

Engenharia de Software

Ciclos de Vida



Marcello Thiry
thiry@univali.br
marcello.thiry@gmail.com

LQPS
<http://www.univali.br/lqps>

Ciclo de Vida de um Produto

- ☐ Qualquer desenvolvimento de produto inicia com uma idéia e termina com o produto pretendido
- ☐ O ciclo de vida de um produto é a definição dos passos que transformam aquela idéia no produto acabado
- ☐ O modelo de ciclo de vida é o centro do processo de gerenciamento do produto

Ciclo de Vida do Software

- ☐ Todas as atividades e produtos de trabalho (artefatos) necessários para desenvolver e manter um sistema de software
- ☐ Um framework contendo os processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, considerando toda vida do software, desde a definição de seus requisitos até o encerramento de seu uso [ISO/IEC12207]
- ☐ É o particionamento da vida de um produto ou projeto em fases [SEI, Glossário do CMMI]
- ☐ Uma passagem completa por quatro fases: concepção, elaboração, construção e transição. O espaço de tempo entre o início da fase de concepção e o final da fase de transição. [IBM/Rational, Glossário do RUP]



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

3

Componentes do desenvolvimento

- ☐ **Fases:** passos que indicam o progresso do projeto
- ☐ **Atividades:** ações requeridas para criar e entregar o projeto
- ☐ **Artefatos:** produtos tangíveis criados durante o projeto (produtos de trabalho)
- ☐ **Marcos (*milestones*):** eventos importantes no projeto (momentos de decisão)



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

4

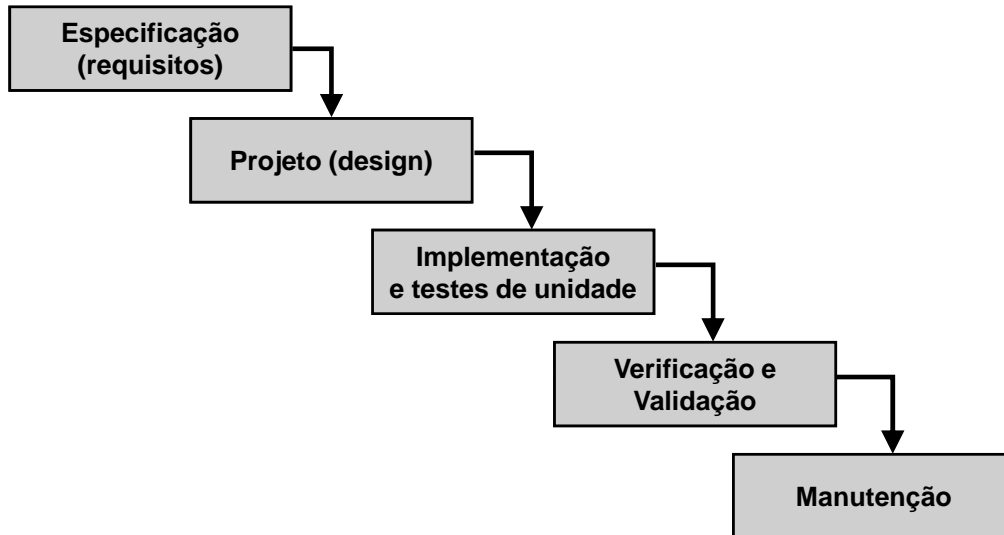
Diferentes modelos

- ☐ Diferentes modelos tem diferentes componentes e não existe um modelo correto
- ☐ É responsabilidade do gerente de projeto, verificar quais modelos são mais indicados para o projeto
- ☐ Ao final, o gerente deve combinar estes modelos, criando um modelo que seja adequado às necessidades do projeto

Expectativas sobre o ciclo de vida

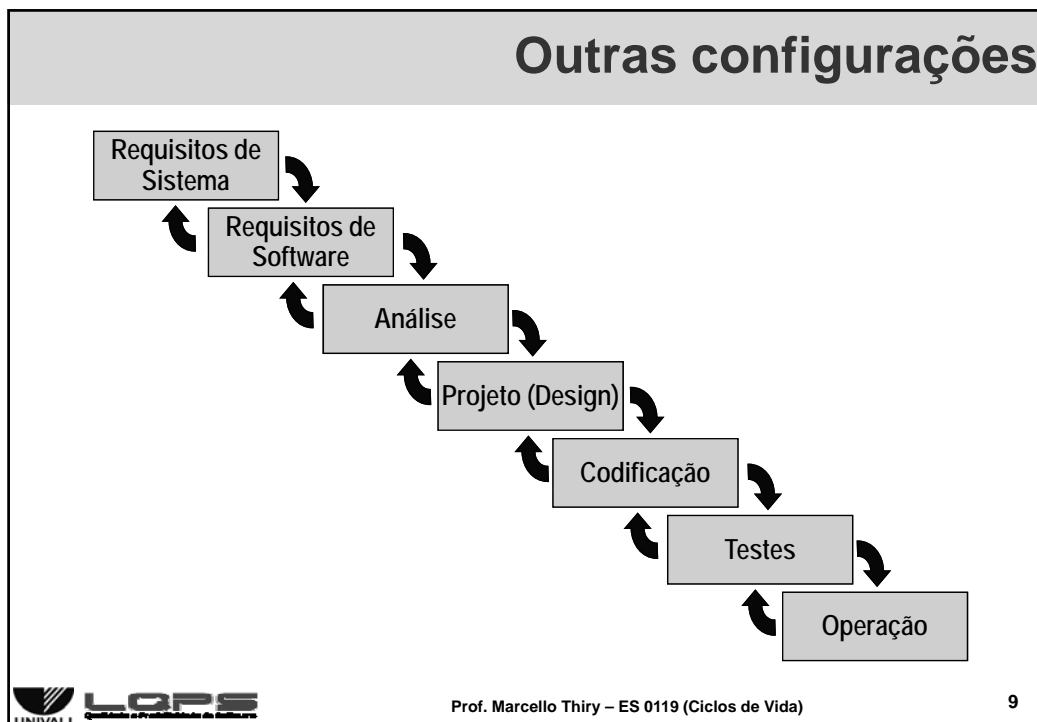
- ☐ Definição de pontos de controle
- ☐ Planejamento e acompanhamento do progresso
- ☐ Planejamento e acompanhamento do orçamento
- ☐ Estimativas
- ☐ Gerência de risco

Modelo Clássico/Cascata (Royce, 1970)



Fases do Modelo Cascata

1. **Especificação (requisitos):** identificação e análise dos requisitos do sistema de acordo com as necessidades dos usuários
2. **Projeto (design):** detalhamento da solução para o sistema a ser construído, incluindo a definição da arquitetura de software, padrões de projeto, algoritmos, etc.
3. **Implementação:** codificação da solução mapeada na fase de projeto, incluindo os testes realizados sobre os módulos desenvolvidos (testes de unidade)
4. **Verificação e Validação:** verificar se o que foi desenvolvido está de acordo com o que foi especificado e validar se o que foi desenvolvido atende realmente as necessidades do usuário; inclui testes de integração
5. **Manutenção:** evolução contínua do sistema, considerando também tratamento de erros e adaptações do sistema



