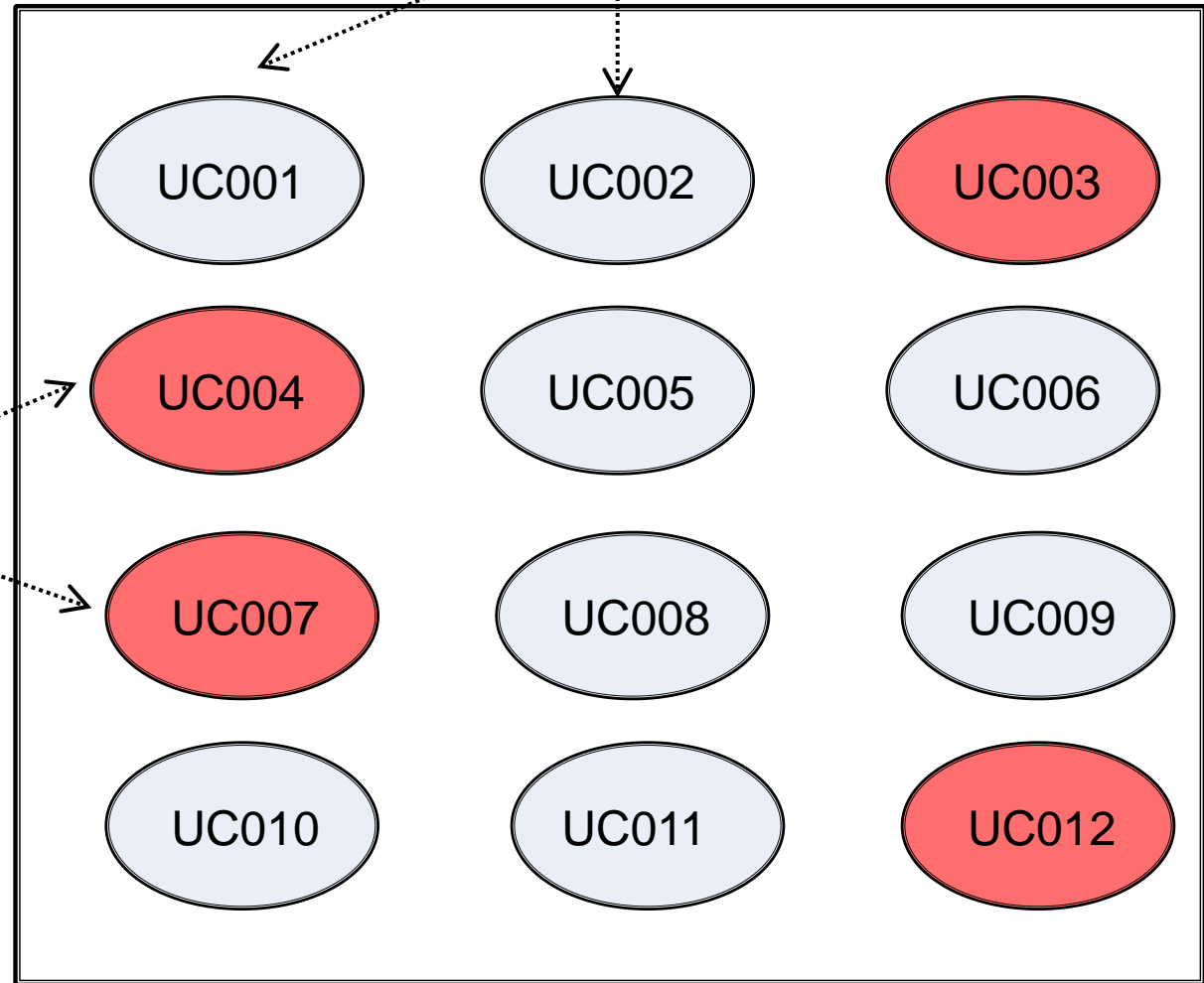
- (D) arquitetura se desenvolve a partir das visões do usuário expressas em casos de uso.
- (E) arquitetura, na fase de construção, é estável, ainda que possa ser evoluída.

Estudo de caso

O sistema...

Após levantados os principais requisitos (fase de Iniciação), por onde devo começar?

Casos de Uso



Casos de Uso
"arriscados"

Priorizando...

É feita uma matriz, que estabelece uma relação entre a importância dos requisitos e o quão “arriscados” eles são

	Risco	Importância
UC001	3	7
UC002	4	7
UC003	8	9
UC004	9	8
UC005	3	5
UC006	3	8
UC007	10	10
UC008	5	5
UC009	2	7
UC010	1	
UC011	1	7
UC012	8	10

Planejando segundo o RUP

- ▶ Segundo o RUP, as funcionalidades mais “arriscadas” e importantes devem ser implementadas primeiro
 - Elas devem ser “estabilizadas” já nas iterações da fase de Elaboração
- ▶ Ao final da fase de Iniciação, já existe o planejamento (de alto nível) do projeto feito
 - O detalhamento das funcionalidades é feito apenas para aquelas que serão feitas na próxima iteração
- ▶ Este ciclo de re-planejamento se dá ao longo das iterações, até o produto estar completo

Gabarito dos Exercícios

- ▶ [1] – [45] E, [27A] C, [109] C, [48] A, [80] C (E)
- ▶ [2] – [76] C, [70] C
- ▶ [3] – [72] E, [27 C] C, [80] E, [81] C, [86] E
- ▶ [4] – [79] C, [32–C] E, [32–D] E, [32–E] E
- ▶ [5] – [44–A] C, [44–C] E
- ▶ [6] – [72] E
- ▶ [7] – [62] C, [35–D] E
- ▶ [8] – [108] E, [90] E
- ▶ [9] – [53] E
- ▶ [10] – [32–B] C, [51] B

FIM



Rational Unified Process

Fases, Disciplinas e Atividades

Fernando Pedrosa – fpedrosa@gmail.com

Rational Unified Process (RUP)

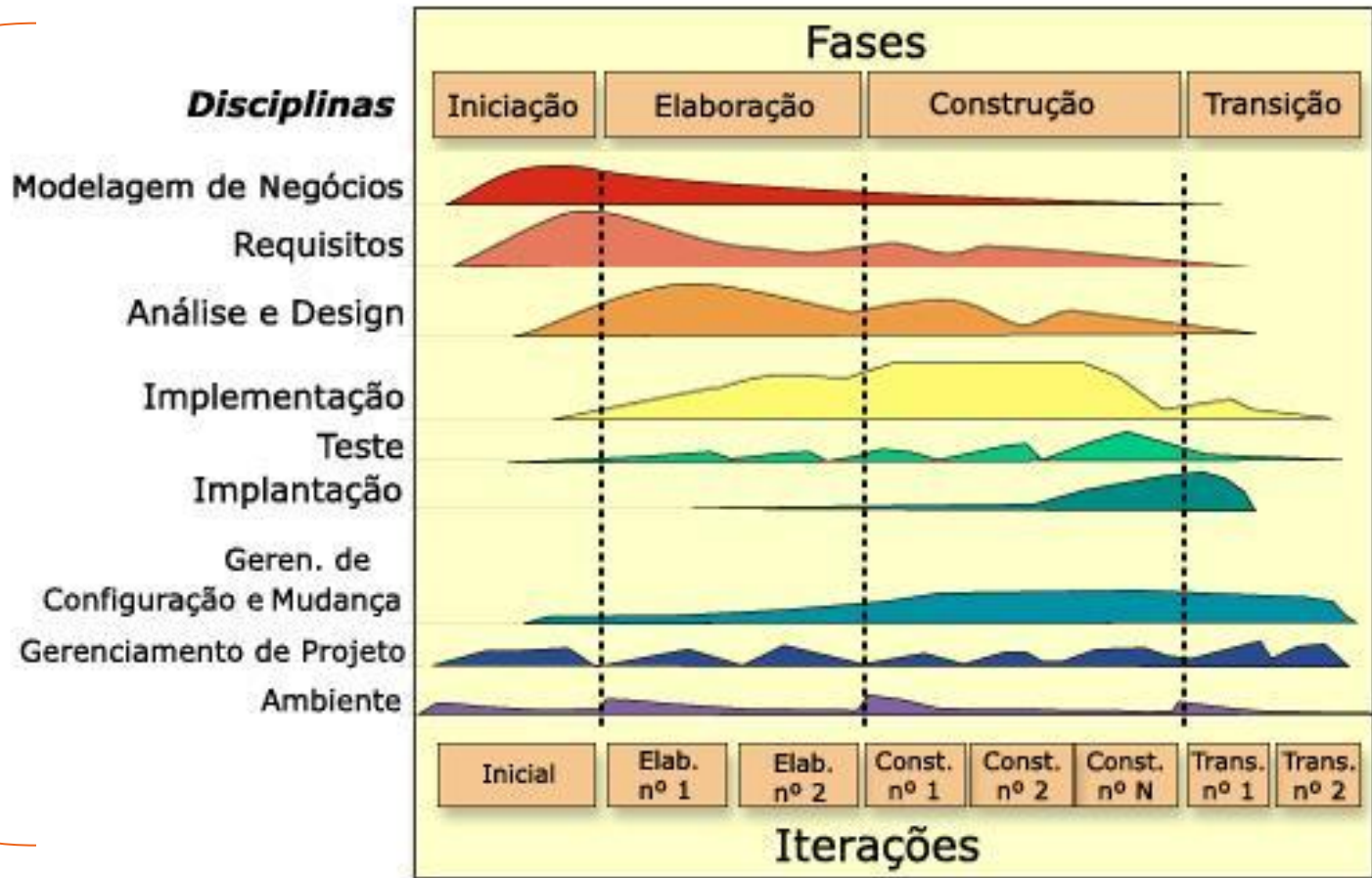
O RUP tem duas dimensões

- ▶ A primeira dimensão representa o aspecto dinâmico do processo
 - Eixo horizontal
 - Expresso em termos de fases, marcos e iterações
- ▶ A segunda dimensão representa o aspecto estático do processo
 - Eixo vertical
 - Expresso em termos de componentes, disciplinas, atividades, artefatos, papéis...

