

# FIA/P GRADUAÇÃO

**DISCIPLINA: COMPLIANCE & QUALITY ASSURANCE**  
**PROJETO DE SISTEMAS APLICADO AS MELHORES PRÁTICAS EM**  
**QUALIDADE DE SOFTWARE E GOVERNANÇA DE TI**

**AULA:**

**6- Processo de Software**

**PROFESSOR:**

**RENATO JARDIM PARDUCCI**


**PROFRENATO.PARDUCCI@FIAP.COM.BR**

## **AGENDA DA AULA**

- ✓ CMMi nível 2 de maturidade - PP,
- ✓ MS.br nível GPR
- ✓ CVS e Processo SW
- ✓ Escolha de um modelo de processo de software - Clássico x Ágil (CVS e processo)





VAMOS REFORÇAR OS CONCEITOS SOBRE GUIAS E NORMAS DA QUALIDADE COM UM JOGO!



**Guias e Normas da Qualidade**

by RenatoJP 9 seconds ago

 Quiz 12 questions  Private

## DESENVOLVIMENTO DA CAPACIDADE E MATURIDADE NA QUALIDADE DE SOFTWARE

Vamos iniciar a jornada de aprendizado sobre como desenvolver práticas que possibilitem evoluir os níveis de Qualidade de uma empresa na produção de Software.

A partir de agora, serão percorridos os níveis do CMMi, com demonstração das principais técnicas e ferramentas que permitem cumprir com as determinações do Guia da Qualidade e alcançar certificações de reconhecimento do mérito.

**CMMi nível 2 de maturidade - PP,  
MS.br nível GPR**

**Escolha de um Processo de  
Software**

## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Existem **metodologias de condução de projetos** de software que, seguindo um Ciclo de Vida (Cascata, Espiral, Incremental ou Prototipação), **explicam processos, tarefas, artefatos e responsabilidades** na execução de trabalhos de planejamento execução e controle da produção.

## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Os métodos podem seguir uma linha Clássica ou Ágil.

Na **linha Clássica ou Preditiva**, existe uma **preocupação grande com a formalização** de atividades e as etapas do projeto são bem definidas e **pouco flexíveis**. Em geral são **empregadas em situações de projetos com escopo fechado** ou situações de implantação de software que envolvem praticamente parametrizações (configurações) de instalação, **com baixa probabilidade de mudança** em escopo, prazo e custo inicialmente planejados.

Na **linha Ágil**, o **foco está na geração de um produto útil com muita rapidez**. Os controles sobre o projeto são simplificados para que o foco não se perca. **Aceita-se mudanças ao longo do projeto**, as quais envolverão **renegociação e replanejamento rápido**. **O escopo não precisa estar completamente confirmado para iniciar**.



## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Métodos **Clássicos** em geral **seguem** os modelos de Ciclo de Vida **Cascata ou Incremental**.

Métodos **Ágeis** geralmente **seguem** o **Ciclo de Vida Espiral ou Prototipação**.

## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

O método Clássico mais difundido é o RUP.



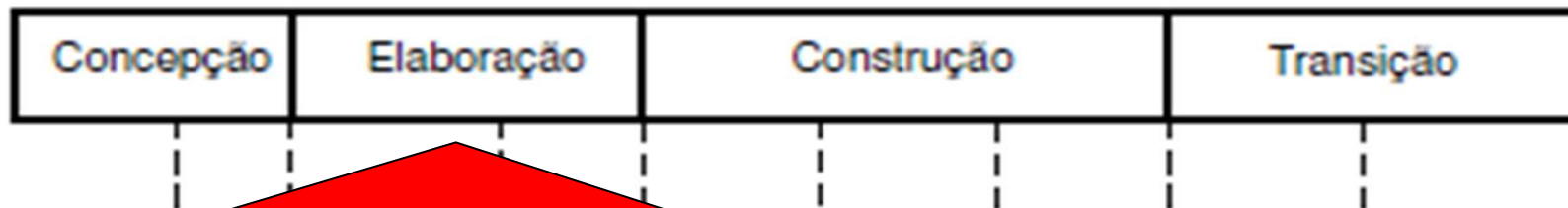




## Visão geral do RUP (Rational Unified Process)

### Fases

- Cada fase é subdividida em iterações.