Considerações sobre a Manutenção

- ☐ Deve ser considerada em todos os ciclos de vida
- ☐ **Não deve ser encarada como uma fase isolada**, mas a aplicação **contínua** das atividades envolvidas nas fases anteriores
- ☐ **Manutenção corretiva:** correção de erros encontrados (não deve ser o foco da atividade de manutenção)
- ☐ **Manutenção adaptativa:** adaptação do software em relação a uma mudança externa (legislação, novo sistema operacional)
- ☐ **Manutenção evolutiva:** melhoria contínua do software, incluindo novas funcionalidades, maior atenção à aspectos de qualidade, etc
- ☐ **Manutenção preventiva:** usualmente, é uma melhoria interna do sistema para otimizar algum recurso ou algoritmo; refactorings no código; pode ser considerada como uma manutenção evolutiva

UNIVALI LQPS Laboratório de Qualidade e Produtividade de Software

Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

10

Características do Modelo Cascata

- ☐ Baseado nos processos convencionais de outras engenharias
- ☐ Abordagem sistemática e seqüencial com fácil identificação dos marcos e dos **entregáveis** (*deliverables*)
- ☐ Fortemente documental, pouco iterativo
- ☐ Requer especificação completa e bem entendida
- ☐ Dificulta a introdução de mudanças após o início do processo (validação no final)



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

11

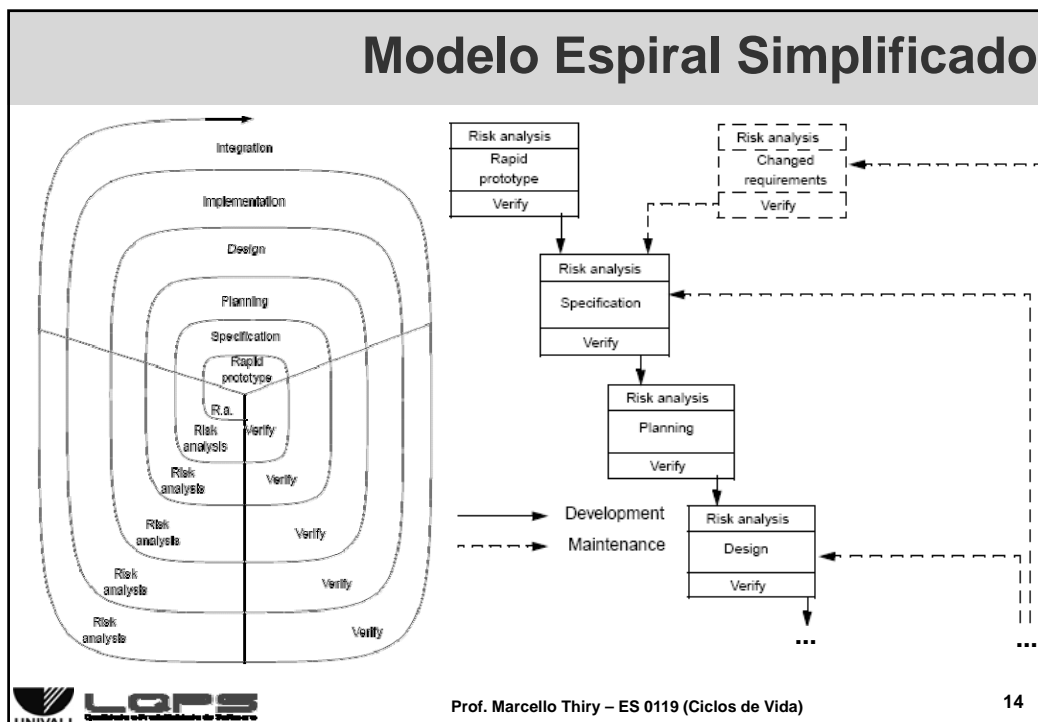
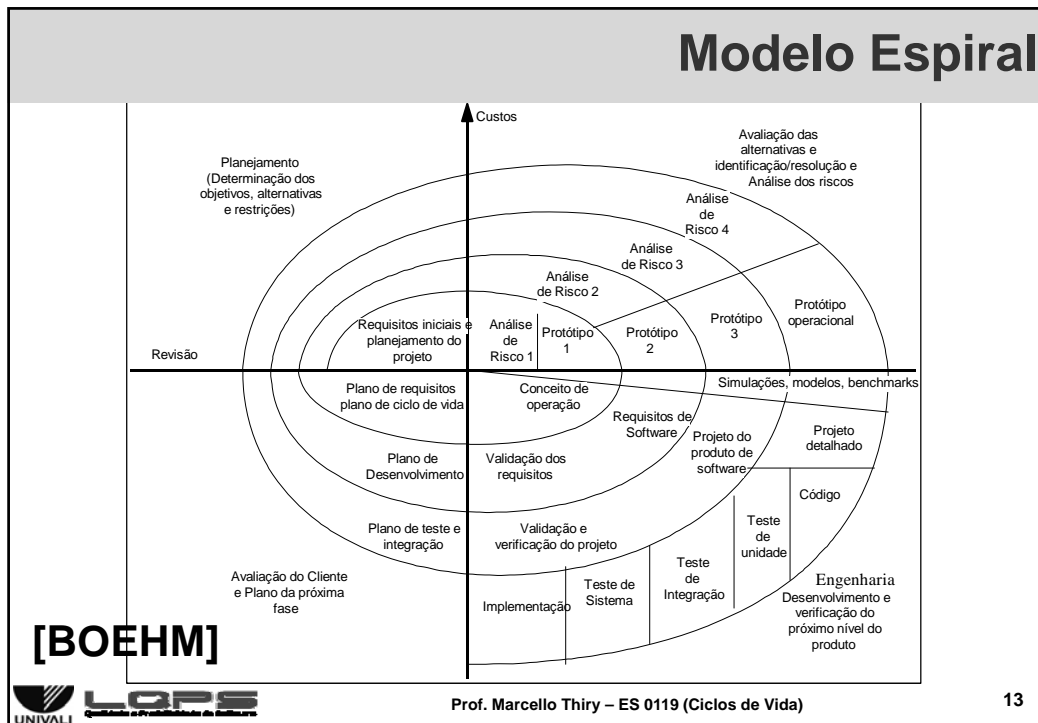
Modelo em Espiral (Boehm, 1986)

- ☐ Iterações através de quatro fases de alto nível:
 - ☐ determinação de objetivos: definição do produto, determinação de objetos de negócio, restrições
 - ☐ avaliação de alternativas: análise de risco, prototipação
 - ☐ desenvolvimento do produto: projeto detalhado, codificação, testes das unidades, integração
 - ☐ planejamento da próxima iteração: avaliação do cliente, planejamento do projeto, entrega ao cliente
- ☐ Premissa: 2 ou mais builds
- ☐ Raio (distância a partir do centro) = esforço gasto