Gráfico das Baleias

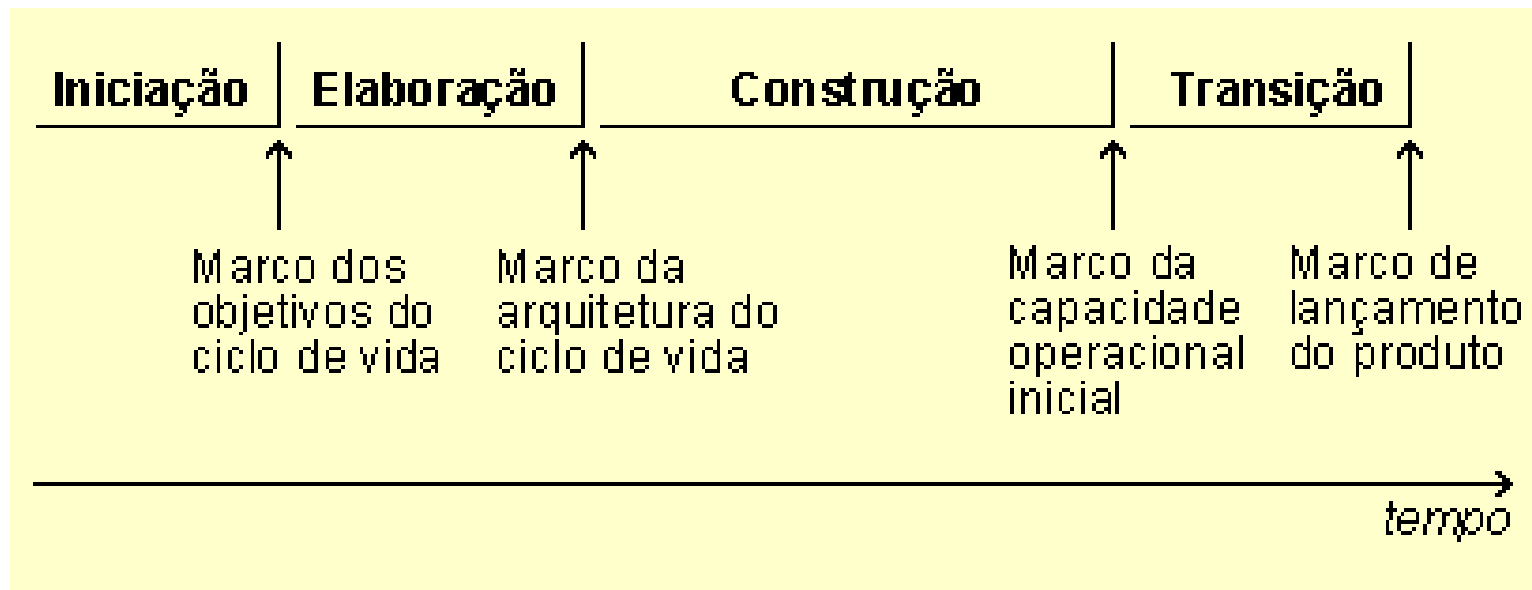
Eixo
dinâmico

Eixo
estático



Fases do RUP

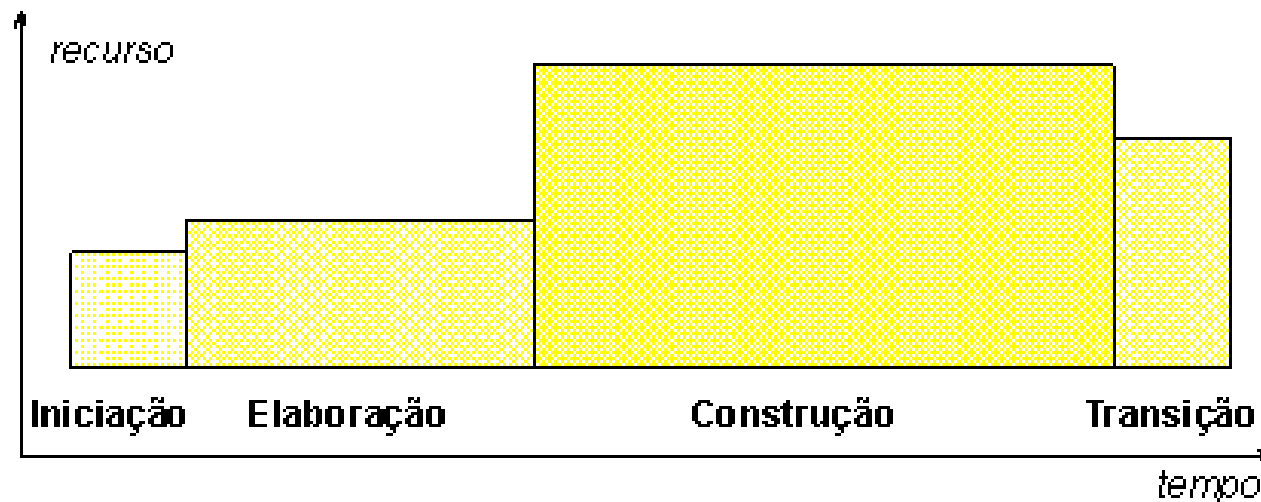
- ▶ São quatro fases sequenciais, cada uma concluída por um marco principal
- ▶ Cada fase é basicamente um intervalo de tempo entre dois marcos principais



Fases do RUP

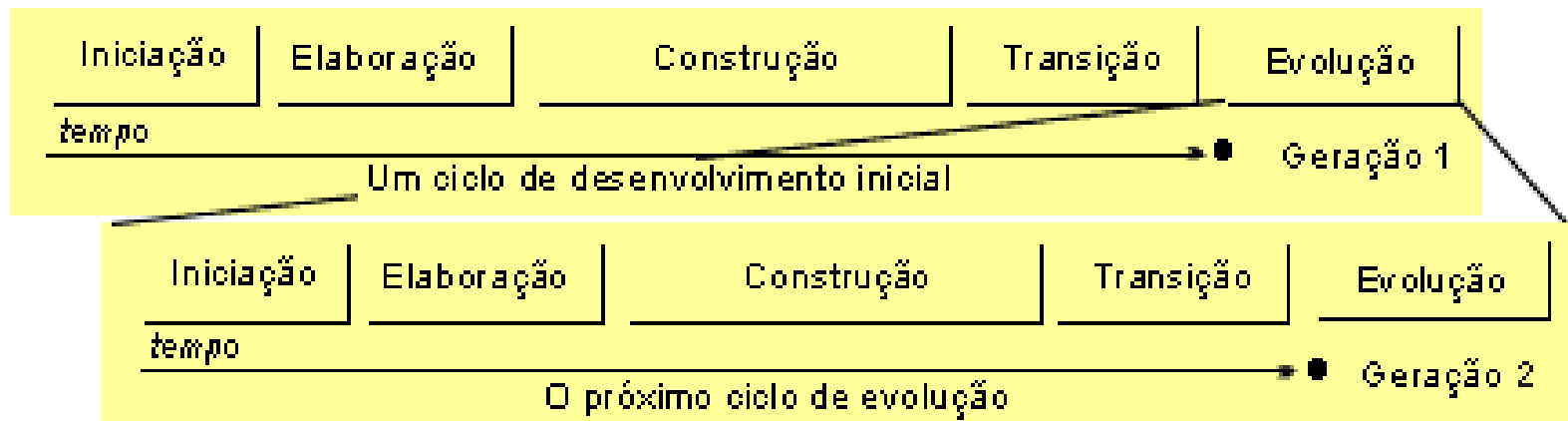
- ▶ As fases não são idênticas em termos de programação e esforço

	<u>Iniciação</u>	<u>Elaboração</u>	<u>Construção</u>	<u>Transição</u>
Esforço	~5 %	20 %	65 %	10%
Programação	10 %	30 %	50 %	10%



