

Planejamento Iterativo

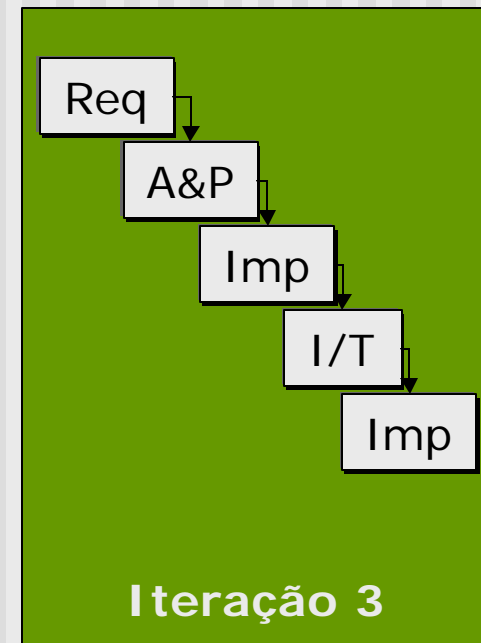
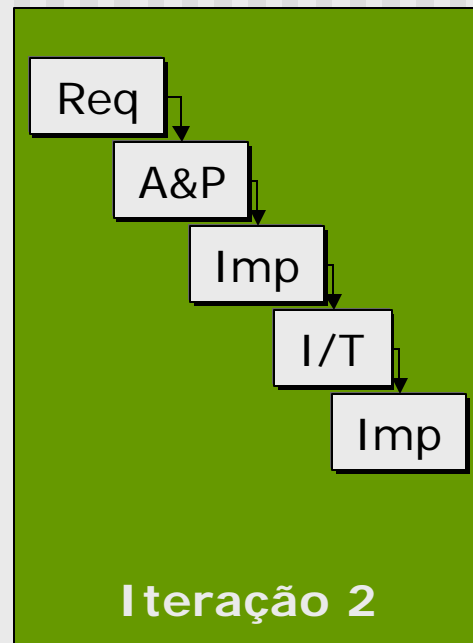
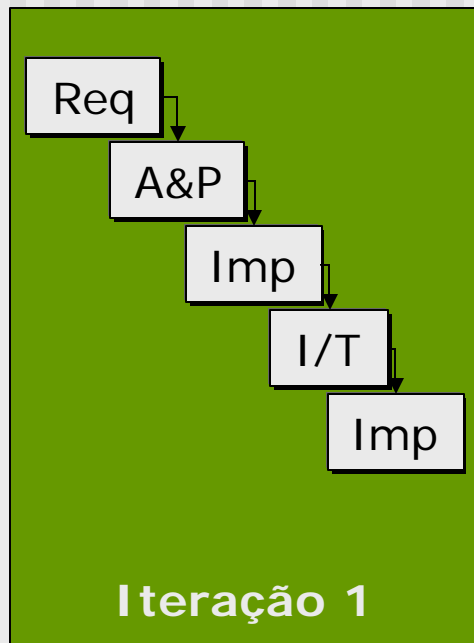
Planejando as Fases e
Iterações

Hermano Perrelli

hermano@cin.ufpe.br

Revisando...

Processo iterativo



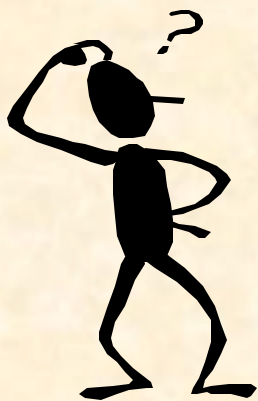
→
TEMPO

Revisando...