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

12



Aspectos do Modelo Espiral

- ☐ Melhorias do modelo em cascata
- ☐ Oferece a oportunidade para o usuário/cliente se envolver
- ☐ Iterativo com releases incrementais e fortemente centrado na análise de riscos
- ☐ Difícil e alto custo para adoção (muitos estágios intermediários: problemas com documentação e treinamento)
- ☐ Pode não convergir para uma solução
- ☐ Não é largamente utilizado



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

15

RUP e o Modelo Iterativo Incremental

- ☐ O RUP (*Rational Unified Process*) é um framework de processo de desenvolvimento de software que fornece uma abordagem disciplinada para associar tarefas e responsabilidades dentro de uma organização de desenvolvimento
- ☐ Criado pela Rational, atualmente é um produto da IBM
- ☐ O RUP foi projetado para suportar a implementação de **melhores práticas**
- ☐ Adota o **ciclo de vida Iterativo Incremental**



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

16

RUP e o Modelo Iterativo Incremental

- ❑ O RUP (*Rational Unified Process*) é um framework de processo de desenvolvimento de software que fornece uma abordagem disciplinada para associar tarefas e responsabilidades dentro de uma organização de desenvolvimento
- ❑ Criado pela Rational, atualmente é um produto da IBM
- ❑ O RUP foi projetado para suportar a implementação de **melhores práticas**
- ❑ Adota o **ciclo de vida**

Um conjunto de abordagens provadas comercialmente para o desenvolvimento de software. Elas atacam as causas dos problemas tipicamente encontrados no desenvolvimento de software.



Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

17

Princípios do RUP

- ❑ Tratar os principais riscos desde o início e continuamente
- ❑ Assegurar algo de valor será entregue ao cliente
- ❑ Ficar focado no software executável (avaliar progresso)
- ❑ Tratar mudanças desde o início no projeto
- ❑ Estabelecer uma **baseline** da **arquitetura** desde o início
- ❑ Construir o sistema com **componentes**
- ❑ Trabalhar realmente em equipe
- ❑ Qualidade deve ser inerente a tudo




Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

18

Princípios do RUP

Uma liberação revisada e aprovada de artefatos que constituem uma base acordada para futura evolução ou desenvolvimento e que pode ser mudada por meio de um procedimento formal (gerência de mudança ou controle de configuração)


- ☐ Tratar mudanças desde o início no projeto
- ☐ Estabelecer uma **baseline** da **arquitetura** desde o início
- ☐ Construir o sistema com **componentes**
- ☐ Trabalhar realmente em equipe
- ☐ Qualidade deve ser inerente a tudo

 Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 19

Princípios do RUP

- ☐ Tratar os principais riscos desde o início e continuamente
- ☐ Assegurar algo de valor será entregue ao cliente
- ☐ Ficar focado no software executável (avaliar progresso)
- ☐ Tratar mudanças desde o início no projeto
- ☐ Estabelecer uma **baseline** da **arquitetura** desde o início
- ☐ Construir o sistema com **componentes**
- ☐ Trabalhar realmente em equipe
- ☐ Qualidade deve ser inerente a tudo

Organização dos elementos estruturais do sistema que compreende a colaboração (através de interfaces) entre estes elementos

 Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 20

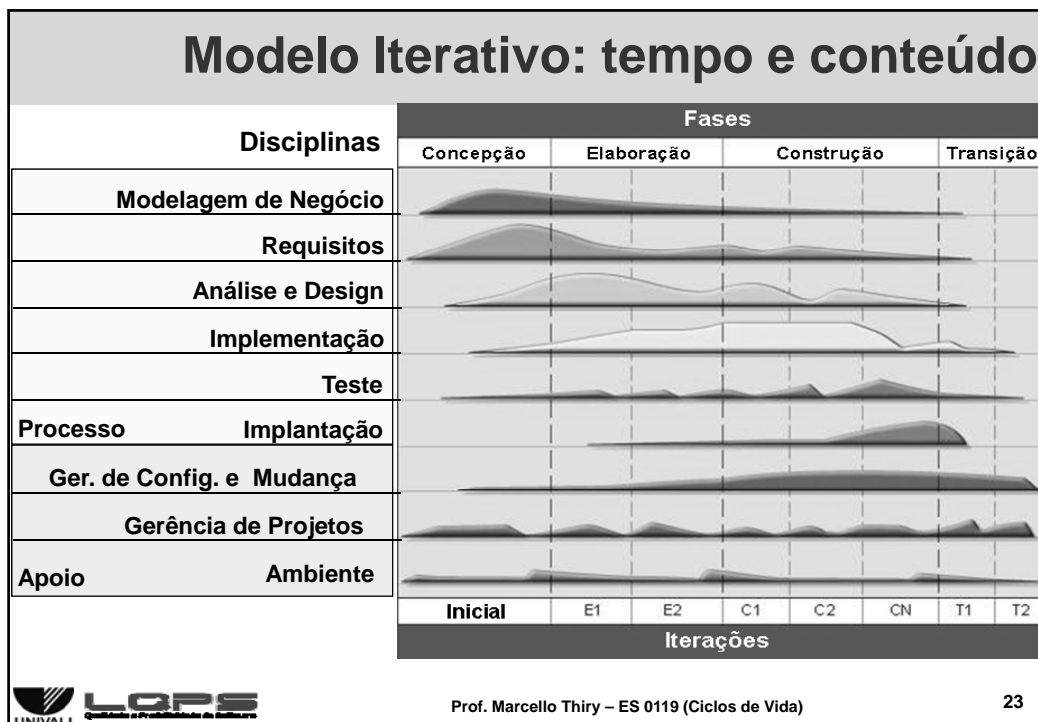
Princípios do RUP

- ☐ Tratar os principais riscos desde o início e continuamente
- ☐ Assegurar a qualidade
- ☐ Ficar atento a mudanças
- ☐ Tratar o sistema como uma coleção de componentes
- ☐ Estabelecer uma arquitetura desde o início
- ☐ Construir o sistema com componentes
- ☐ Trabalhar realmente em equipe
- ☐ Qualidade deve ser inerente a tudo

Uma parte do sistema intercambiável, não trivial e com um grau de independência, que preenche uma função clara no contexto de uma arquitetura bem definida. Um componente atende e fornece a realização física de um conjunto de interfaces.