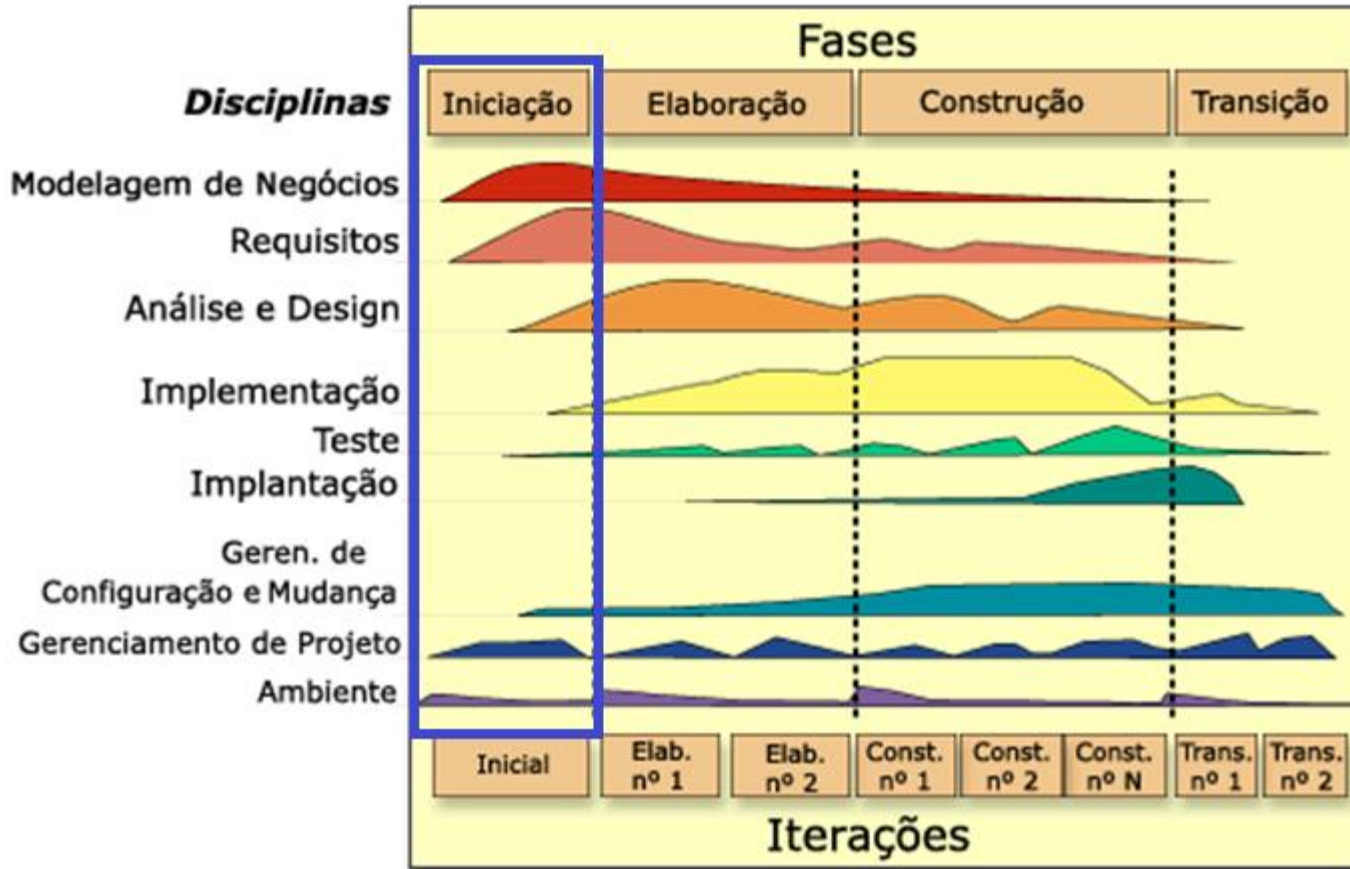
Fases do RUP

- ▶ Uma passagem pelas quatro fases é um **ciclo de desenvolvimento**
- ▶ As próximas “gerações” do produto têm ênfase em fases diferentes e são geradas por **ciclos de evolução**



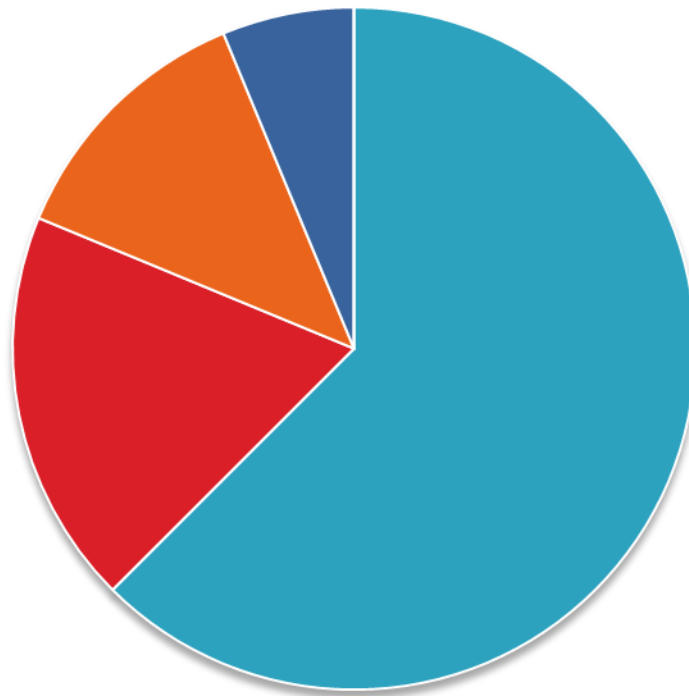
Fase de Iniciação (Concepção)

Fase de Iniciação



Fase de Iniciação

Esforço



- Requisitos
- Análise e Design
- Implementação
- Testes
- Implantação

Visão Geral

- ▶ Meta principal: atingir o consenso sobre os objetivos do ciclo de vida do projeto
 - Muito importante para projetos novos
 - Para projetos evolutivos, é uma fase mais rápida
- ▶ **Compensa fazer o projeto?**
- ▶ **É possível fazer o projeto?**

Objetivos

- ▶ Estabelecer o **escopo** do Software
- ▶ Estimar **custos**
- ▶ Estimar **tempo** (cronograma)
- ▶ Estimar **riscos**
- ▶ Identificar casos de uso críticos e principais cenários operacionais
- ▶ Propor pelo menos uma opção de arquitetura para alguns cenários básicos

Principais artefatos

- ▶ Visão
 - Necessidades e características mais importantes do sistema
- ▶ Caso de Negócio
 - Informações do ponto de vista do negócio
 - Determina se vale a pena investir no projeto (ROI)
- ▶ Plano de Desenvolvimento de Software
 - Reúne todas as informações necessárias ao gerenciamento do projeto

Principais artefatos

- ▶ Modelo de Casos de Uso
 - Contém as funções pretendidas do sistema
 - Serve como um contrato estabelecido entre os clientes e os desenvolvedores
- ▶ Glossário
 - Define termos importantes usados pelo projeto

<<Iniciação>>

Ênfase na Disciplina:
Modelagem de Negócios

Modelagem de Negócios:

Objetivos

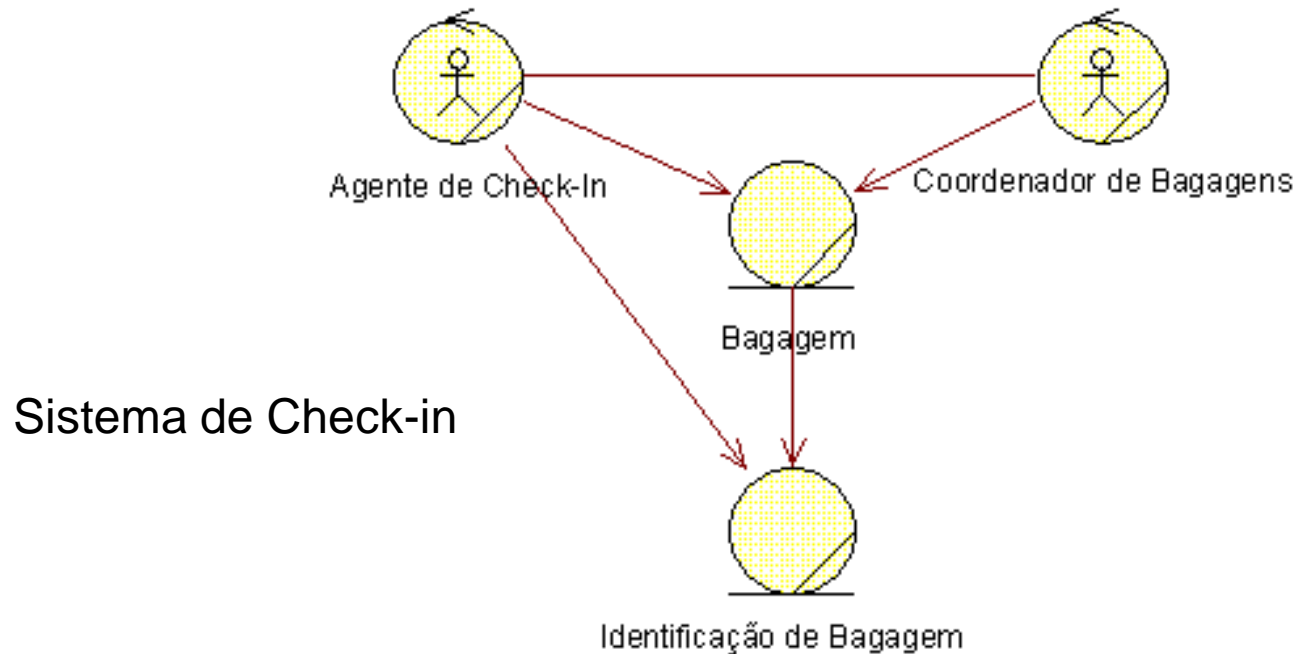
- ▶ Entender a estrutura e a dinâmica da organização-alvo, identificando oportunidades de melhoria
- ▶ Assegurar que todos os interessados tenham um entendimento comum sobre a organização
- ▶ Derivar os requisitos de sistema necessários para sustentar a organização-alvo