Fases e iterações

Concepção

Estabelecer o escopo e viabilidade econômica do projeto



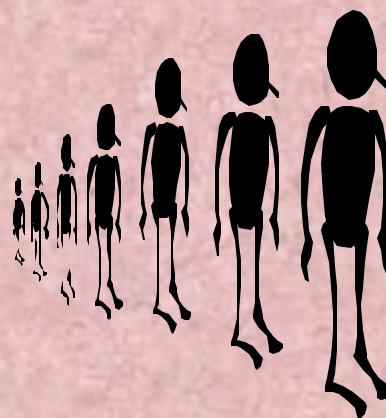
Elaboração

Eliminar principais riscos e definir arquitetura estável



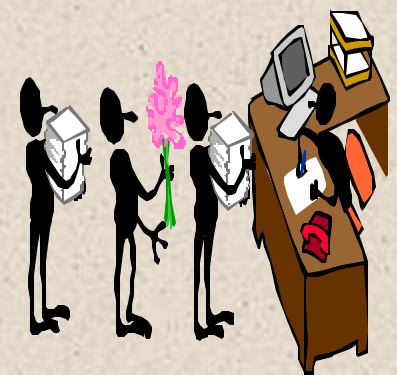
Construção

Desenvolver o produto até que ele esteja pronto para beta testes



Transição

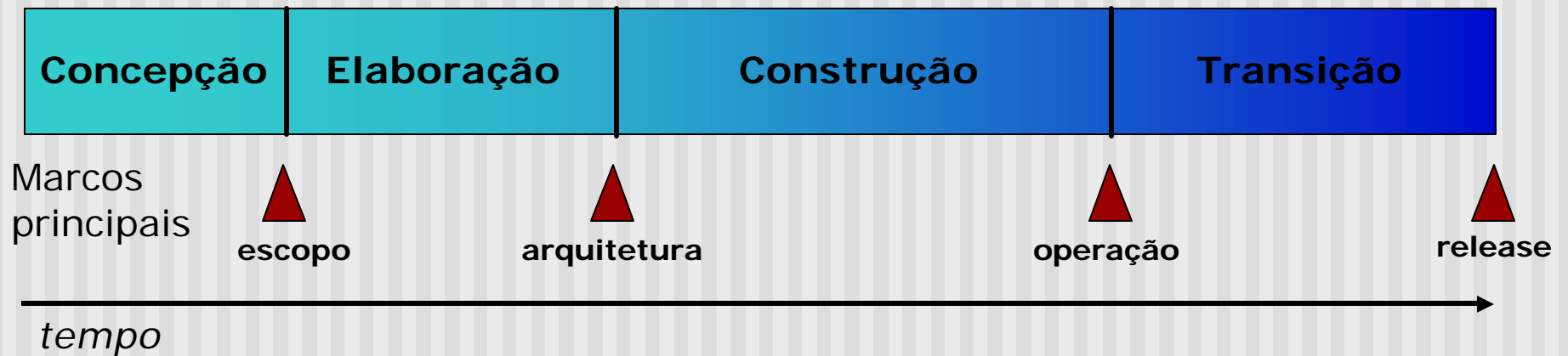
Entrar no ambiente do usuário



Revisando...

Fases e iterações

- O ciclo de vida de um sistema consiste de quatro fases:



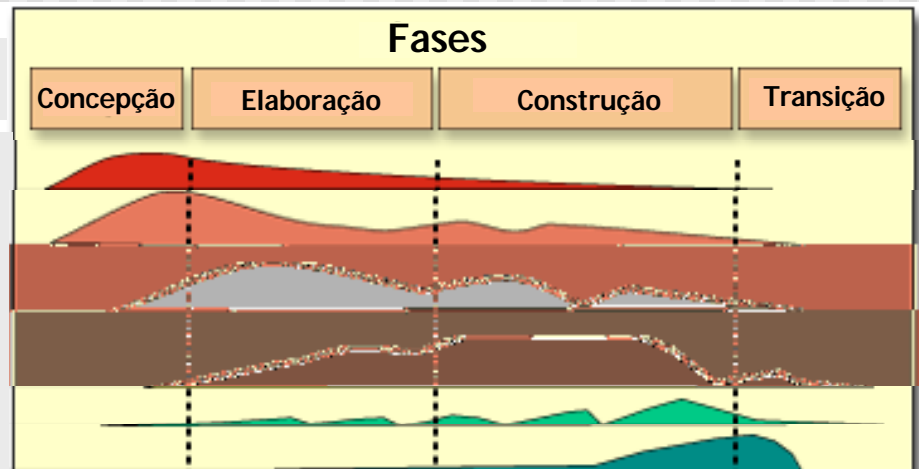
- As fases indicam a maturidade do sistema!

Revisando...

Fases, iterações e fluxos de atividades

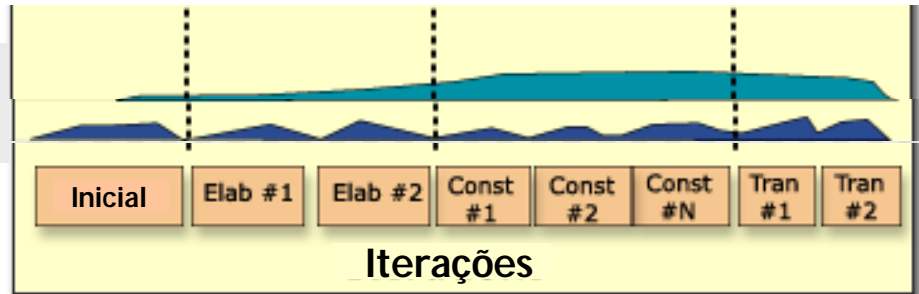
Fluxos Principais

Modelagem de Negócio.....
 Requisitos.....
 Análise e Projeto.....
 Implementação.....
 Testes.....
 Implantação.....



Fluxos de Apoio

Gerência de Configuração.....
 Planejamento e Gerenciamento.....



Objetivos desta aula

- Apresentar padrões do ciclo de vida de projetos iterativos
- Discutir a estrutura de cronogramas iterativos e incrementais
- Responder a perguntas comumente feitas durante o planejamento de projetos iterativos
 - Como definir a quantidade e duração das iterações?
 - O quanto realizar de cada fluxo de atividades em cada fase/iteração?
 - Como sincronizar o trabalho da equipe? Não é possível obter mais paralelismo e produtividade?
 - O cliente não topa esta estória de artefatos “incrementais”! Não quer revisar os requisitos diversas vezes, só enxerga retrabalho, acha que estamos perdendo seu tempo... Que fazer?

