

Processo Unificado

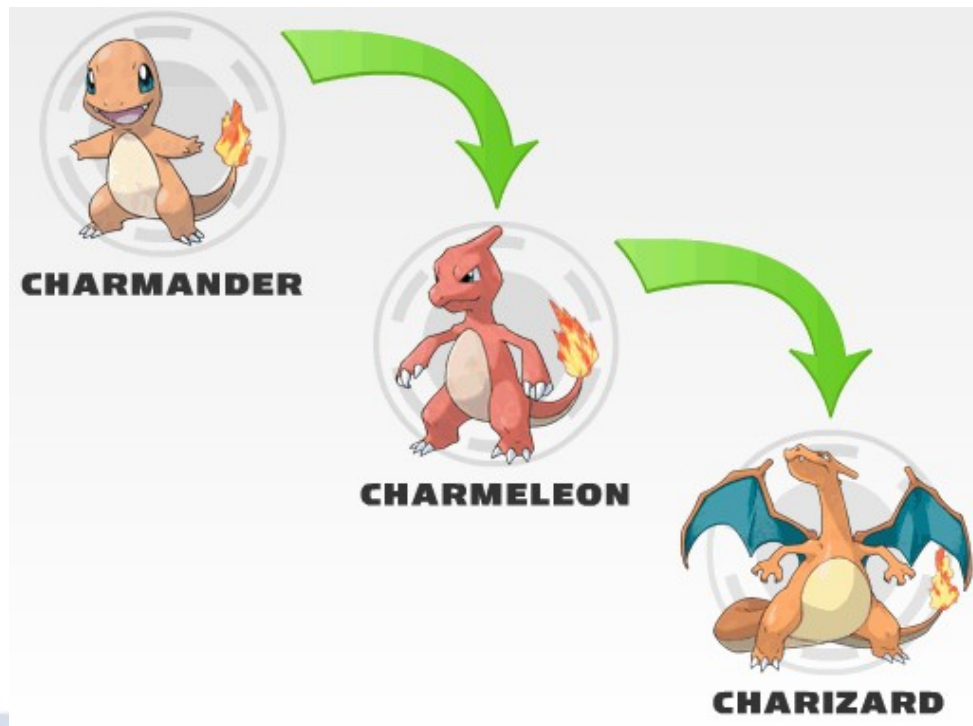
Prof. André Takeshi Endo

Até Agora...

- Modelos de processo
 - Cascata
 - Prototipação
 - RAD
 - Modelo em V
 - Desenvolvimento formal
- Modelos de processo evolutivos
- ***Unified Process (UP)***
- Métodos ágeis

Retomando: Modelos Evolutivos

- Natureza do software
 - “Software não se desgasta se deteriora”
- Software evolui !!!!



Retomando: Modelos Evolutivos

- Modelos de processo para
 - Requisitos mutantes
 - Entregas em períodos mais curtos
 - Software que evolui!!
- Modelos evolutivos são ***iterativos***
- Possibilita a entrega de ***versões*** cada vez mais complexas

Processo Unificado

- Não é um modelo de processo
- “Processo ***iterativo*** popular para o desenvolvimento de software visando à construção de sistemas ***orientados a objetos***” [Larman]
- O PU é muito flexível e aberto
 - Embora tenha sido associado a processo pesados “*heavyweight*” com alto foco em documentação (UML)

Processo Unificado

- Histórico
 - 1960 – Jacobson
 - 1987 – Jacobson cria a Objectory AB
 - 1995 – Rational compra a Objectory, aperfeiçoa o Objectory e cria o ROP (Rational Objectory Process)
 - Jacobson, Rumbaugh e Booch (UML)
 - 1998 – Evolui para o RUP
- *Rational Unified Process (RUP)*
 - Refinamento detalhado do PU