RUP: 6 melhores práticas

- ☐ Desenvolver iterativamente
- ☐ Gerenciar requisitos
 - ☐ cobrir o desenvolvimento de requisitos
- ☐ Utilizar arquitetura de componentes
 - ☐ orientar as atividades de projeto (design)
- ☐ Modelar visualmente (UML)
- ☐ Verificar a qualidade continuamente
- ☐ Gerenciar configuração e mudanças



Iterações

- ☐ Dentro de um ciclo de vida iterativo, vários passos são feitos através de cada disciplina. Cada um destes passos é chamado iteração
- ☐ Uma iteração é uma sequência distinta e com duração fixa de atividades que resulta em uma **liberação/release** (interna ou externa) de um produto executável
- ☐ Com o progresso do projeto, liberações evoluem de um subconjunto do produto final para o sistema final
- ☐ Cada iteração resulta em um melhor entendimento dos requisitos, uma arquitetura mais robusta, uma organização de desenvolvimento mais experiente e uma implementação mais completa

LQPS Laboratório de Qualidade e Produtividade de Software

Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 24

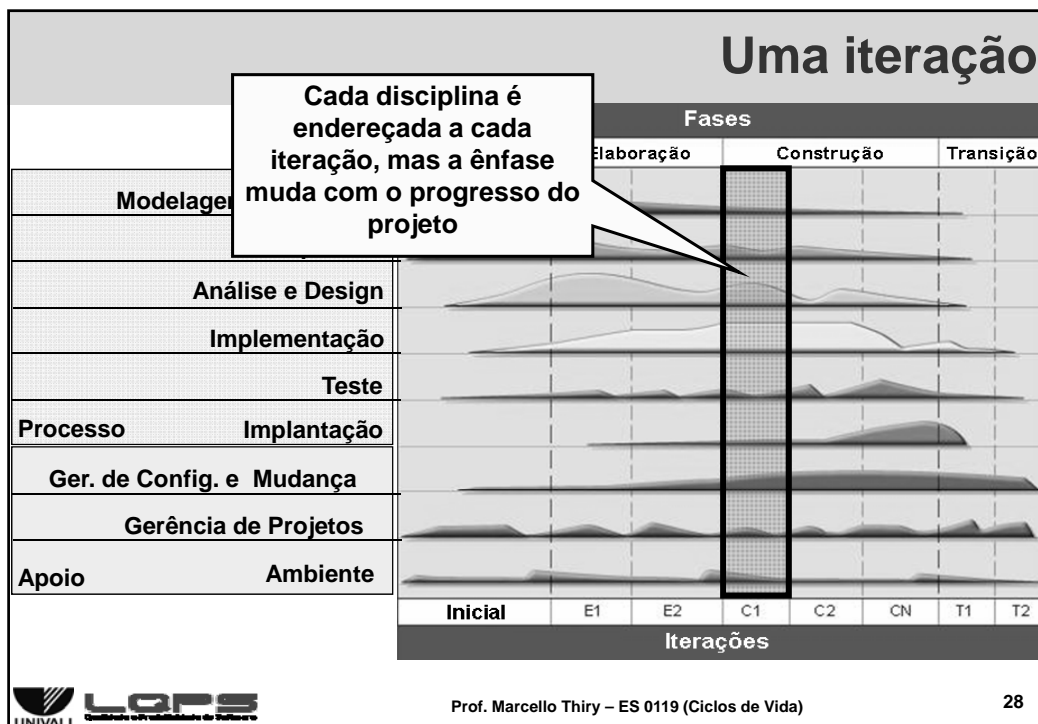
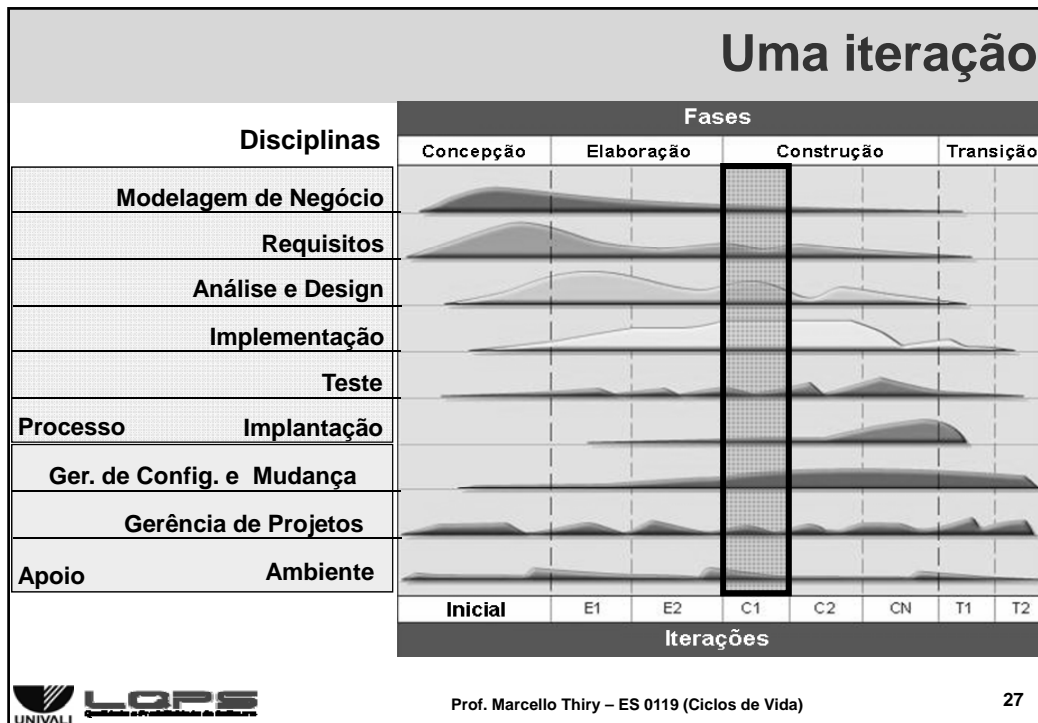
Iterações