Modelagem de Negócios: Papeis, Atividades e Artefatos

- ▶ Principal Papel e suas Atividades
 - **Analista de Processo de Negócios**
 - Identificar os processos na organização
 - Descrever os processos
 - Definir o que pode e deve ser melhorado
 - Redesenhar os processos, se necessário
- ▶ Artefato importante para o marco
 - Modelo de Domínio (modelo de objetos de negócio)

Modelo de Domínio

Captura os tipos mais importantes de objetos no contexto de domínio



Relação com outras disciplinas

▶ Requisitos

- Utiliza modelos de negócio como subsídio para entender os requisitos do sistema

▶ Análise e Design

- Utiliza entidades de negócio para identificar classes de entidade no projeto

▶ Ambiente

- Desenvolve e mantém artefatos de suporte, como o Guia de Modelagem de Negócios

<<Iniciação>>

Ênfase na Disciplina:
Requisitos

Requisitos: Objetivos

- ▶ Estabelecer o que o sistema deve fazer
- ▶ Definir as fronteiras (escopo) do sistema
- ▶ Fornecer uma base para planejar o conteúdo técnico das iterações
- ▶ Fornecer uma base para estimar o custo e o tempo de desenvolvimento do sistema

Requisitos: Papeis, Atividades e Artefatos

▶ Principais Papeis e Atividades

◦ Analista de Sistemas

- Levantar Requisitos do Sistema (Atores e CDU's)
- Estruturar Modelo de Casos de Uso

◦ Especificador de Requisitos

- Detalhar Especificação de Casos de Uso

▶ Artefatos importantes para o marco

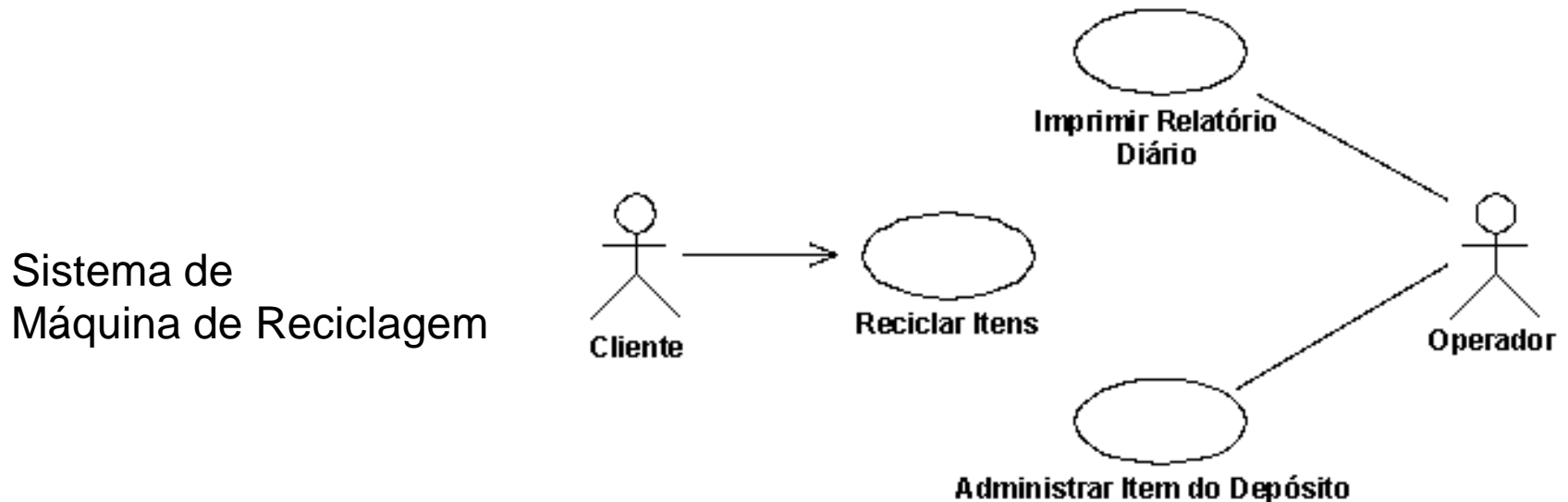
◦ Visão

◦ Glossário

◦ Modelo de Casos de Uso

Modelo de Casos de Uso

Modelo das funções pretendidas do sistema. Serve como contrato entre o cliente e desenvolvedores.



Documento de Visão



- ▶ É o documento que define a visão que os envolvidos têm do produto a ser desenvolvido
- ▶ Contém as necessidades e características mais importantes do sistema
- ▶ Fornece uma base de alto nível para que o leitor possa compreender o sistema a ser desenvolvido

Relação com outras disciplinas

- ▶ Modelagem de Negócios
 - Fornece as regras de negócio e um contexto organizacional para o sistema
- ▶ Análise e Design
 - Obtém suas informações primárias dos Requisitos. Pode encontrar falhas nos modelos de Caso de Uso
- ▶ Teste
 - Valida o sistema com base nos casos de uso

Relação com outras disciplinas

- ▶ Gerenciamento de Configuração e Mudança
 - Fornece o mecanismo de controle para as mudanças nos requisitos
- ▶ Gerenciamento de Projeto
 - Usa o modelo de casos de uso para planejar as iterações
- ▶ Ambiente
 - Desenvolve e mantém os artefatos utilizados na disciplina de Requisitos

Marco da Iniciação: Objetivos do Ciclo de Vida

- ▶ É o primeiro marco mais importante do projeto
- ▶ Critérios de avaliação
 - Os casos de uso definem claramente o escopo?
 - Caso necessário, foi possível fazer um protótipo da arquitetura?
 - Todos os riscos críticos foram encontrados? Se sim, foram mitigados?
 - Há condições de se fazer o projeto?

Exercícios [1]

(BASA – CESPE 2007)

[53] Na fase de concepção (inception), há atividades voltadas para a definição do escopo do sistema, identificação de atores e casos de uso, definição de vocabulário que possa ser usado nas descrições textuais do sistema, e definição de uma arquitetura candidata para o sistema que está sendo desenvolvido.

(Min. Comunicações – CESPE 2008)

[72] São objetivos da fase de concepção (inception): preparar ambiente para o projeto; elaborar plano para o projeto; definir escopo do sistema; identificar atores e casos de uso; identificar as necessidades dos stakeholders; definir níveis de prioridade dos casos de uso; propor arquitetura candidata; e definir objetivos do esforço de teste.

Exercícios [1]

(CGU – ESAF 2008)

[22] No Processo Unificado (PU), o termo Modelo de Domínio significa uma representação visual de classes conceituais ou objetos do mundo real. Assinale a opção que apresenta uma afirmativa correta quanto ao Modelo de Domínio.

- A) Não trata da representação de objetos de software.
- B) Significa um conjunto de diagramas que descreve classes de software.
- C) Representa a camada de domínio de uma arquitetura de software.
- D) Representa objetos de software com responsabilidades.
- E) Aplicando a notação UML, é ilustrado como um conjunto de diagramas de classe em que são definidas as operações.

Exercícios [1]