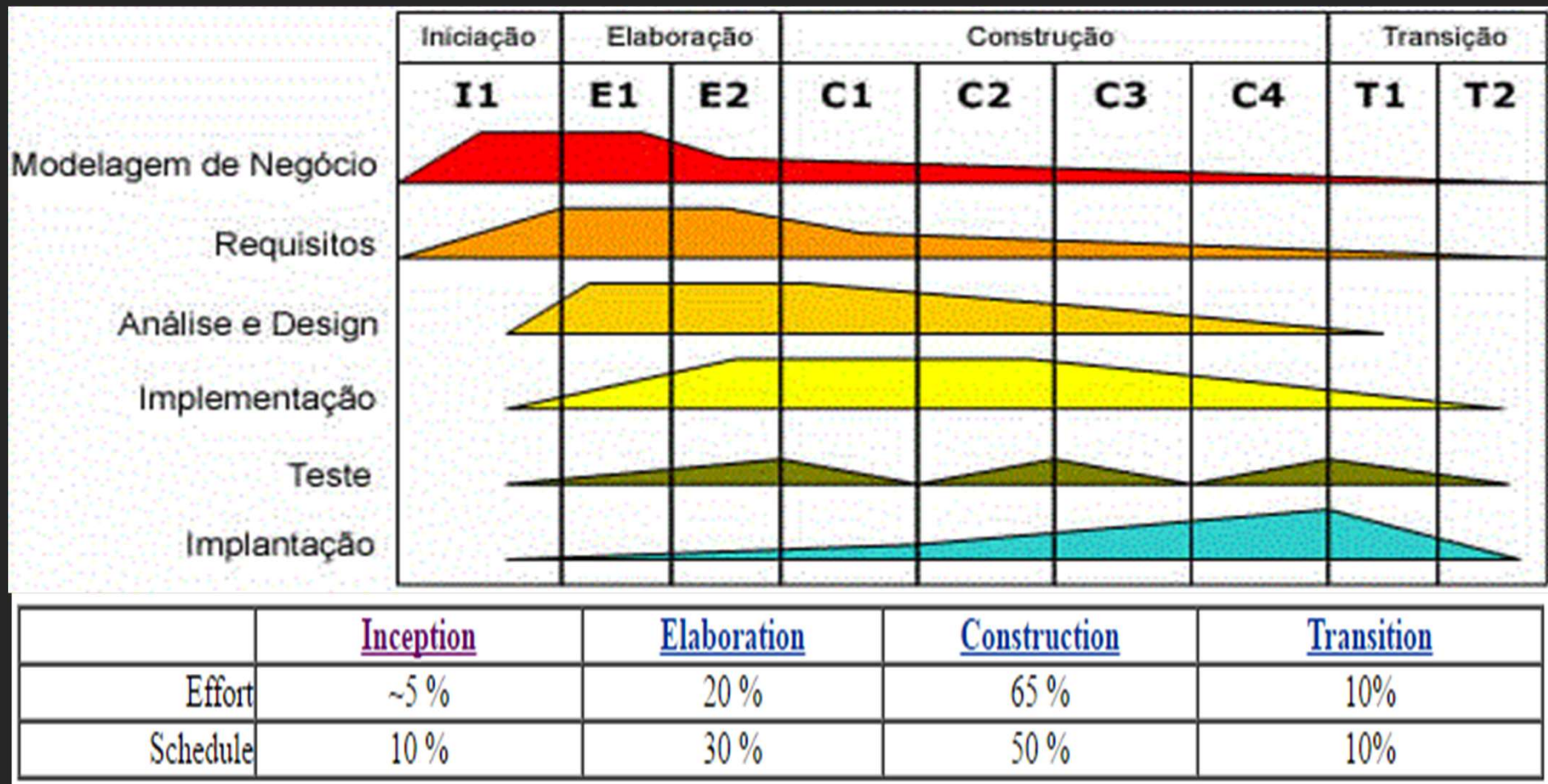


**Para cada fase, são definidos os “Workers”:**

- Papel (exemplo – analista de sistemas, programador, usuário final, analista de testes, etc.);
- Responsabilidades em termos de atividades que deve cumprir e produtos (artefatos) que deve gerar (código, caso de teste, plano de projeto, etc.).