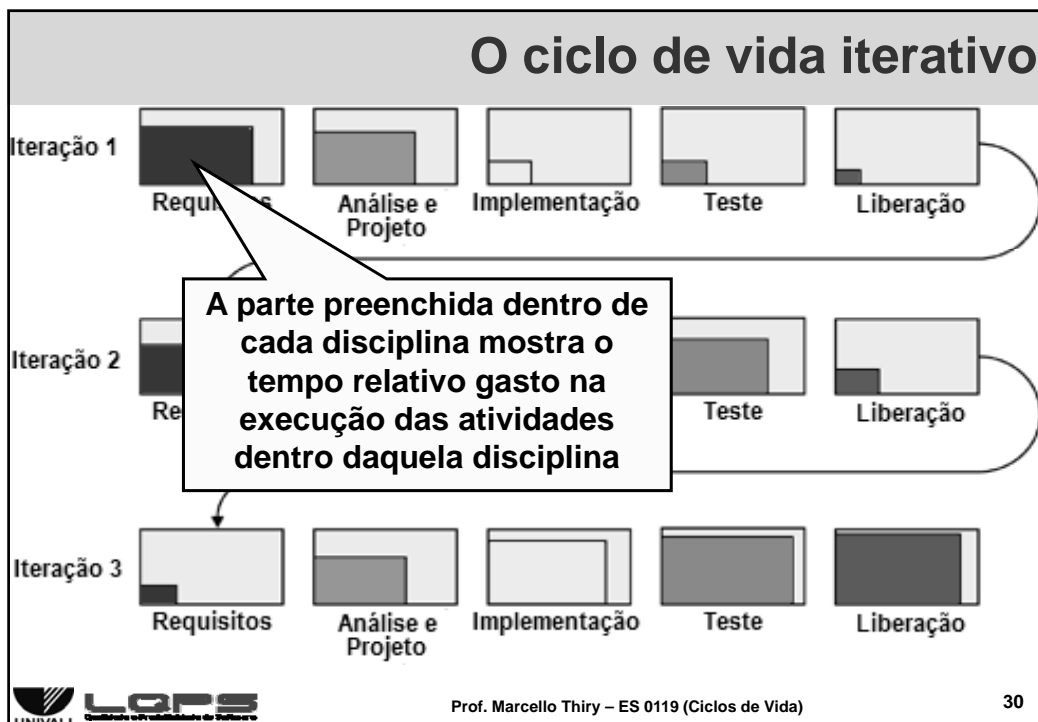
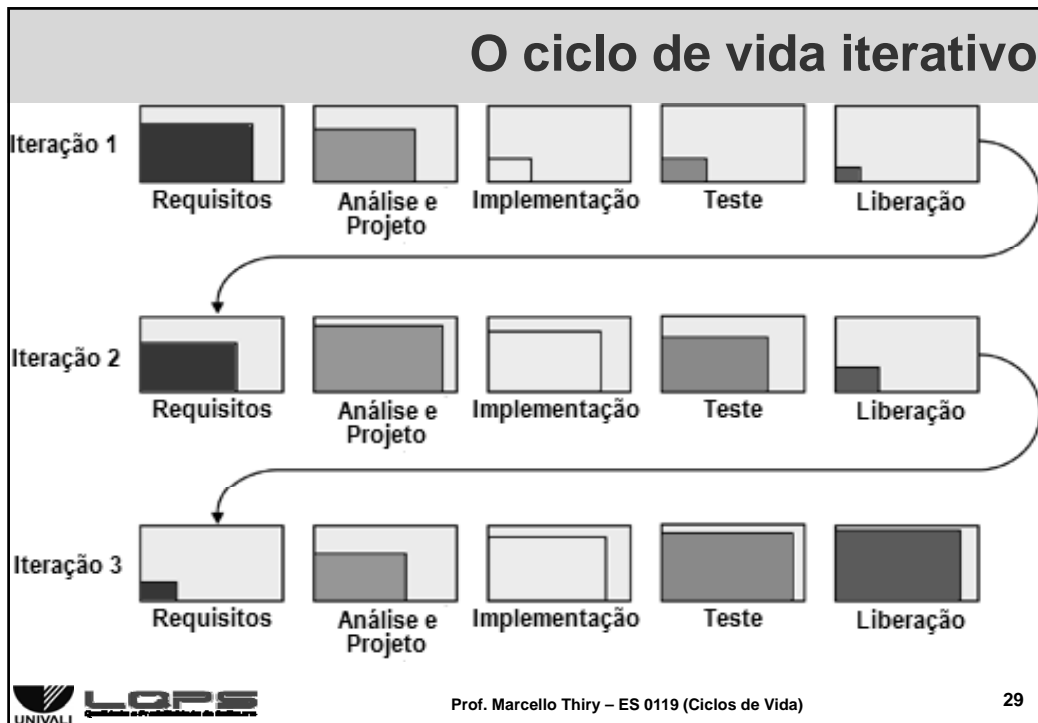
- ❑ Dentro de um ciclo de vida iterativo, vários passos são feitos através de cada disciplina. Cada um destes passos é chamado iteração
- ❑ Uma iteração é uma sequência distinta e com duração fixa de atividades que resulta em uma **liberação/release** (interna ou externa) de um produto executável

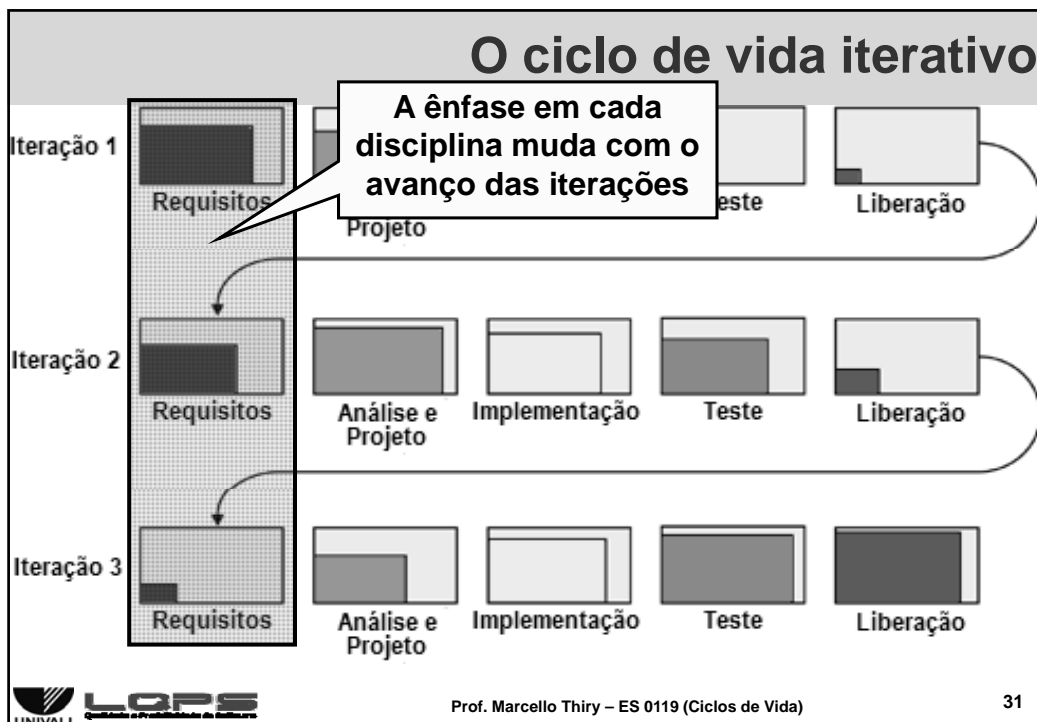
Um subconjunto de produtos finais que é o objeto de avaliação em um marco. Uma release é uma versão executável estável do produto, juntamente com quaisquer artefatos necessários para a utilização desta release (por exemplo, notas ou instruções de instalação). Uma release pode ser interna ou externa. Uma release interna é usada somente pela organização de desenvolvimento, como parte de um marco, ou para uma demonstração aos usuários/clientes. Uma release externa (ou liberação) é entregue aos usuários. Uma release não é necessariamente um produto completo, podendo ser um passo intermediário com utilidade limitada.