Processo Unificado

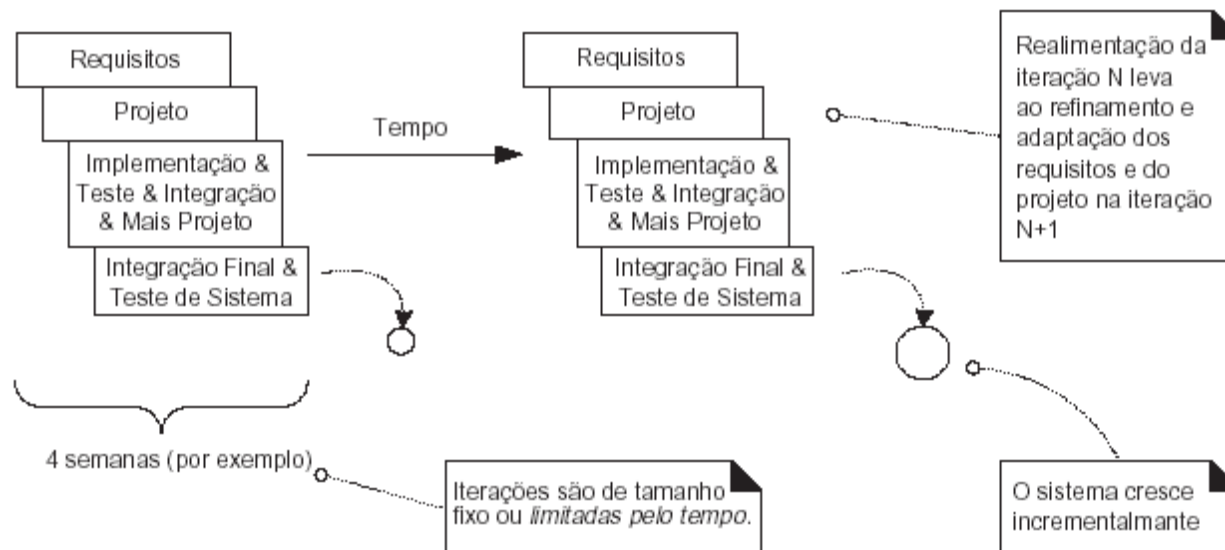
- Baseado no modelo incremental
- Usa a UML como linguagem de apoio
- É um framework que pode ser personalizado
 - Necessidades específicas
 - Recursos disponíveis

Processo Unificado

- Princípios básicos
 - Desenvolvimento iterativo
 - Baseado em Casos de uso
 - Centrado em arquitetura

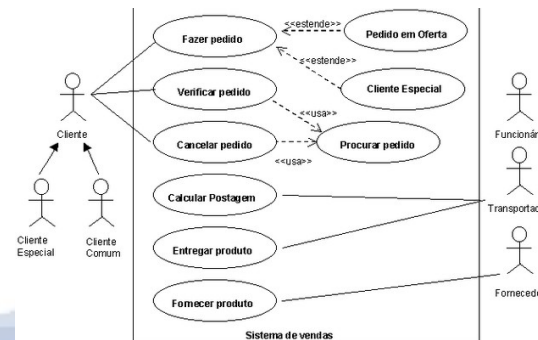
Processo Unificado

- **Desenvolvimento iterativo**
- Ciclos iterativos → incremento
- A duração de cada ciclo pode variar
 - 4 semanas, 3 meses



Processo Unificado

- Baseado em casos de uso
 - O que caso de uso?
- Os casos de uso guiam o desenvolvimento
 - Elicitação e negociação dos requisitos
 - Testes de aceitação
- Planejamento
 - Iterações