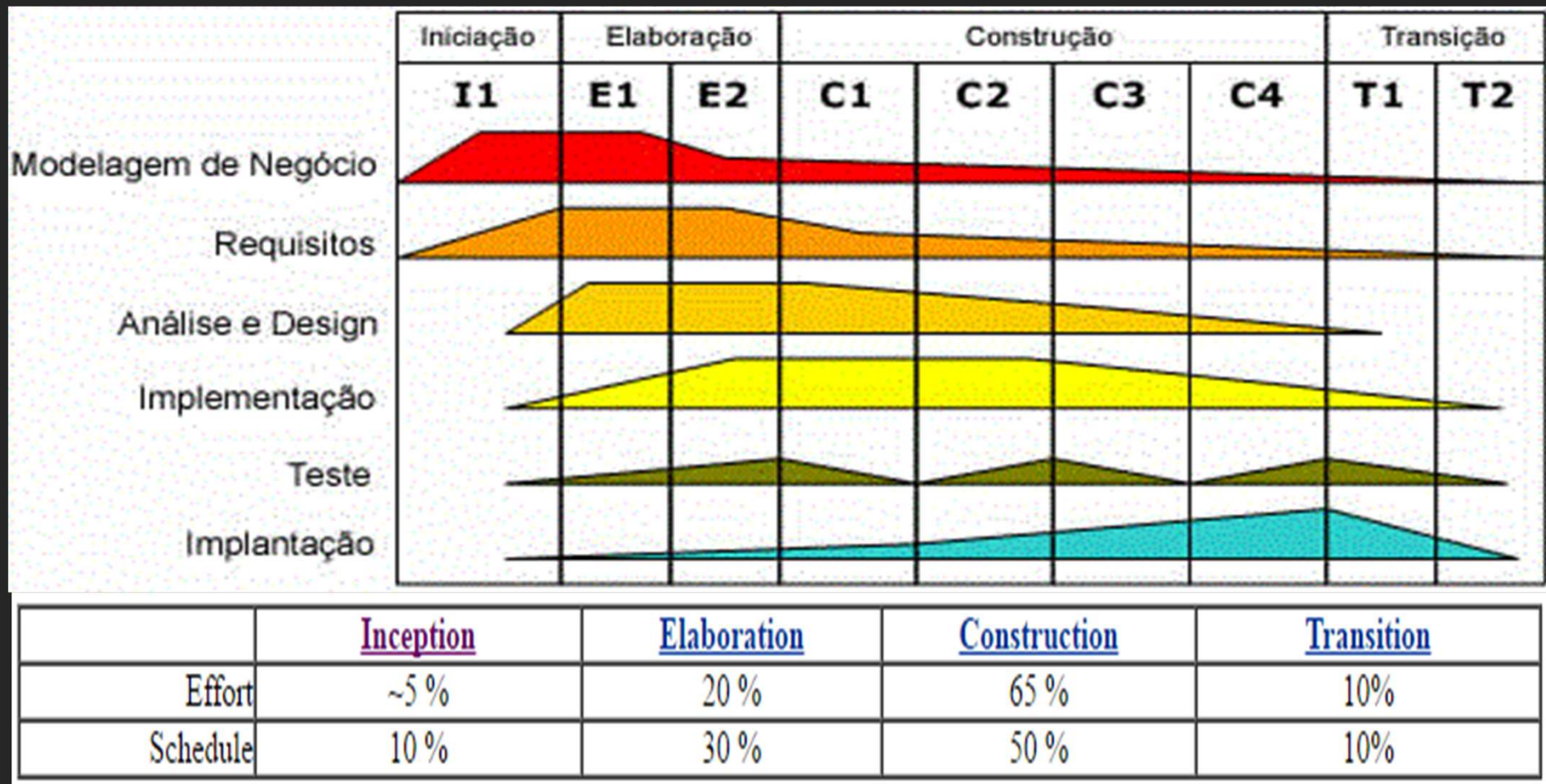


O RUP define uma previsão de distribuição de esforço para realizar os diversos tipos de atividades envolvidas no desenvolvimento do software, ao longo das fases do ciclo de vida (popularmente chamado de gráfico de baleias).