Builds e Releases

- ❑ O RUP define um build como uma versão operacional de um sistema ou parte de um sistema que demonstra um subconjunto de capacidades a serem fornecidas no produto final
- ❑ Quais as diferenças entre um build e uma release?
 - ❑ “Build” refere-se usualmente ao software que ainda está em teste, enquanto que “release” refere-se usualmente ao software que já foi testado (pode ser liberado para o usuário/cliente)
 - ❑ “Builds” são gerados com maior frequência
 - ❑ “Versões” são baseadas nos “builds” e não o contrário. Builds (ou uma série de builds) são gerados primeiro e uma release é baseada em um build (ou vários builds – o código acumulado de vários builds)







Uma iteração bem sucedida

- ❑ A iteração possui critérios claros de avaliação
- ❑ A iteração tem uma capacidade planejada que é demonstrável
- ❑ A iteração é concluída por um marco menor, onde o resultado obtido é avaliado em relação aos critérios de sucesso daquela iteração
- ❑ Durante a iteração, artefatos são atualizados (artefatos evoluem com o sistema)
- ❑ Durante a iteração, o sistema é integrado e testado


UNIVALI LQPS Laboratório de Qualidade e Produtividade de Software

Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

32

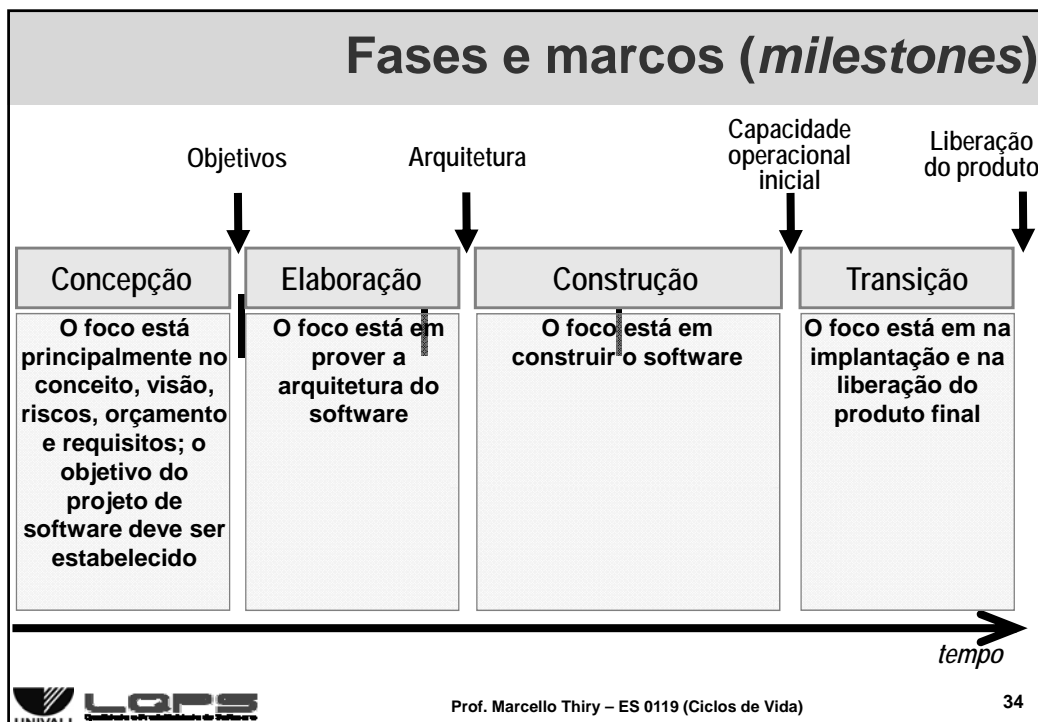
Fases

- ❑ As fases formam um framework onde as iterações são executadas e representam um plano estratégico para o projeto, orientando os objetivos para cada uma das iterações
- ❑ As fases fornecem marcos bem definidos que asseguram o andamento das iterações e a convergência para uma solução (evitar interações infinitamente)
- ❑ Os objetivos de cada fase são alcançados com a execução de uma ou mais iterações dentro de cada fase




Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)