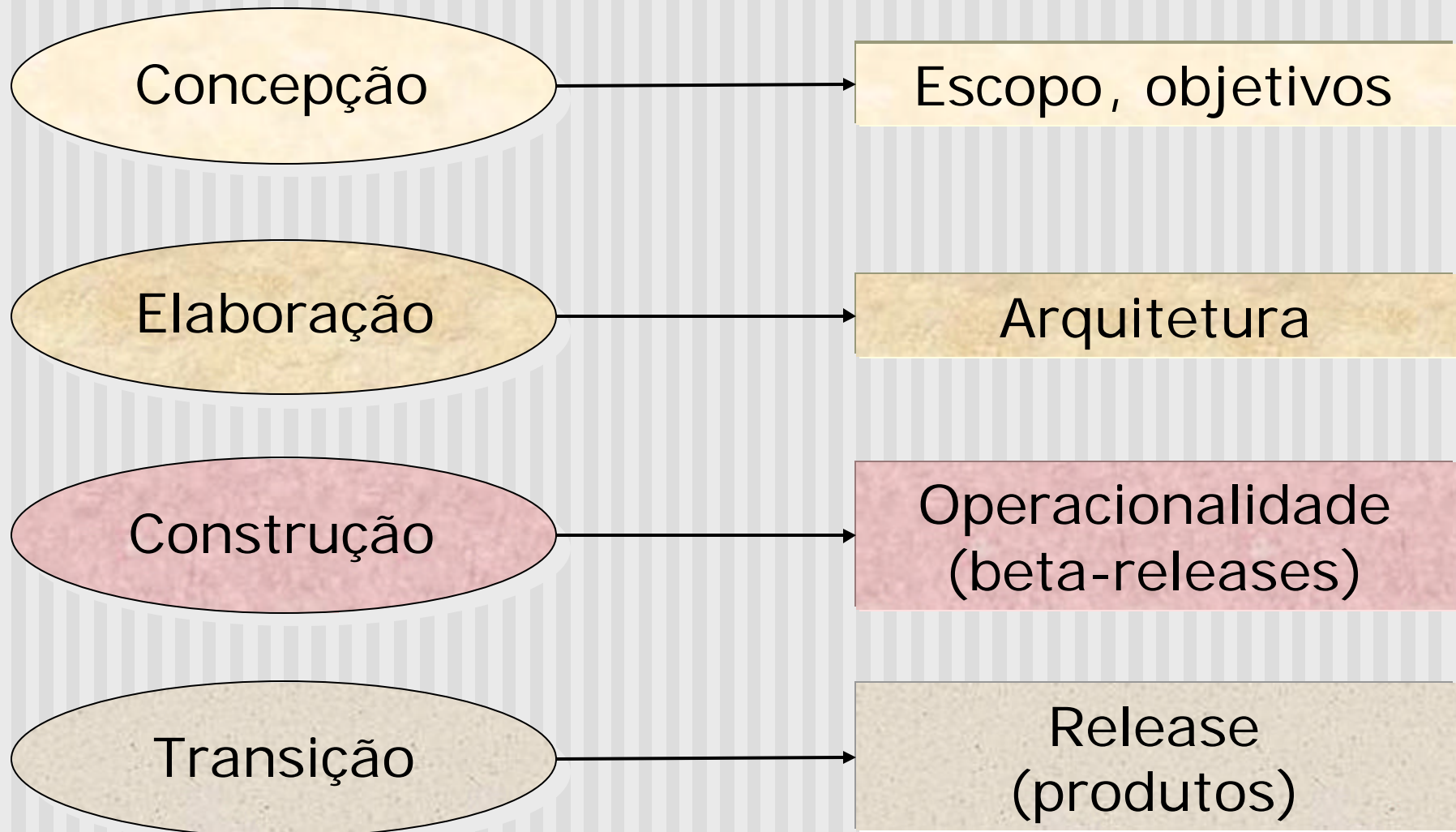
Agenda

- Padrões de ciclo de vida
- Cronogramas iterativos e incrementais
- Quantidade e duração das iterações
- Estratégias para as iterações
- Sincronização do trabalho
- Comunicação com o cliente/usuário
- Resumindo...

Padrões de ciclo de vida

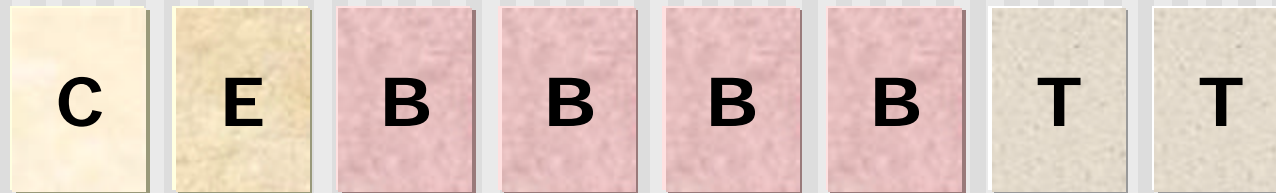
- Ferramenta para auxiliar no planejamento das fases
- Dependem das características do projeto
- Exemplos:
 - Incremental
 - Entrega incremental
 - Evolucionário
 - Híbridos

Para começar, não esqueça!