Maiores informações sobre o RUP:

<http://sce.uhcl.edu/helm/rationalunifiedprocess/Ruphomepage.com>



## RUP

Define uma previsão de distribuição de esforço para realizar os diversos tipos de atividades envolvidas no desenvolvimento do software, ao longo das fases do ciclo de vida.

	<u>Inception</u>	<u>Elaboration</u>	<u>Construction</u>	<u>Transition</u>
Effort	5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%



O RUP ainda apresenta descrições detalhadas de cada atividade a ser desenvolvida num projeto, com exemplificação de estruturas padrão (templates) de artefatos a serem produzidos a cada atividade realizada.

Ele traz bastante formalismo ao processo de desenvolvimento.

## DEMONSTRAÇÃO DO RUP

<http://sce.uhcl.edu/helm/rationalunifiedprocess/>

[Ruphomepage.com](http://Ruphomepage.com)

O processo de software é usado como base para o planejamento e controle do ciclo de vida de produção de software!

Consulte o arquivo ProjetoReferencia-EngSW-RUP.mpp na sua área de apostilas!

Ele demonstra um plano de projeto com base na abordagem RUP

## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Métodos Ágeis mais difundidos:

**RAD MODEL**

**AGILE  
METHODOLOGY**



## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

**AGILE**  
METHODOLOGY

Métodos Ágeis mais difundidos:

1º padrão  
internacional  
de processo  
ágil

Tornou-se um  
padrão para o  
mercado de Sw



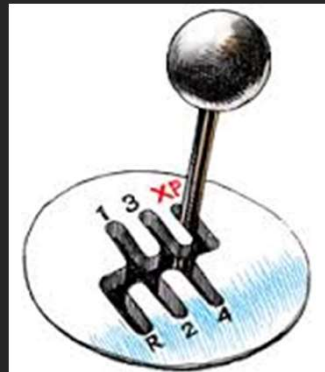
## MÉTODO ÁGIL

Visão geral da  
Metodologia Ágil

**AGILE**  
METHODODOLOGY

## PROJETO ÁGIL

Os projetos conduzidos sob uma **metodologia ágil** focam na **geração de valor para o cliente**, ou seja, procuram dedicar **maior atenção, tempo e capital na produção daquilo que o cliente vai usufruir**, sem dispendar recursos demasiados em controles acessórios.