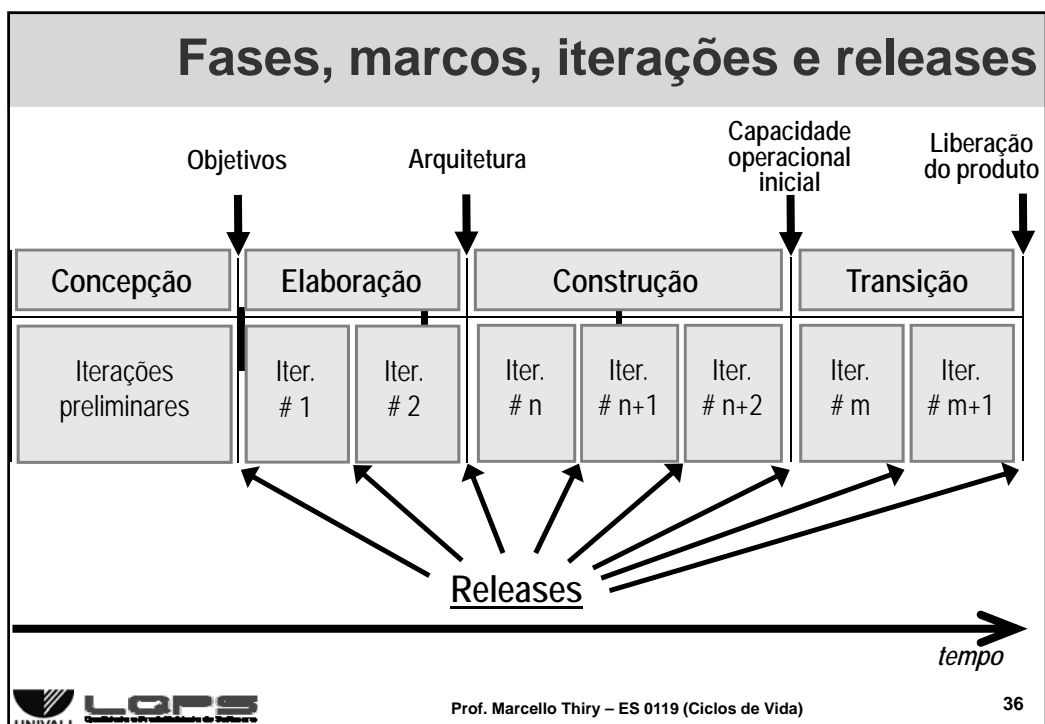
33

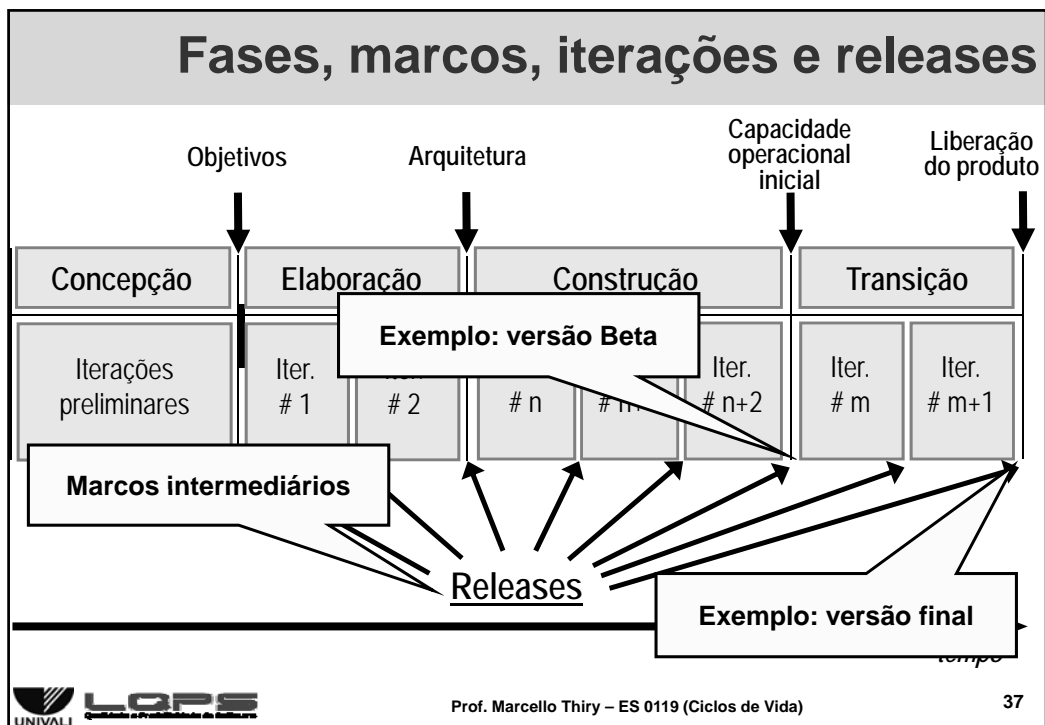


Fases

- ❑ Cada fase é concluída com um marco maior e uma avaliação para determinar se os objetivos da fase foram alcançados
- ❑ Uma avaliação satisfatória permite que o projeto possa se mover para a próxima fase
- ❑ Iterações são baseadas no tempo (elas ficam dentro de uma duração fixa), enquanto que as fases são baseadas nos objetivos
- ❑ Uma fase não pode ter uma duração fixa uma vez que a conclusão de uma fase é avaliada com base no estado do projeto

 Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 35





Concepção

- ☐ Marco: Objetivos do projeto (projeto é viável ou não)
- ☐ Disciplinas:


☐ Modelagem de Negócio
☐ Requisitos

 - ☐ Análise e Design
 - ☐ Implementação
 - ☐ Teste
 - ☐ Liberação (deploy)


LQPS UNIVALI Laboratório de Qualidade e Produtividade de Software

Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 38


Concepção: objetivos e critérios	
Objetivos primários	Critério de avaliação do marco
Estabelecer o escopo do projeto	Os stakeholders concordam sobre o escopo
Estabelecer os critérios de aceitação do projeto	Os stakeholders concordam sobre os critérios
Identificar as funcionalidades do sistema e selecionar aquelas que são críticas	Os stakeholders concordam que o conjunto correto de requisitos foi elicitado e que existe um entendimento comum sobre estes requisitos; todos os requisitos são priorizados
Estimar o custo e cronograma total do projeto (estimativas mais detalhadas para a fase seguinte: Elaboração)	Os stakeholders concordam que as estimativas de custo/cronograma, prioridades, riscos e processo são apropriadas
Estimar riscos em potencial	Todos os riscos foram registrados e avaliados, e uma estratégia de mitigação foi definida
Configurar o ambiente de suporte para o projeto (por ex: hardware, software, processo, recursos)	O ambiente de suporte está pronto


Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 39


Elaboração
<input type="checkbox"/> Marco: <u>Arquitetura do produto (arquitetura é provada)</u>
<input type="checkbox"/> Disciplinas: <ul style="list-style-type: none"> <input type="checkbox"/> Modelagem de Negócio <input type="checkbox"/> Requisitos <input type="checkbox"/> Análise e Design <input type="checkbox"/> Implementação <input type="checkbox"/> Teste <input type="checkbox"/> Liberação (deploy)
<input type="checkbox"/> Fornece a base estável para o projeto (design) e implementação


Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) 40

Elaboração: objetivos e critérios	
Objetivos primários	Critério de avaliação do marco
Assegurar que a arquitetura, requisitos e planos são estáveis; estabelecer uma arquitetura controlada por baselines	A visão do produto, requisitos e arquitetura são estáveis
Assegurar que os riscos são suficientemente mitigados para permitir prever custo/cronograma para conclusão do desenvolvimento	Os riscos mais significantes foram endereçados e estão sendo resolvidos adequadamente
Demonstram que a arquitetura suportará os requisitos do sistema dentro de custo e cronograma aceitáveis	Todos os aspectos arquiteturalmente significantes do sistema e determinadas funcionalidades estão sendo avaliadas em um protótipo evolucionário (protótipo que evolui para o sistema real)


Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) **41**

Construção
<p><input type="checkbox"/> Marco: <u>Capacidade operacional (toda funcionalidade foi desenvolvida)</u></p> <p><input type="checkbox"/> Disciplinas:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Modelagem de Negócio <input type="checkbox"/> Requisitos <input type="checkbox"/> Análise e Design <input checked="" type="checkbox"/> Implementação <input checked="" type="checkbox"/> Teste <input type="checkbox"/> Liberação (deploy) <p><input type="checkbox"/> O foco muda do entendimento do problema e da identificação dos elementos chave da solução para o desenvolvimento de um produto entregável</p>


Prof. Marcello Thiry – ES 0119 (Ciclos de Vida) **42**

Construção: objetivos e critérios

Objetivos primários	Critério de avaliação do marco
Alcançar versões úteis (alfa, beta e outras releases de teste) em intervalos de tempo	O sistema foi desenvolvido de acordo com as expectativas especificadas no plano da fase e nos planos das iterações
Completar a análise, design, implementação e teste de toda a funcionalidade requerida	Toda a funcionalidade requerida foi incorporada no sistema
Certificar que o sistema está pronto para ser colocado no ambiente do usuário final	O sistema atendeu todos os critérios de aceitação quando foi testado no ambiente de desenvolvimento