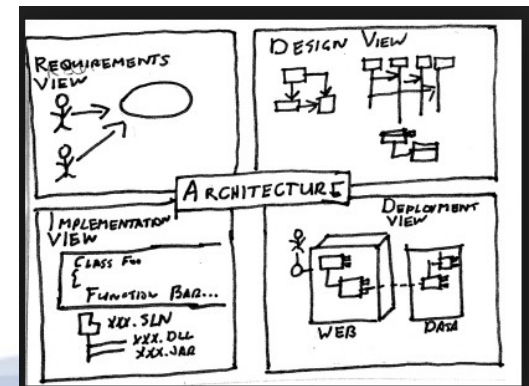


Use Case 1 Buy Stocks over the Web

Primary Actor: Purchaser
Scope: Personal Advisors / Finance package (PAF)
Level: User goal
Stakeholders and Interests:
Purchaser—wants to buy stocks and get them added to the PAF automatically.
Stock agency—wants full purchase information.
Precondition: User already has PAF open.
Minimal Guarantee: Sufficient logging information will exist so that if something went wrong and ask the user to provide details.
Success Guarantee: Remote web site has acknowledged the purchase and the user's portfolio are updated.
Main Success Scenario:
1. Purchaser selects to buy stocks over the web.
2. PAF gets name of web site to use (E*Trade, Schwab, etc.) from user.
3. PAF opens web connection to the site, retaining control.
4. Purchaser browses and buys stock from the web site.
5. PAF intercepts responses from the web site and updates the purchase.
6. PAF shows the user the new portfolio standing.
Extensions:
2a. Purchaser wants a web site PAF does not support:
2a1. System gets new suggestion from purchaser, with option to continue.
3a. Web failure of any sort during setup:
3a1. System reports failure to purchaser with advice, backs up to 2a2. Purchaser either backs out of this use case or tries again.