## MANIFESTO ÁGIL

### **“Manifesto para o Desenvolvimento Ágil de Software”**

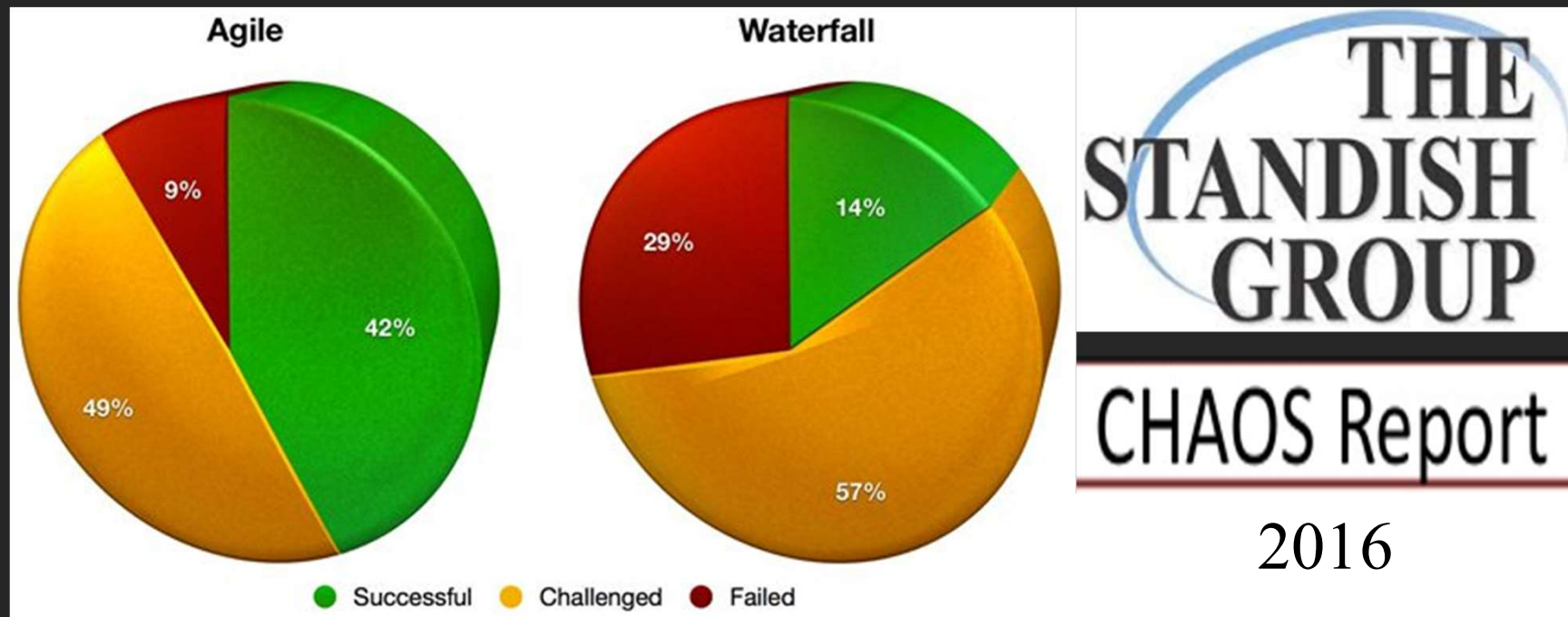
- **Indivíduos e interações** acima de processos e ferramentas.
- **Software funcionando** acima de documentação abrangente.
- **Colaboração** com o cliente acima de negociação de contratos.
- **Responder a mudanças** acima de seguir um plano.



## MÉTODOS ÁGEIS

Atualmente, os métodos ágeis ganham grande espaço no mercado por:

- Atenderem rapidamente as expectativas dos clientes;
- Focarem equipes de desenvolvimento naquilo que gostam de fazer
- Aumentarem as taxas de sucesso quanto a qualidade do projeto em termos de processo e produto



## MÉTODO ÁGIL

Visão geral do Extreme  
Programming



## Valores do XP

**Comunicação:** XP foca em construir um entendimento pessoa-a-pessoa do problema, com o uso mínimo de documentação formal e com o uso máximo de interação “cara -a-cara” entre as pessoas envolvidas no projeto. As práticas de XP como programação em pares, testes e comunicação com o cliente têm o objetivo de estimular a comunicação entre gerentes, programadores e clientes.

**Simplicidade:** XP sugere que cada membro da equipe adote a solução mais fácil que possa funcionar. O objetivo é fazer aquilo que é mais simples hoje e criar um ambiente em que o custo de mudanças no futuro seja baixo.