Ciclo de vida incremental

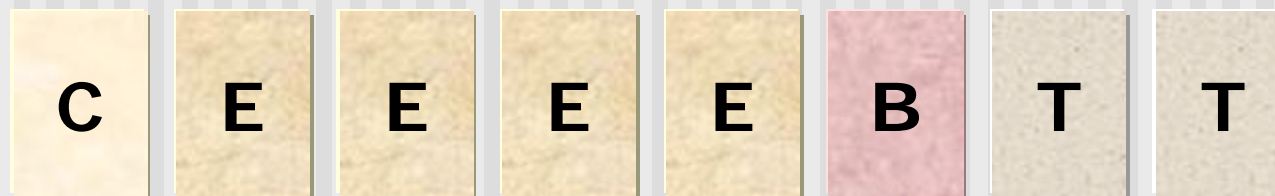
- O domínio do problema é conhecido, familiar
- Os riscos estão bem entendidos e razoavelmente controlados
- A equipe é experiente



Determinar as necessidades dos usuários, definir os requisitos e desenvolver o sistema como uma seqüência de builds, cada um acrescentando mais funcionalidades, até completar o produto.

Ciclo de vida evolucionário

- O domínio do problema é novo ou desconhecido
- A equipe é inexperiente



Entrega incremental

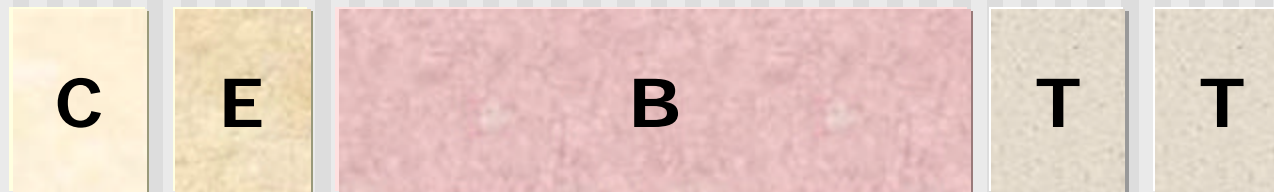
- O domínio do problema é conhecido, familiar
- Os requisitos e a arquitetura podem ser estabilizados bem cedo, durante o desenvolvimento (não existe muita novidade no sistema)
- A equipe é experiente
- É preciso liberar Releases incrementais do produto



Bastante útil para produtos já conhecidos que, para ganhar mercado, precisam liberar certas funcionalidades em um curto espaço de tempo.

“Grande Projeto”

- Um pequeno conjunto de funcionalidades vai ser adicionado a um produto já estável
- As novas funcionalidades são bem conhecidas e entendidas
- A equipe é experiente, tanto no domínio do problema quanto no produto já existente



O ciclo de vida Cascata pode ser visto como um caso particular deste, onde existe apenas uma única – e loooooonga – iteração na Construção.

Estratégias Híbridas

- Na prática, poucos projetos seguem apenas uma dessas estratégias de ciclo de vida
- A regra geral é:
 - para sistemas onde existe alto risco associado ao *negócio* do desenvolvimento:
Ênfase na Concepção
 - para sistemas complexos ou onde não se tem domínio do problema:
Ênfase na Elaboração
 - para sistemas onde se espera maior complexidade/esforço na produção de código:
Ênfase na Construção
 - para sistemas onde é preciso entregar o produto em uma série de releases incrementais:
Ênfase na Transição



Cronogramas iterativos e incrementais

- Bem mais complexos que os tradicionais cronogramas em cascata
- Normalmente organizados por fases e iterações

Cronogramas iterativos e incrementais

- Concepção
 - Iteração 1
 - atividade X
 - atividade Y
 - atividade Z
- Elaboração
 - Iteração 2
 - Iteração 3
- Construção
 - Iteração 4
 - Iteração 5
 - Iteração 6
- Transição
 - Iteração 7

- O cronograma não é feito todo de uma vez!

Lembre-se:
o processo
é iterativo!

Cronogramas iterativos e incrementais

- Concepção
 - Iteração 1
 - atividade A
 - atividade B
 - atividade C
- Elaboração
 - Iteração 2
 - atividade D
 - atividade B
 - atividade E
 - Iteração 3
- Construção
 - Iteração 4
 - Iteração 5
 - Iteração 6
- Transição
 - Iteração 7

- Devido a natureza do processo, várias atividades vão ficar repetidas

As atividades serão as mesmas, mas com *escopos/objetivos diferentes*

Cronogramas iterativos e incrementais

- A maneira de organizar as atividades no cronograma pode variar bastante
- Nível de detalhes
 - por atividade x por passos das atividades
 - por artefatos
- Agrupamento
 - por fluxo de atividades x por casos de uso
- Granularidade
 - casos de uso x cenários

caso de uso/cenário ...