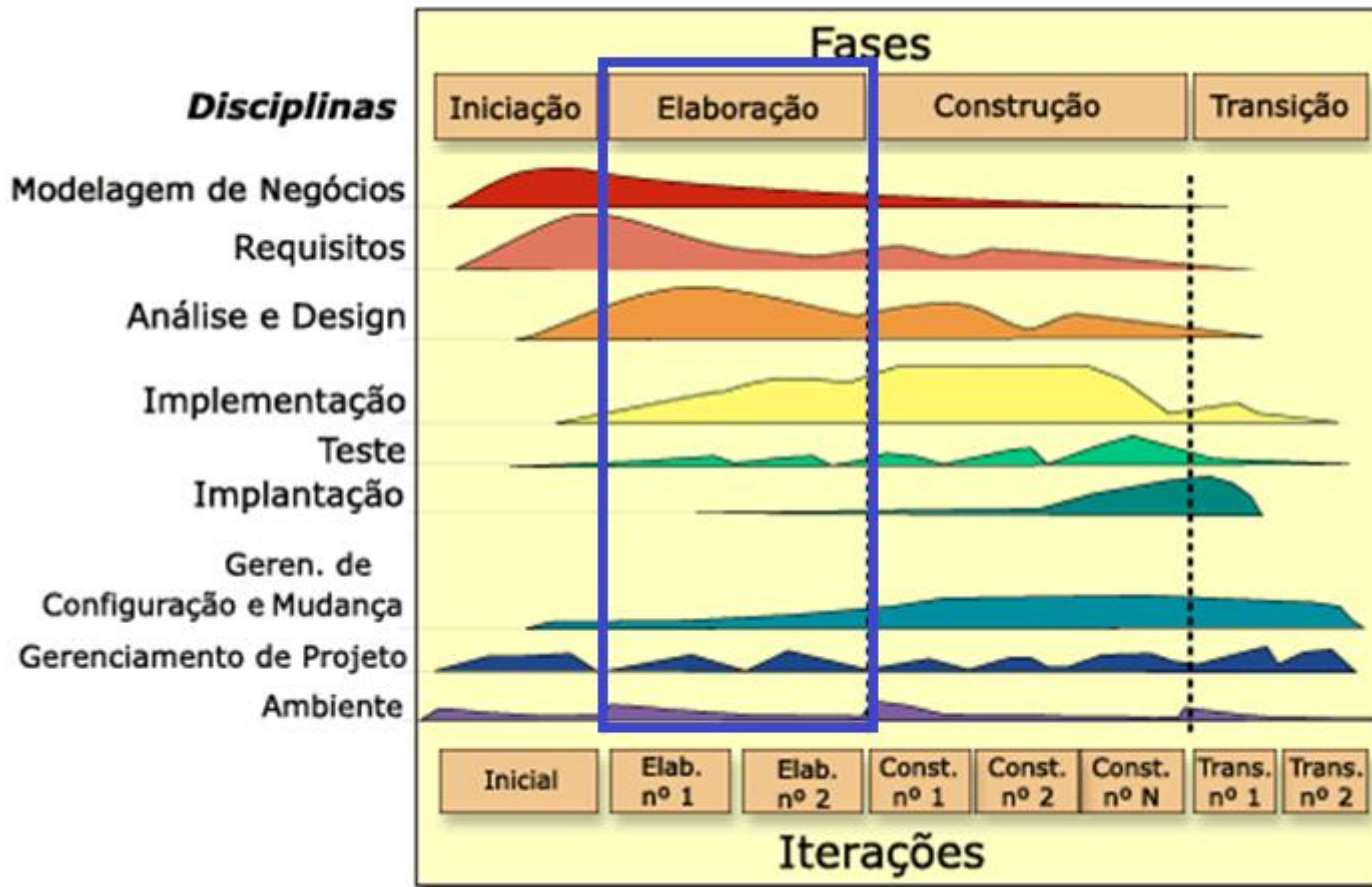
(Sergipe Gás – FCC 2010)

[31] No Processo Unificado, o Modelo de Domínio é um

- a) diagrama de classes em nível de análise.
- b) diagrama de classes em nível de desenho.
- c) produto da modelagem de negócios e, como tal, captura o vocabulário do sistema ou negócio sob modelagem.
- d) modelo que carrega todo o detalhamento do comportamento e estrutura, que devem estar presentes em um modelo de análise.
- e) modelo de domínio que carrega informações de armazenamento de informações ou normalizações, que devem estar presentes em um DER

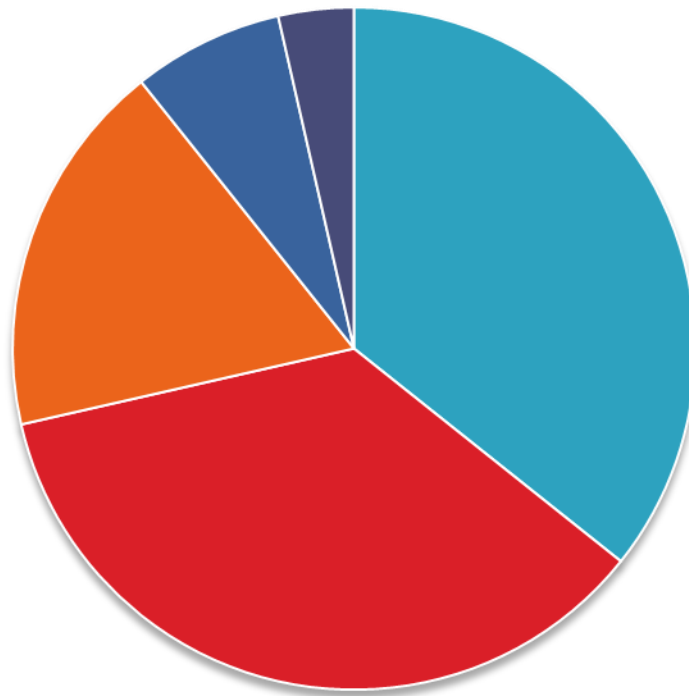
Fase de Elaboração

Fase de Elaboração



Fase de Elaboração

Esforço



- Requisitos
- Análise e Design
- Implementação
- Testes
- Implantação

Visão Geral

- ▶ A meta principal da fase de Elaboração é fornecer uma base estável para o esforço de Construção
- ▶ A arquitetura é desenvolvida a partir dos requisitos que têm maior impacto na arquitetura
- ▶ A estabilidade da arquitetura é avaliada através de um ou mais protótipos de arquitetura

Objetivos

- ▶ Assegurar que a arquitetura e os requisitos estejam estáveis para mitigar riscos
 - “Ultrapassar esta marca significa passar de uma operação rápida e de baixo risco para uma operação de alto custo e alto risco”
- ▶ Tratar todos os riscos significativos do ponto de vista da arquitetura do projeto
- ▶ Selecionar componentes e criar planos de iterações detalhados para a fase de Construção

Principais artefatos

▶ Protótipos

- São usados de uma maneira direta para reduzir o risco e elicitar requisitos significativos.

▶ Documento de Arquitetura de Software

- Fornece a visão geral de arquitetura abrangente do sistema, usando diversas visões de arquitetura para descrever diferentes aspectos do sistema.

Principais artefatos

▶ Modelo de Projeto

- É um modelo de objeto que descreve a realização dos casos de uso e serve como uma abstração do modelo de implementação e seu código-fonte.

▶ Modelo de Dados

- É um subconjunto do modelo de implementação que descreve a representação lógica e física dos dados persistentes no sistema.

<<Elaboração>>

Ênfase na Disciplina:
Análise e Design

Análise e Design: Objetivos

- ▶ Transformar os requisitos em um projeto do sistema a ser criado
- ▶ Desenvolver uma arquitetura refinada para o sistema
- ▶ Adaptar o projeto para que corresponda ao ambiente de implementação, considerando restrições de tecnologia

Análise e Design: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - **Arquiteto de Software**
 - Projetar arquitetura
 - **Designer (Projetista)**
 - Analisar casos de uso
 - Projetar casos de uso
 - Projetar subsistemas
 - **Projetista de Banco de Dados**
 - Projetar base de dados

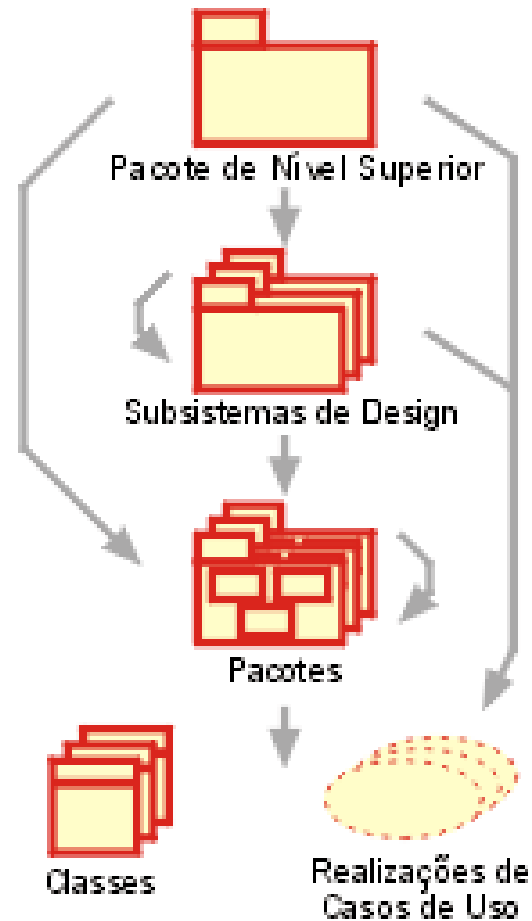
Análise e Design: Papeis, Atividades e Artefatos

- ▶ Artefatos importantes para o marco
 - **Protótipos**
 - Quanto ao que exploram:
 - Comportamentais
 - Estruturais
 - Quanto ao seu resultado:
 - Exploratórios (ou de descarte)
 - Evolutivos
 - **Documento de Arquitetura de Software**
 - **Modelo de Design**

Modelo de Design

Modelo que descreve as realizações dos casos de uso e serve como uma abstração do modelo de implementação

O Modelo de Design



Relação com outras disciplinas

- ▶ Modelagem de negócio
 - Fornece o contexto organizacional para o sistema
- ▶ Requisitos
 - Fornece a visão das funcionalidades críticas a serem implementadas
- ▶ Teste
 - Testa o sistema projetado durante a disciplina de Análise e Design
- ▶ Ambiente, Gerenciamento de Projeto

Marco da Elaboração: Arquitetura do Ciclo de Vida

- ▶ É o segundo marco mais importante do projeto. Deve-se analisar a arquitetura executável e a resolução dos principais riscos
- ▶ Critérios de Avaliação
 - A arquitetura é estável e robusta, comportando requisitos atuais e futuros?
 - Riscos críticos foram resolvidos?
 - O planejamento está bem definido em termos de cronograma, orçamento e níveis de qualidade?
 - Devemos fechar o contrato?