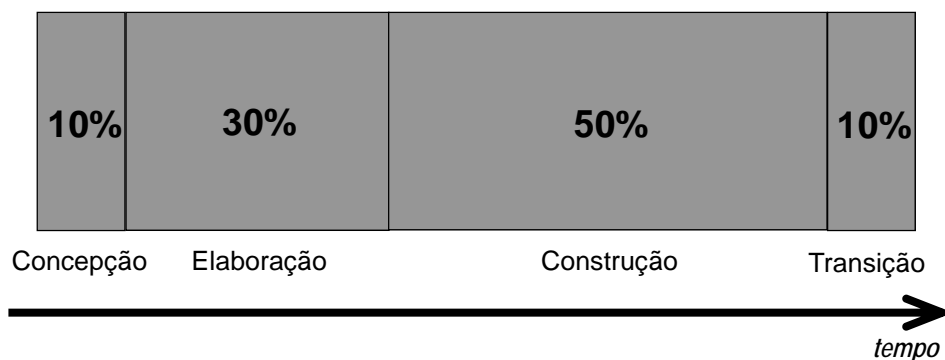
Transição

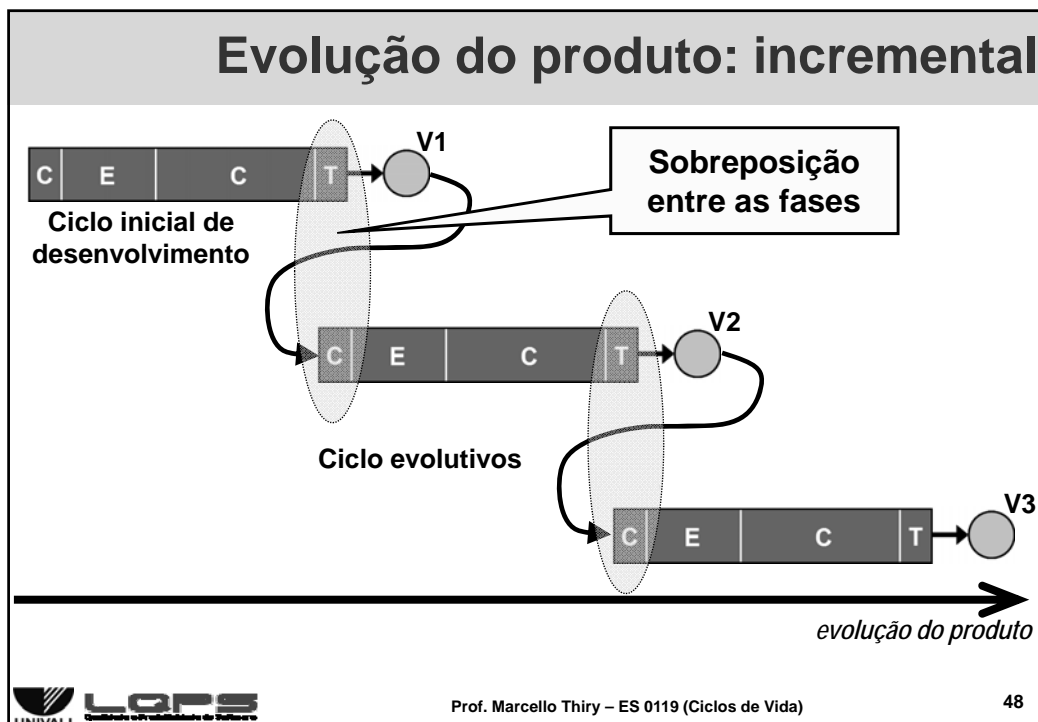
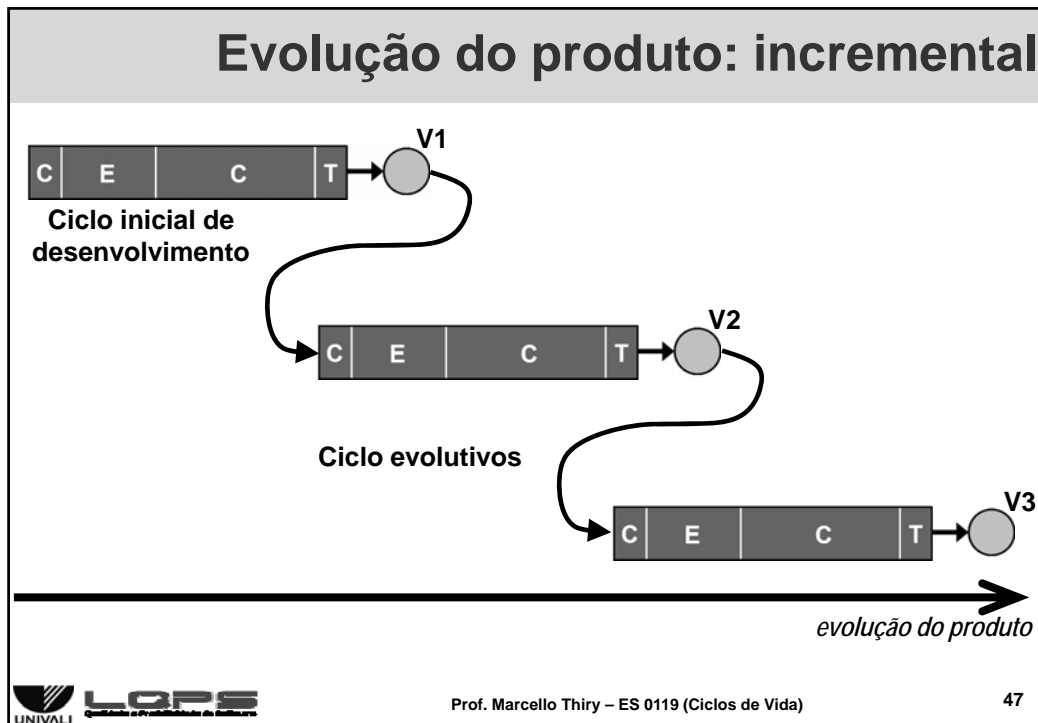
- ☐ Marco: Liberação do produto (produto foi liberado para entrar em produção)
- ☐ Disciplinas:
 - ☐ Modelagem de Negócio
 - ☐ Requisitos
 - ☐ Análise e Design
 - ☐ Implementação
 - ☐ Teste
 - ☐ Liberação (deploy)

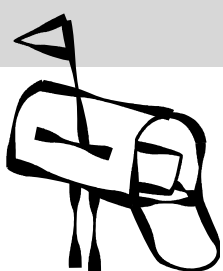

Transição: objetivos e critérios

Objetivos primários	Critério de avaliação do marco
Repassar o sistema para os canais apropriados de liberação	O sistema passou nos critérios formais de aceitação no ambiente dos usuários finais

Distribuição do tempo





Contato	
	<p>Marcello Thiry thiry@univali.br marcello.thiry@gmail.com</p> <p>LQPS http://www.univali.br/lqps</p>
	<p>Prof. Marcello Thiry – ES 0119 (Ciclos de Vida)</p> <p>49</p>