	Nome da tarefa	Duração	Início	Término	Precede	Nomes de recursos
8	Validação do escopo do sistema	1 dia	Qua 12/6/00	Qua 12/6/00	10	Cliente,Usuário final,F
9	<input type="checkbox"/> Elaboração	9 dias	Ter 12/5/00	Sex 12/15/00		
0	<input type="checkbox"/> Iteração 2	9 dias	Ter 12/5/00	Sex 12/15/00		
1	<input type="checkbox"/> Atividades do fluxo de Requisitos	2 dias	Ter 12/5/00	Qua 12/6/00		
2	<input type="checkbox"/> Detalhar especificação de caso de uso/cenário	1 dia	Ter 12/5/00	Ter 12/5/00		
3	caso de uso/cenário A	1 dia	Ter 12/5/00	Ter 12/5/00		Desenvolvedor
4	caso de uso/cenário B	1 dia	Ter 12/5/00	Ter 12/5/00		Desenvolvedor
5	caso de uso/cenário ...	1 dia	Ter 12/5/00	Ter 12/5/00		Desenvolvedor
6	Opcional - Estruturar modelo de casos de uso	1 dia	Qua 12/6/00	Qua 12/6/00	22	Desenvolvedor
7	Opcional - Prototipar interface com o usuário	1 dia	Qua 12/6/00	Qua 12/6/00	22	Projetista da interface
8	Revisar requisitos	1 dia	Ter 12/5/00	Ter 12/5/00		Revisor,Cliente,Usuár
9	<input type="checkbox"/> Atividades do fluxo de Análise e projeto	4 dias	Ter 12/5/00	Sex 12/8/00		
0	<input type="checkbox"/> Analisar caso de uso/cenário	1 dia	Qua 12/6/00	Qua 12/6/00		
1	caso de uso/cenário A	1 dia	Qua 12/6/00	Qua 12/6/00	23	Desenvolvedor
2	caso de uso/cenário B	1 dia	Qua 12/6/00	Qua 12/6/00	24	Desenvolvedor
3	caso de uso/cenário ...	1 dia	Qua 12/6/00	Qua 12/6/00	25	Desenvolvedor
4	Projetar arquitetura	1 dia	Qui 12/7/00	Qui 12/7/00	30	Arquiteto
5	<input type="checkbox"/> Projetar caso de uso/cenário	1 dia	Sex 12/8/00	Sex 12/8/00	34	
6	caso de uso/cenário A	1 dia	Sex 12/8/00	Sex 12/8/00	31	Desenvolvedor
7	caso de uso/cenário B	1 dia	Sex 12/8/00	Sex 12/8/00	32	Desenvolvedor
8	caso de uso/cenário ...	1 dia	Sex 12/8/00	Sex 12/8/00	33	Desenvolvedor
9	<input type="checkbox"/> Projetar subsistema	1 dia	Sex 12/8/00	Sex 12/8/00	34	
0	subsistema P	1 dia	Sex 12/8/00	Sex 12/8/00		Desenvolvedor
1	subsistema Q	1 dia	Sex 12/8/00	Sex 12/8/00		Desenvolvedor
2	subsistema ...	1 dia	Sex 12/8/00	Sex 12/8/00		Desenvolvedor
3	Projetar base de dados	1 dia	Ter 12/5/00	Ter 12/5/00		Projetista da base de
4	Revisar projeto	1 dia	Ter 12/5/00	Ter 12/5/00		Revisor
5	<input type="checkbox"/> Atividades do fluxo de Implementação	9 dias	Ter 12/5/00	Sex 12/15/00		
8	<input type="checkbox"/> Atividades do fluxo de Testes	2 dias	Ter 12/5/00	Qua 12/6/00		
1	<input type="checkbox"/> Atividades do fluxo de Planejamento e gerenciamento	2 dias	Ter 12/5/00	Qua 12/6/00		
	<input type="checkbox"/> Atividades do fluxo de gerência de ambiente e controle de versões	1 dia	Ter 12/5/00	Ter 12/5/00		

Algumas perguntas comuns...

Como definir a quantidade e duração das iterações?

- Iterar é bom, mas acrescenta certo overhead!
 - planejamento
 - avaliação
 - sincronização de atividades
- A agilidade para iterar depende basicamente de:
 - tamanho da equipe
 - experiência com o processo
- A complexidade e conhecimento do produto também pesam
 - estratégias de ciclo de vida

Duração das iterações

- Alguns dados da Rational:

Linhas de código	Equipe	Duração de 1 iteração
10.000	5	1 semana
50.000	15	1 mês
500.000	45	6 meses
1.000.000	100	1 ano

Duração das iterações

- Iterações pequenas (menores que um mês)
 - planejar bem o escopo
 - típicas da Construção
 - com pouca adição de funcionalidades
 - com pouca/nenhuma atividade formal de análise e projeto
- Iterações grandes (maiores que seis meses)
 - adicionar marcos (milestones) intermediários
 - o foco da iteração está claro?!
 - considerar a possibilidade de reduzir o escopo da iteração (e reduzir a sua duração)

**As iterações não precisam ter
todas o mesmo tamanho!**

Exceto dentro da mesma fase!

Quantidade de iterações

- Projetos simples normalmente têm uma iteração por fase
- Projetos mais complexos, no seu primeiro ciclo de desenvolvimento normalmente apresentam:
 - 1 iteração na Concepção
 - 2 iterações na Elaboração
 - 2 iterações na Construção
 - 1 iteração na Transição
- Projetos grandes, em domínios desconhecidos, envolvendo novas tecnologias e muitos riscos:
 - 2 iteração na Concepção
 - 3 iterações na Elaboração
 - 3 iterações na Construção
 - 2 iteração na Transição

Quantidade de iterações

- Resumindo...
- Projetos simples: 3/4 iterações [0/1, 1, 1, 1]
- Projetos típicos: 6 iterações [1, 2, 2, 1]
- Projetos grandes: 10 iterações [2, 3, 3, 2]

Em geral, planeja-se de 3 a 10 iterações!