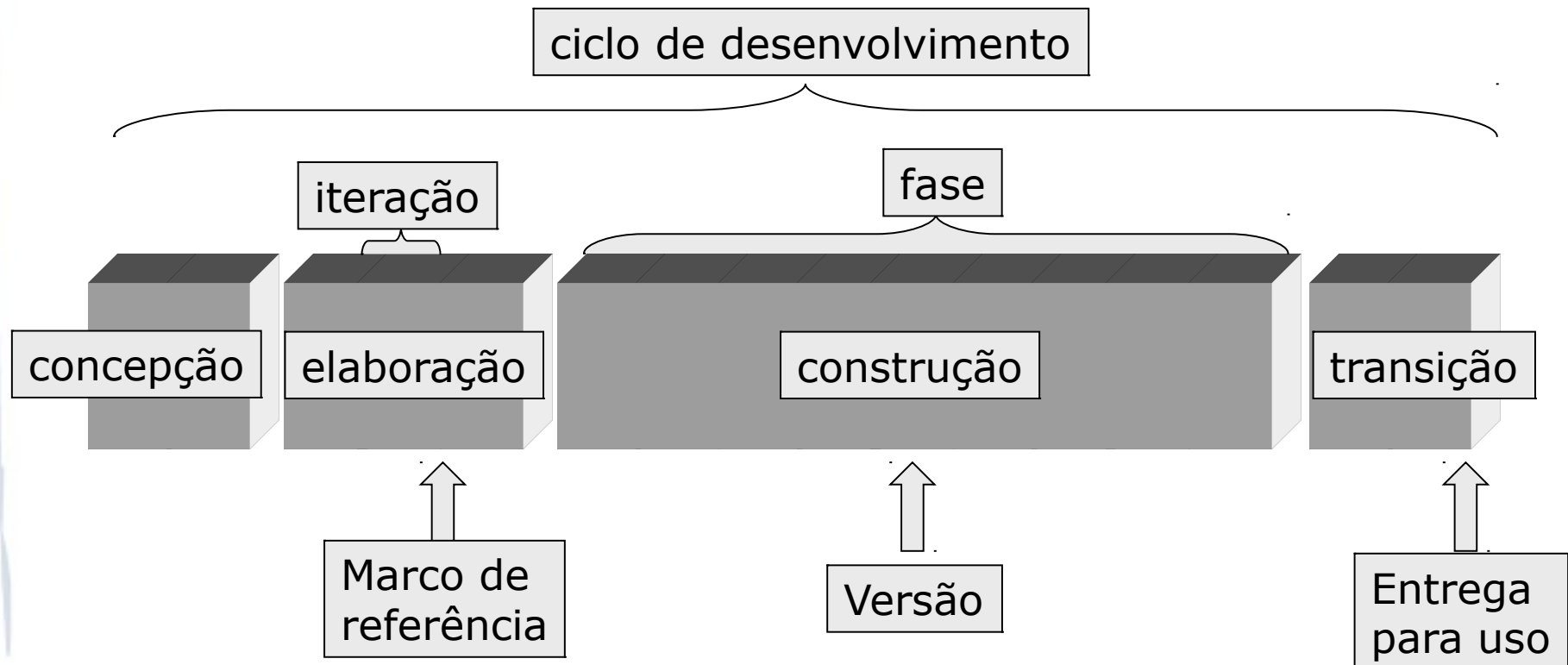
Processo Unificado

- **Centrado em arquitetura**
- “Arquitetura é a organização fundamental do sistema como um todo. Inclui elementos estáticos, dinâmicos, o modo como trabalham juntos e o estilo arquitetural que guia a organização do sistema.”
- Alicerce do sistema
- Arquitetura + casos de uso → explorar o sistema



PU - Fases

- O processo é dividido em várias iterações
- Cada fase agrupa iterações



PU - Fases

- **Concepção**
- Viabilidade do projeto
- Definição do *escopo do sistema*
- Estimativas vagas
- Riscos
- Esboço da *arquitetura*

PU - Fases

- **Elaboração**
- Definição de *requisitos funcionais* (maioria)
- Resolução dos altos riscos
- Detalhamento da arquitetura
- Estimativas mais realistas

PU - Fases

- **Construção**
- O sistema é efetivamente desenvolvido
- Incrementos
- Acontece o desenvolvimento iterativo e incremental
- Preparação para a implantação

PU - Fases

- **Transição**
- Testes Beta
- Entrega do software ao cliente
- Incrementos são implantados
- Correção de defeitos