Exercícios [2]

(SERPRO – CESPE 2010)

[73] No modelo RUP, a primeira linha de base da arquitetura de um software é produzida ao final da fase de elaboração.

(ISJN – CESPE 2010)

[57] Modelo de domínio, descrição da arquitetura de software e versão preliminar do manual são resultados-alvo da fase elaboração do RUP.

[54] Na fase de elaboração, muitos componentes do sistema são implementados, testados e integrados. Essas atividades, que partem de uma arquitetura definida, validada e implementada em fases anteriores do ciclo de desenvolvimento, produzem um sistema operacional pronto para ser instalado em um ambiente em que serão feitos testes beta.

Exercícios [2]

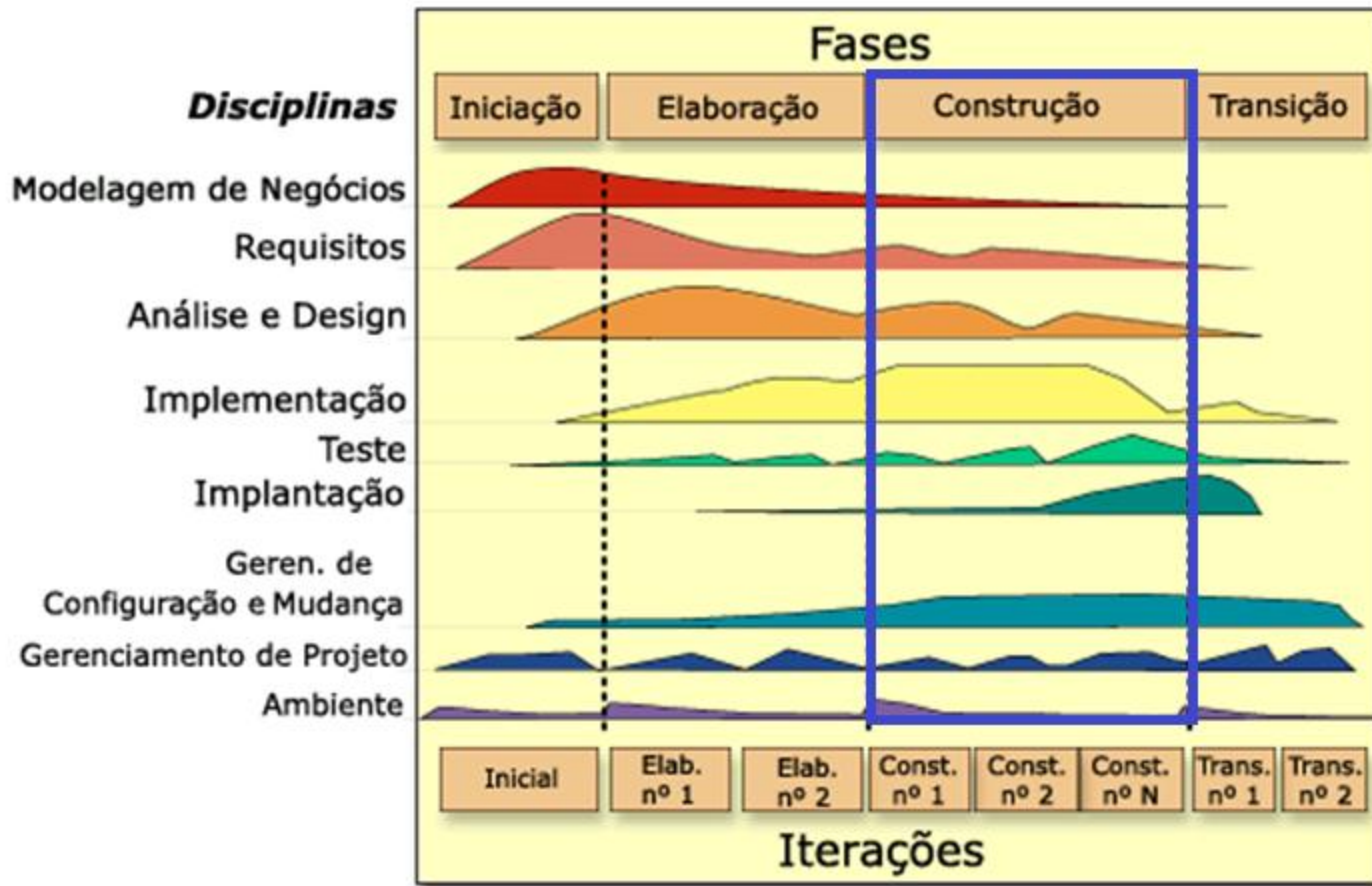
(PETROBRAS – CESGRANRIO 2010)

[39] A análise de risco no RUP é algo constante nas diversas fases do processo de desenvolvimento. Em cada uma das fases, o foco da gerência de riscos se diferencia em função do objetivo de cada fase. Assim, a manipulação dos riscos está relacionada, na fase de

- (A) análise, ao refinamento do modelo de requisitos e à sua possível alteração.
- (B) construção, à instalação e distribuição do produto no ambiente do cliente.
- (C) transição, à logística, uma vez que é a fase que envolve o maior número de profissionais.
- (D) requisitos, à modelagem de negócio.
- (E) elaboração, a questões técnicas, envolvendo a arquitetura escolhida.

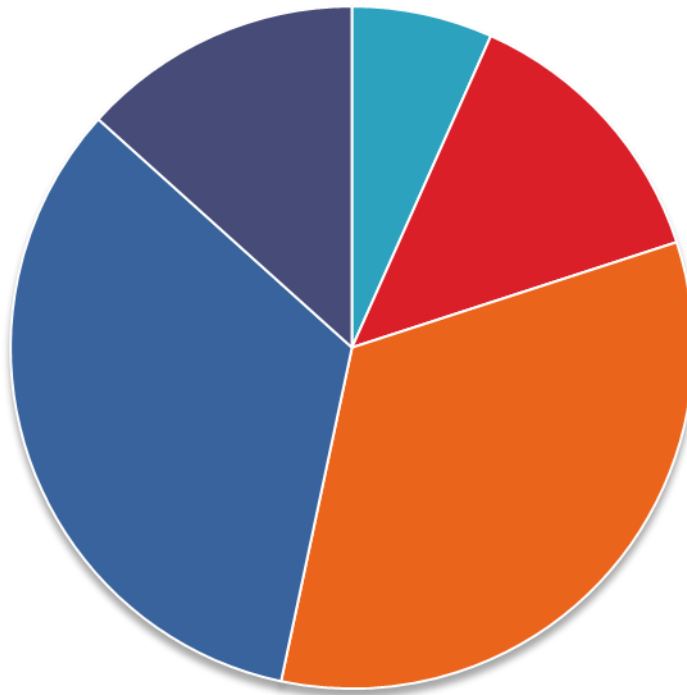
Fase de Construção

Fase de Construção



Fase de Construção

Esforço



- Requisitos
- Análise e Design
- Implementação
- Testes
- Implantação

Visão Geral

- ▶ Esclarecer os requisitos restantes
- ▶ Concluir o desenvolvimento do sistema com base na arquitetura estável
- ▶ É, de certa forma, um processo de manufatura
- ▶ A ênfase está no gerenciamento de recursos e controle de operações para alcançar maior produtividade e qualidade

Objetivos

- ▶ Minimizar custos de desenvolvimento, otimizar recursos e evitar retrabalho
- ▶ Atingir as versões úteis (alfa, beta e outros releases de teste) com rapidez
- ▶ Concluir a análise, o projeto, o desenvolvimento e o teste de todas as funcionalidades necessárias.
- ▶ Decidir se o software está pronto para ser implantado

Principais artefatos

▶ “O Sistema”

- O próprio sistema executável, pronto para iniciar os testes beta

▶ Plano de Implantação

- Versão inicial desenvolvida, analisada e com baseline
- Em projetos menores pode estar embutido no Plano de Desenvolvimento do Software

▶ Conjunto de testes

- Testes implementados e executados para validar a estabilidade dos releases

<<Construção>>

Ênfase na Disciplina:
Implementação

Implementação: Objetivos

- ▶ Definir o código em subsistemas e camadas
- ▶ Implementar classes e objetos em termos de componentes
- ▶ Testar os componentes desenvolvidos como unidades
- ▶ Integrar os resultados produzidos por implementadores individuais (ou equipes) ao sistema executável

Implementação: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - Implementador
 - Implementar componente
 - Realizar testes unitários
 - Integrador
 - Integrar o sistema
- ▶ Artefatos importantes para o marco
 - “O Sistema”
 - Componentes
 - Builds

Relação com outras disciplinas

▶ Análise e Design

- Representa o propósito da implementação, sendo o Modelo de Design a principal entrada desta disciplina

▶ Teste

- Descreve como realizar o teste de integração de cada build
- Descreve também como testar o sistema

▶ Implantação

- Descreve como usar o modelo de implementação para produzir e liberar o código para o cliente final

▶ Ambiente, Gerenciamento de Projeto

<<Construção>>

Ênfase na Disciplina:
Teste

Objetivos

- ▶ Localizar e documentar defeitos na qualidade do software
- ▶ Validar as suposições feitas nas especificações de design e requisito através de demonstração concreta
- ▶ Validar as funções do software conforme projetadas
- ▶ Verificar se os requisitos foram implementados de forma adequada

Testes: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - **Analista de Teste**
 - Elaborar plano de testes
 - **Projetista de Teste**
 - Projetar testes
 - **Testador**
 - Implementar teste
 - Executar Testes
- ▶ Artefatos importantes para o marco
 - Conjunto de testes

Relação com outras disciplinas

- ▶ Requisitos
 - Os casos de uso fornecem a base para o que vai ser testado na disciplina de Teste
- ▶ Análise e Design
 - Fornece o projeto a ser testado pela disciplina de Teste
- ▶ Implementação
 - Produz os builds do software que serão validados pelos testes
- ▶ Ambiente, Gerenciamento de Projeto, Gerenciamento de Configuração e Mudanças

Marco da Construção: Capacidade Operacional Inicial

- ▶ Neste marco, o produto está pronto para ser passado para a Equipe de Transição.
- ▶ Toda a funcionalidade já foi desenvolvida e todos os testes **alfa** foram concluídos.
- ▶ O manual do usuário é produzido e uma existe uma descrição do release atual
- ▶ Critérios de avaliação
 - O produto está estável para ser implantado?
 - O resultado está coerente com o planejado?
 - Os envolvidos estão prontos para a Transição?