Na maioria dos casos temos de 6 a 8 iterações!



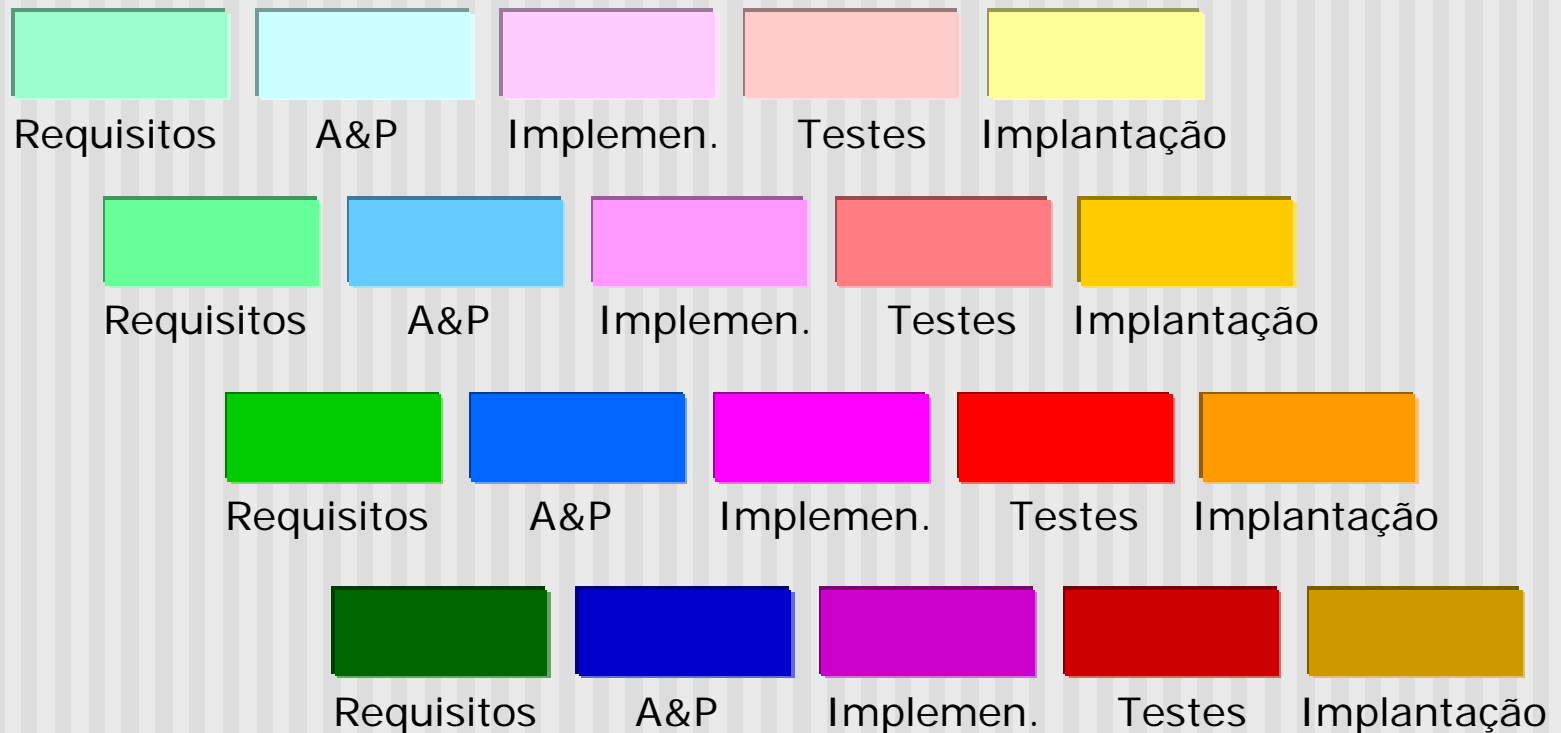
O quanto realizar de cada fluxo de atividades em cada fase/iteração?

■ Desenvolvimento em Cascata



O quanto realizar de cada fluxo de atividades em cada fase/iteração?