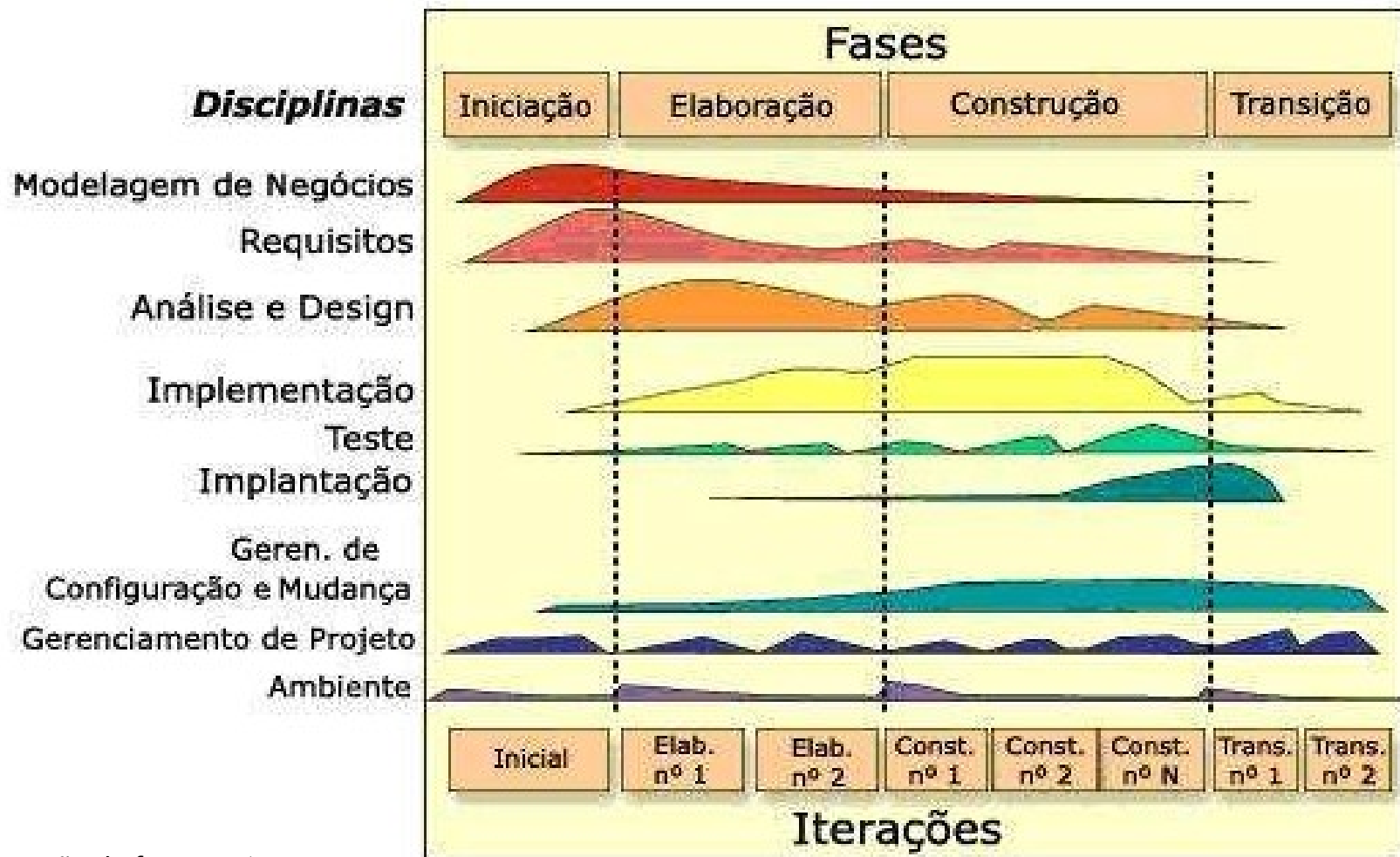
PU - Disciplinas

- As fases não caracterizam uma cascata
- Em cada fase (iteração), as disciplinas estão acontecendo
 - Análise de requisitos, projeto, testes, etc
- Disciplinas são ortogonais a todas as fases
 - A fase define a ênfase de cada disciplina
- As disciplinas no PU definem **quem** (papel), **como** (atividades) e **o que** (artefato) deve ser feito em cada iteração.