Exercícios [3]

(PETROBRAS – CESPE 2007)

[98] Na fase de construção, são implementados os casos de uso que tenham impacto sobre a arquitetura; na fase de transição, os casos sem impacto sobre a arquitetura, mas que descrevam funcionalidades que deverão estar presentes na versão que está sendo desenvolvida.

(PGE/PA – CESPE 2007)

[34] Na disciplina de teste, entre os artefatos que podem ser produzidos, tem-se o plano de teste. O plano de teste pode definir os objetivos dos testes no escopo de uma iteração ou do projeto, os itens a serem testados e as abordagens dos testes.

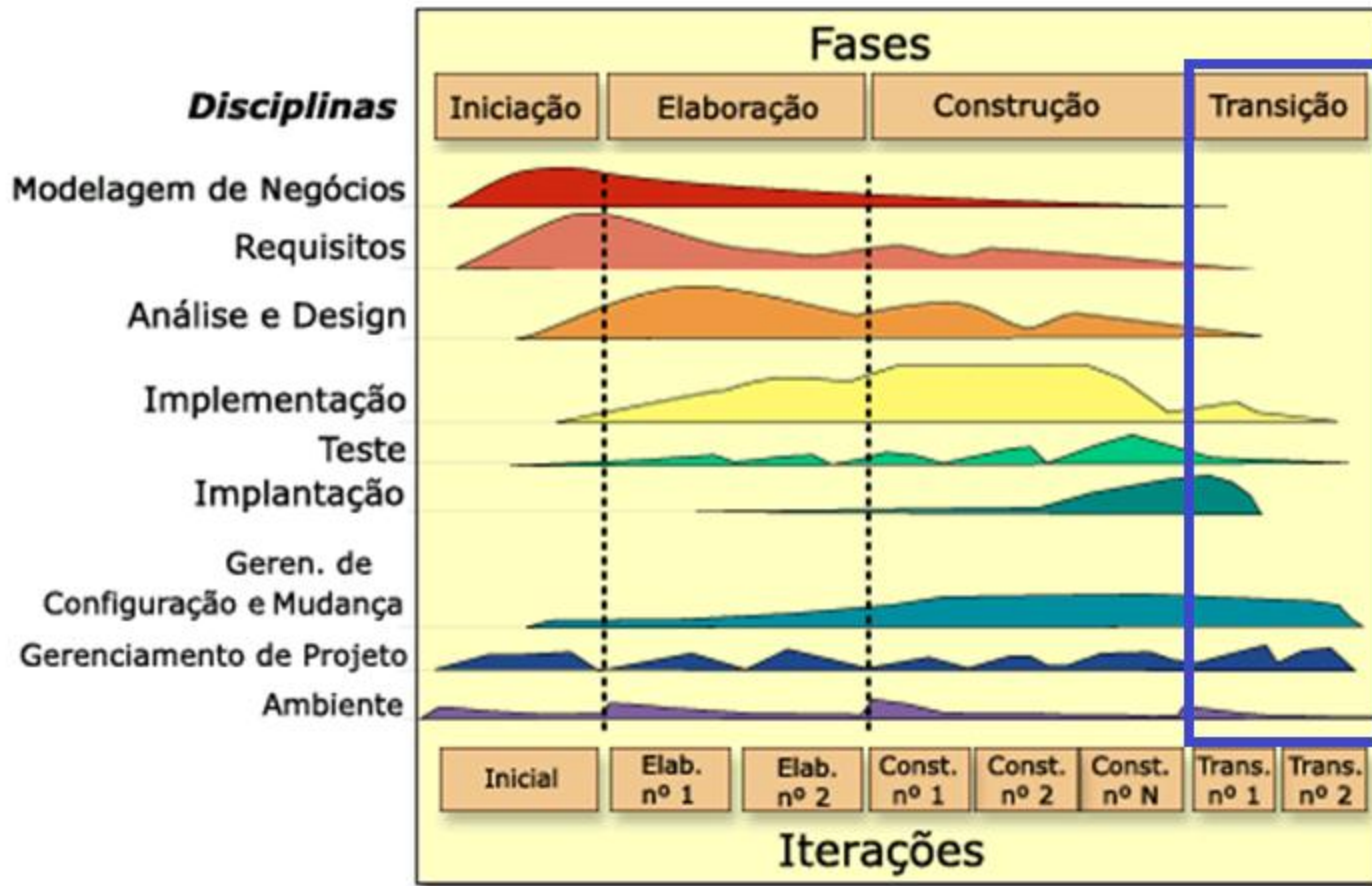
Exercícios [3]

(Min. Comunicações – CESPE 2008)

[73] A fase de construção (construction) tem os seguintes objetivos: detalhar casos de uso e requisitos do software; refinar a arquitetura proposta e demonstrar que essa arquitetura suporta os requisitos do sistema; testar e avaliar protótipos visando demonstrar que os principais riscos foram avaliados; e construir protótipos executáveis para a avaliação da arquitetura proposta.

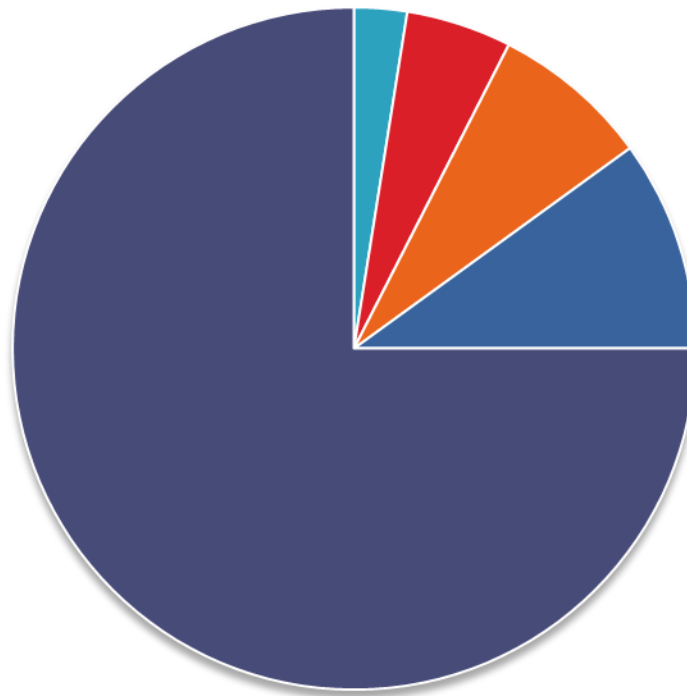
Fase de Transição

Fase de Transição



Fase de Transição

Esforço



- Requisitos
- Análise e Design
- Implementação
- Testes
- Implantação

Visão Geral

- ▶ O foco da fase de Transição é disponibilizar o software aos usuários finais
- ▶ Pode atravessar várias iterações e inclui testar os releases e ajustar pequenos defeitos com base no feedback do usuário
- ▶ O feedback prioriza apenas ajustes finos – todos os problemas estruturais mais graves devem ter sido trabalhados muito antes no ciclo de vida do projeto
- ▶ Pode ser uma fase muito fácil ou muito complexa, dependendo do tipo do produto

Objetivos

- ▶ Teste beta para validar o novo sistema
- ▶ Treinamento de usuários e equipe de manutenção
- ▶ Atividades de ajuste, como correção de erros, melhoria no desempenho e na usabilidade.
- ▶ Consentimento dos envolvidos que o software implantado atende a visão inicial

Principais artefatos

- ▶ Notas de Release
- ▶ Artefatos de instalação
- ▶ Material de treinamento

<<Transição>>

Ênfase na Disciplina:
Implantação

Implantação: Objetivos

- ▶ Coordenar e gerenciar os testes beta e de aceitação
- ▶ Desenvolver artefatos de instalação e materiais de treinamento
- ▶ Liberar para fabricação
- ▶ Há três tipos de instalação
 - a instalação personalizada
 - o produto em uma forma "compacta"
 - acesso ao software por meio da Internet

Implantação: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - **Gerente de Implantação**
 - Desenvolver plano de implantação
 - Escrever notas de release
 - **Desenvolver do curso**
 - Desenvolver materiais de treinamento
 - **Implementador**
 - Desenvolver artefatos de instalação

Implantação: Papeis, Atividades e Artefatos

- ▶ Artefatos importantes para o marco
 - O Build do produto
 - Notas de Release
 - Artefatos de instalação
 - Material de treinamento

Relação com outras disciplinas

▶ Requisitos

- As especificações de requisitos constituem a principal fonte para a elaboração de materiais de suporte e treinamento para o usuário final

▶ Teste

- Os testes são indispensáveis para a implantação do sistema

▶ Ambiente, Gerenciamento de Projeto, Ger. de Configuração e Mudanças

Marco da Implantação: Release do Produto

- ▶ Neste marco, você decide se os objetivos foram atendidos e se outro ciclo de desenvolvimento deve ser iniciado
- ▶ Critérios de avaliação
 - As despesas reais com recursos são aceitáveis se comparadas às planejadas?
 - O usuário está satisfeito?