■ Desenvolvimento iterativo e incremental

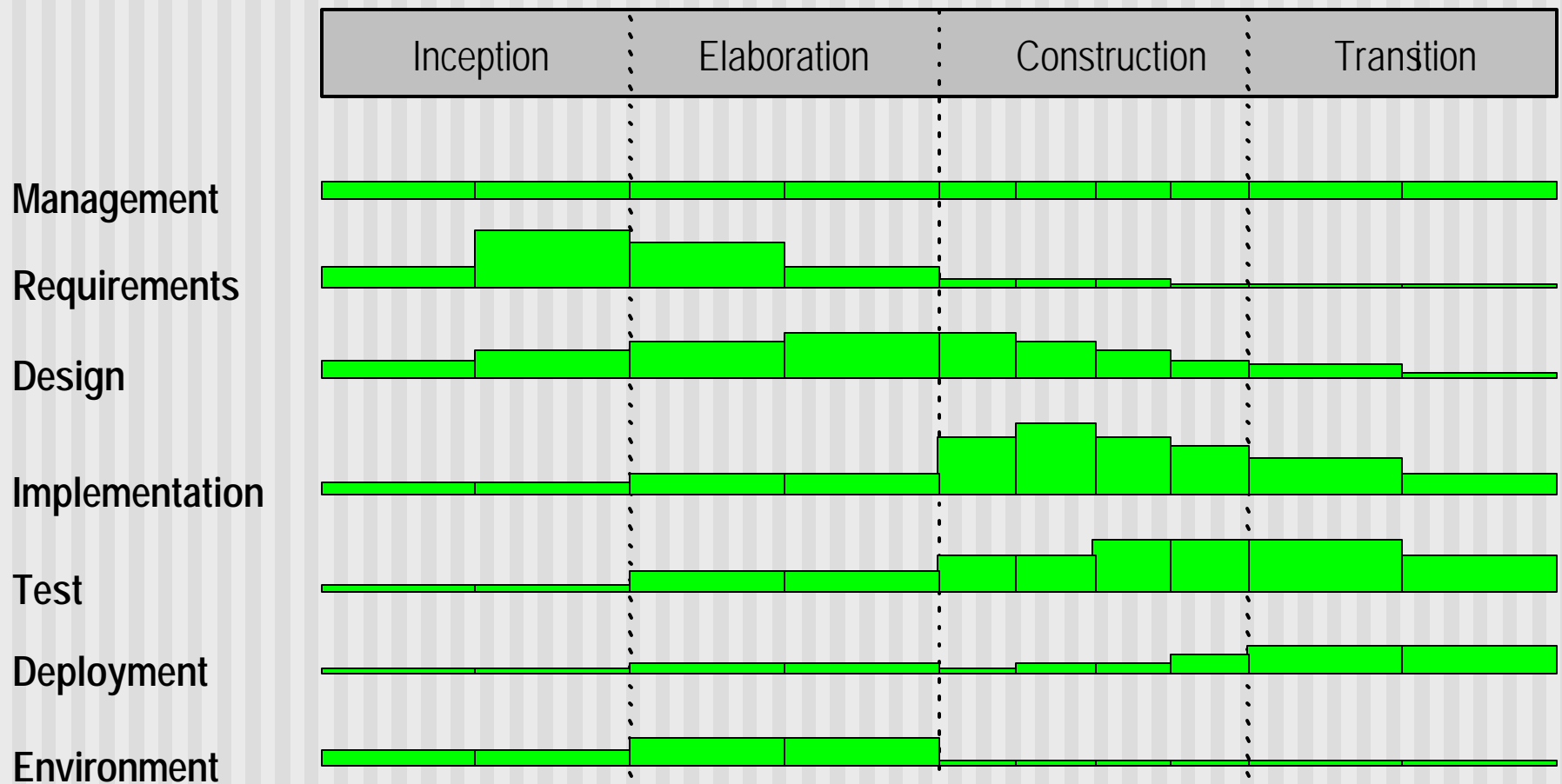


Fluxos de atividades e fases

Em geral:

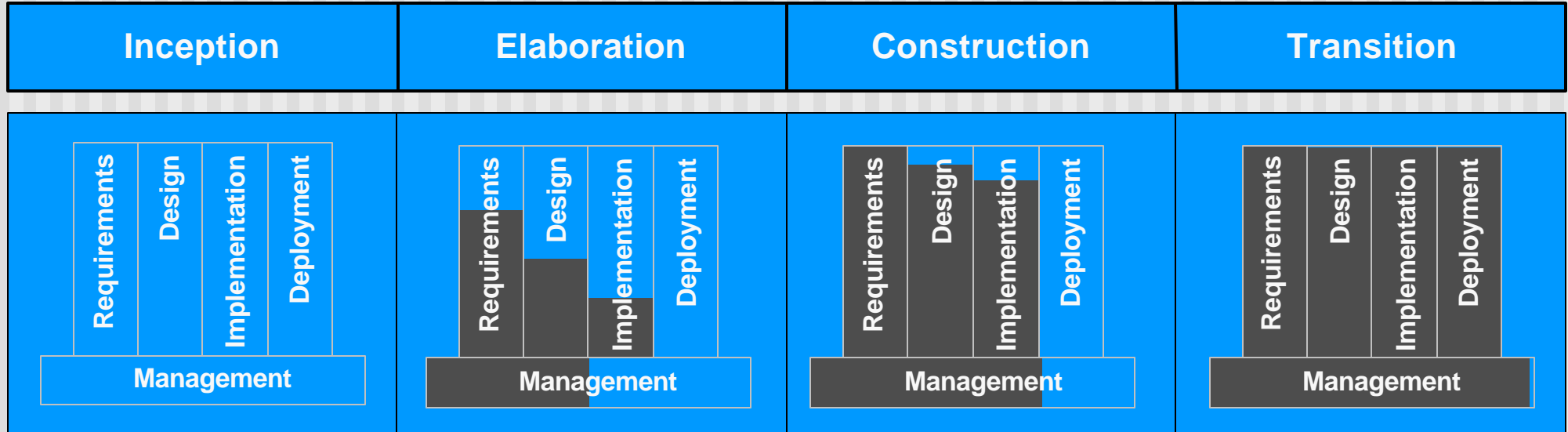
- Requisitos – concentração na:
 - Concepção e Elaboração
- Análise e projeto – concentração na:
 - Elaboração
- Implementação – concentração na:
 - Construção
- Testes – concentração na:
 - Construção
- Distribuição – concentração na:
 - Transição

Esforço dos fluxos de atividades por fase



Fonte: Software Project Management, Walker Royce

Completude dos artefatos por fase



Fonte: Software Project Management, Walker Royce

O quanto realizar de cada fluxo de atividades em cada fase/iteração?