PU - Disciplinas

- Representação



PU - Disciplinas

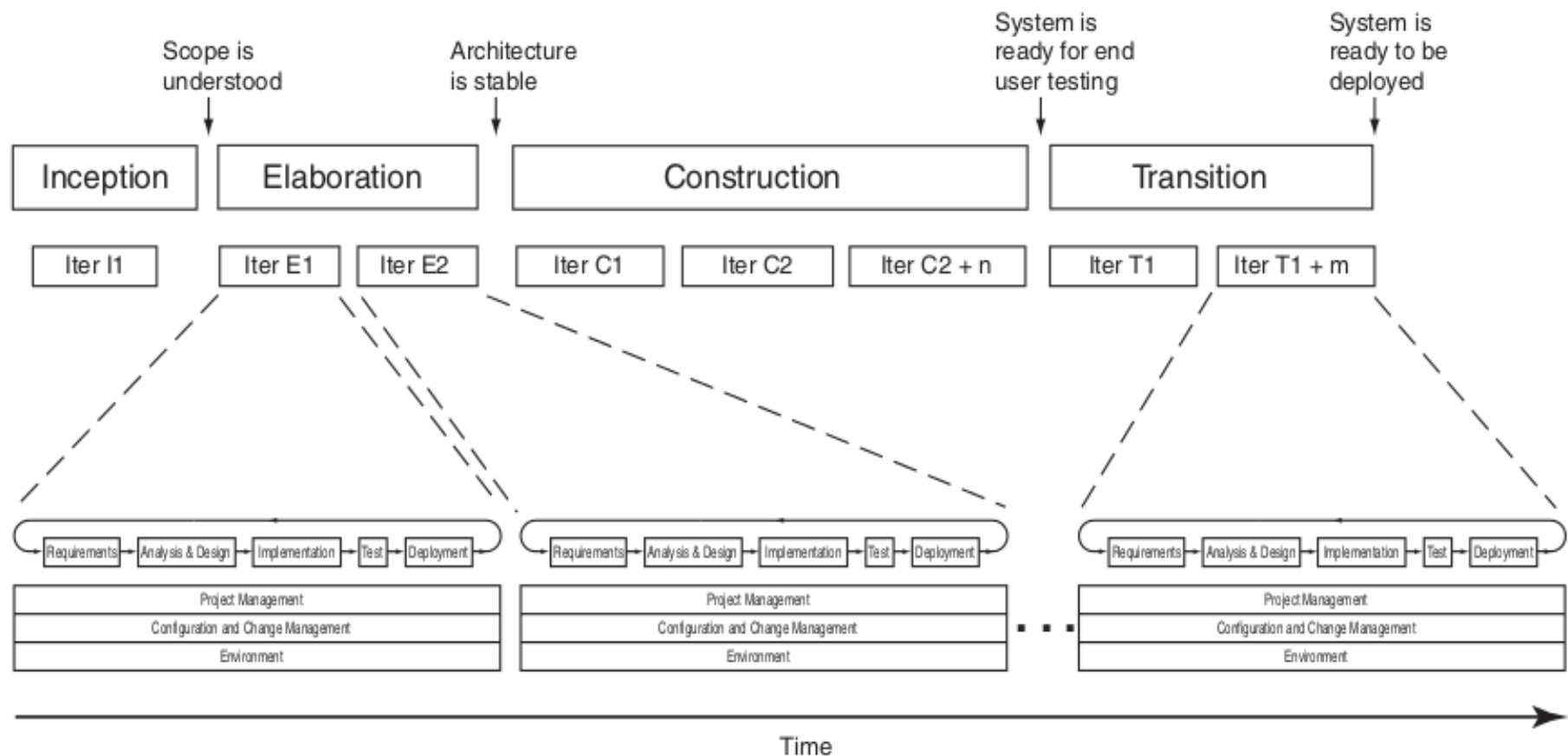
- Exemplo de artefatos produzidos

| Disciplina | Artefato <i>Iteração</i> → | Concepção C_1 | Elaboração $E_1...E_n$ | Construção $C_1...C_n$ | Transição $T_1...T_n$ |
|----------------------|--------------------------------------|--------------------|---------------------------|---------------------------|--------------------------|
| Modelagem de Negócio | Modelo Conceitual | | p | | |
| Requisitos | Diagrama de Casos de Uso | p | r | | |
| | Casos de Uso Textuais | p | r | | |
| | Diagrama de Seqüência do Sistema | p | r | | |
| | Contratos para operações | p | r | | |
| | Glossário | p | r | | |
| Projeto | Diagrama de Classes | | p | r | |
| | Diagrama de Colaboração | | p | r | |
| | Diagrama de Pacotes | | p | r | |
| | Documento de Arquitetura do Software | | p | | |
| Implementação | Código fonte | | | p | r |

p-iniciar; r-refinar

PU - Disciplinas

- Representação (revisando... [Booch])



Processo Unificado (-)

- Pode tornar-se “*heavyweight*” se adotado de forma incorreta
- Casos de uso não definem todos os requisitos
- Complexo

Processo Unificado (+)

- Alterações facilmente gerenciáveis
- Integração de módulos desenvolvimentos com os já existentes
- Análise de pontos críticos e planejamento de estratégias a cada iteração
- Tudo é opcional
 - Framework para instanciar seu processo

Bibliografia

- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

Bibliografia

- [UUU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP, DC-UFSCAR e UTFPR-CP.