Exercícios [4]

(CGU – ESAF 2008)

[25] No RUP (Rational Unified Process), dois dos exemplos dos artefatos de Implantação são:

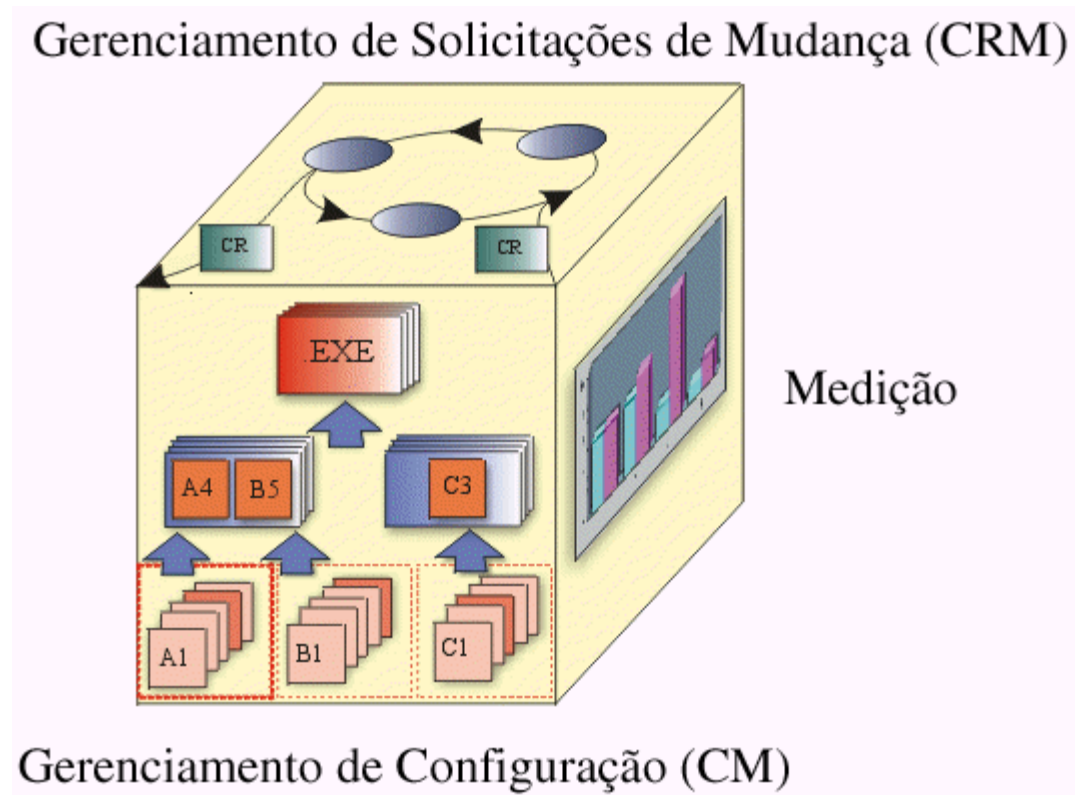
- A) Guia de design e Arte final do produto.
- B) Material de suporte para o usuário e Guia de teste.
- C) Plano de implantação e Manual de guia de estilo.
- D) Notas de release e Materiais de treinamento.
- E) Artefatos de Instalação e Guia de ferramentas.

Disciplinas de Suporte:

Gestão de Configuração e Mudanças

Visão Geral

- ▶ Controla mudanças feitas nos artefatos de um projeto e mantém a integridade entre eles



Objetivos

- ▶ Identificar e controlar itens de configuração
- ▶ Restringir as mudanças nesses itens
- ▶ Auditar as mudanças nesses itens
- ▶ Evitar confusões de:
 - Atualização simultânea
 - Notificação limitada
 - Várias versões

Benefícios

- ▶ Preservação da estabilidade e integridade do produto
- ▶ Suporte a métodos de desenvolvimento
- ▶ Restrição das mudanças feitas nos artefatos com base nas políticas do projeto
- ▶ Trilha de auditoria indicando por que, quando e por quem um artefato foi alterado

Gestão de Configuração e Mudanças: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - **Gerente Configuração**
 - Configurar ambiente de CM
 - Estabelecer políticas de CM
 - **Gerente de Mudanças**
 - Estabelecer processo de controle de mudanças
 - Confirmar ou recusar CR
- ▶ Principais artefatos
 - Repositório do projeto
 - Solicitações de mudanças

Disciplinas de Suporte:

Gerenciamento de Projeto

Objetivos

- ▶ Gerenciar pessoas, equipes fases e iterações para executar e monitorar o projeto
- ▶ Planejar o cronograma do projeto
- ▶ Gerenciar a qualidade e realizar revisões
- ▶ **Gerenciar os riscos do projeto**

Atenção!

- ▶ Não cobre problemas como:
 - Gerenciamento de Pessoal: contratação, treinamento, ensino
 - Gerenciamento de Orçamento: definição, alocação, etc.
 - Gerenciamento de contratos com fornecedores e clientes

É para isso que temos o PMBOK!

Gestão de Projetos: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - **Gerente de Projeto**
 - Planejar Fases e Iterações
 - Identificar e Avaliar Riscos
 - Monitorar o Projeto e Resolver Problemas
- ▶ Principais artefatos
 - Plano de Desenvolvimento de Software
 - Planos de Iteração
 - Lista de Riscos

Disciplinas de Suporte: Ambiente

Objetivos

- ▶ **Configurar o processo para o projeto**
- ▶ Selecionar e adquirir ferramentas
- ▶ Desenvolver ou adaptar templates específicos para o projeto
- ▶ Desenvolver ou adaptar guias de atividades
- ▶ Oferecer à organização o ambiente de desenvolvimento de software que dará suporte à equipe de desenvolvimento.

Ambiente: Papeis, Atividades e Artefatos

- ▶ Principais Papeis e atividades
 - **Engenheiro de Processo**
 - Configurar o processo
 - Iniciar Caso de Desenvolvimento
 - **Especialista em Ferramentas**
 - Selecionar, adquirir e configurar ferramentas
- ▶ Principais artefatos
 - Caso de Desenvolvimento

Caso de Desenvolvimento

- ▶ Descreve o processo de desenvolvimento escolhido para ser seguido no projeto
- ▶ A finalidade é capturar o processo adaptado para o projeto individual
- ▶ É criado no início da fase de Iniciação e atualizado durante todo o projeto
 - À medida que o projeto progride, você adiciona mais disciplinas a cada iteração
- ▶ **É o documento que “configura” o próprio processo de desenvolvimento**

Exercícios [5]

(IJSN – CESPE 2010)

[55] Na disciplina gerência de configuração e mudanças do RUP (rational unified process), garantir a integridade dos artefatos relacionados ao projeto de software é função da gerência de solicitação de mudanças.

Exercícios [5]

(TJ/SE – FCC 2009)

[54] De acordo com o RUP, balancear objetivos, administrar riscos e superar restrições para entregar um produto que atenda às necessidades de clientes e usuários é papel do

- (A) Gerente de Projetos.
- (B) Analista de Sistemas.
- (C) Administrador de Dados.
- (D) Analista de Requisitos.
- (E) Arquiteto de Software.

Exercícios [5]

(BNDES – CESGRANRIO 2008)

[1] Considerando o processo de desenvolvimento de software unificado, associe cada produto de trabalho com a fase em que deve Ser realizado. Marque a opção que ilustra a associação correta.

- a) I-P, II-S, III-R, IV-P, V-Q
- b) I-P, II-S, III-Q, IV-P, V-Q
- c) I-P, II-R, III-Q, IV-P, V-R
- d) I-Q, II-R, III-Q, IV-P, V-R
- e) I-Q, II-S, III-R, IV-Q, V-S

ARTEFATOS	FASES
I - Avaliação inicial de riscos	P - Concepção
II - Relatório de execução de testes beta	Q - Elaboração
III - Modelo de projeto completo	R - Construção
IV - Modelo de negócio preliminar	S - Transição
V - Protótipo arquitetural executável	

Gabaritos dos Exercícios

- ▶ [1] 53 C, 72 C, 22 A, 31 C
- ▶ [2] 73 C, 57 E, 54 E, 39 E
- ▶ [3] 98 E, 34 C, 73 E
- ▶ [4] 25 D
- ▶ [5] 55 E, 54 A, 1 A

FIM