- De maneira geral, em cada iteração um subconjunto do trabalho total é realizado
 - levantado/especificado
 - analisado e projetado
 - implementado
 - testado
 - preparado para a distribuição/distribuído
- Como escolher esse subconjunto?
 - conhecimento da equipe no domínio do problema e arquitetura a ser adotada
 - necessidade de liberação de releases/ deadline restrito

Estratégias para as iterações

- Larga e superficial
 - Todo o domínio do problema é analisado, mas sem muitos detalhes
 - Casos de uso: todos são definidos e a maioria é detalhada
 - Arquitetura: definida amplamente – todas as interfaces, serviços, etc.
 - Muito pouca implementação até a Construção, onde fica o maior número de iterações
- Estreita e profunda
 - Um pedacinho do domínio é analisado em detalhes
 - Os casos de uso relacionados com este pedacinho são detalhados
 - A arquitetura necessária para suportar esse pedacinho é definida
 - Esse pedacinho é implementado, testado e possivelmente implantado

Híbrida

Larga e superficial

- Apropriada quando:
 - o time é inexperiente no domínio do problema ou nas tecnologias que serão usadas
 - a arquitetura é inédita, ou é um requisito chave para as funcionalidades do sistema
- Possíveis problemas:
 - *analysis paralysis*
 - falta de credibilidade e confiança da equipe
 - riscos técnicos não expostos devido a falta de detalhes (visão apenas de alto nível)

Estreita e profunda

- Apropriada quando:
 - precisa-se de resultados muito rápido (para obter suporte, provar viabilidade ou eliminar certos riscos)
 - os requisitos estão continuamente evoluindo
 - o *deadline* é obrigatório
 - existe alta reusabilidade
- Possíveis problemas:
 - dificuldades de integração
 - desenvolvimento de software integrado “verticalmente”, mas incompatível “horizontalmente”
 - muito retrabalho devido a falta de uma visão geral do problema

Estratégia híbrida

- Na Concepção:
 - larga e superficial para obter bom entendimento do escopo
 - estreita e profunda para verificar a viabilidade de alguma tecnologia
 - construção de um protótipo
- Na Elaboração:
 - na maior parte do tempo, larga e superficial, para garantir que a arquitetura cobre todas as necessidades
 - estreita e profunda em alguns pontos para atacar riscos específicos
- Na Construção:
 - estreita e profunda, para implementar as funcionalidades do sistema, com alto grau de paralelismo e incrementalmente
- Na Transição:
 - completar o que falta, de acordo com o feedback do usuário e *bugs* encontrados



Como sincronizar o trabalho?

Como obter maior paralelismo e produtividade?

- O que os meus analistas vão fazer enquanto os programadores implementam?
- Alguma sobreposição de atividades sempre vai existir...
 - O planejamento da próxima iteração vai se iniciar da metade para o final da iteração corrente
- Muita sobreposição leva a problemas!...
 - comprometimento das pessoas com os objetivos da iteração
 - falta de suporte para o trabalho que foi antecipado
 - não aproveitamento das “lições aprendidas” durante as iterações
- Alternativas:
 - aplicar a estratégia larga e superficial
 - alocar os recursos para outras atividades
 - planejamento e projeto de testes
 - elaboração da documentação do usuário
 - estimular a versatilidade dos profissionais



Como trabalhar com um cliente / usuário que não acredita no processo?

- O cliente não topa esta estória de artefatos “incrementais”! Não quer revisar os requisitos diversas vezes, só enxerga retrabalho, acha que estamos perdendo seu tempo... Que fazer?
- Deixe as intenções e planos bem visíveis
 - Um bom acompanhamento é essencial!
- Ofereça feedback concreto
 - protótipos, componentes, versões alfa do sistema
- Proteja a equipe
 - atue como filtro!
- Revise os artefatos em pontos apropriados
 - Requisitos: no final da Concepção ou +- em 1/3 da Elaboração
 - Arquitetura: no final da Elaboração
 - Projeto: +- em 1/3 da Construção



Planejamento iterativo

- Conheça os riscos
- Planeje as fases
 - duração e marcos (*milestones*)
 - quantidade de iterações
- Planeje a primeira iteração em detalhes
 - atividades, recursos, tempo, ...
- Durante a execução da primeira iteração, planeje a segunda em detalhes

E assim por diante...

Exercício: Planejamento das Fases e Iterações

- Detalhar/ajustar o planejamento do seu projeto, definindo:
 - a duração de cada fase
 - o número de iterações de cada fase
 - que estratégia será seguida para realizar as iterações
 - a duração de cada iteração
- Registre o que for assumido como pressuposto para suas decisões!



Créditos

- Qualiti Software Processes | www.qualiti.com
 - Pela primeira versão destes slides