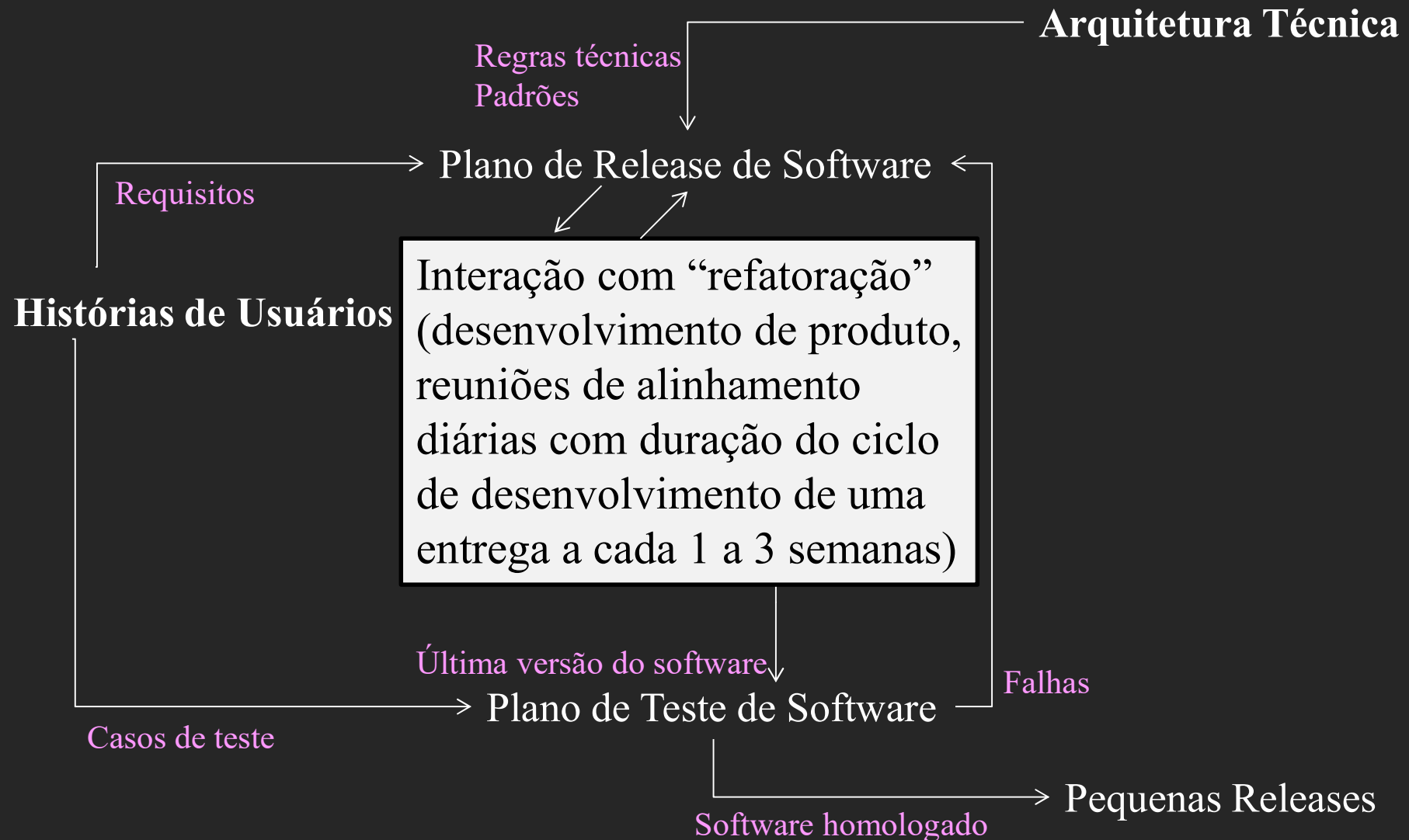
**Feedback:** Os programadores obtêm feedback sobre a lógica dos programas escrevendo e executando casos de teste. Os clientes obtêm feedback através dos testes funcionais criados para todas as histórias (casos de uso simplificados).

**Coragem:** Ela é necessária para que realmente se aplique XP como deve ser aplicado. Exemplos de atitude que exigem coragem são: alterar código já escrito e que está funcionando; jogar código fora e reescrever tudo de novo; e permitir código compartilhado por todos.

## FASES DO CICLO DE VIDA DO SOFTWARE COM XP

- A fase de **exploração** é anterior à construção do sistema. Nela, investigações de possíveis soluções são feitas e verifica-se a viabilidade de tais soluções.
- A fase de **planejamento inicial** deve ser usada para que os clientes concordem em uma data para lançamento do primeiro release. O planejamento funciona da seguinte forma: Os programadores, juntamente com o cliente, definem as histórias (use case simplificados) a serem implementadas e as descrevem em cartões. Os programadores assinalam uma certa dificuldade para cada história e, baseados na sua velocidade de implementação, dizem quantas histórias podem implementar em uma iteração. Depois, os clientes escolhem as histórias de maior valor para serem implementadas na iteração – isso é chamado planejamento de iteração.
- Na fase das **iterações do release** são escritos os casos de teste funcionais e de unidade. Os programadores vão seguindo mais ou menos o seguinte fluxo de atividades na seguinte ordem (Em cada iteração): escrita dos casos de testes; projeto e refatoramento; codificação; realização dos testes; e integração.
- A fase de **manutenção**.
- A fase de **morte**.

# INTEGRAÇÃO DAS FASES DO DESENVOLVIMENTO COM XP

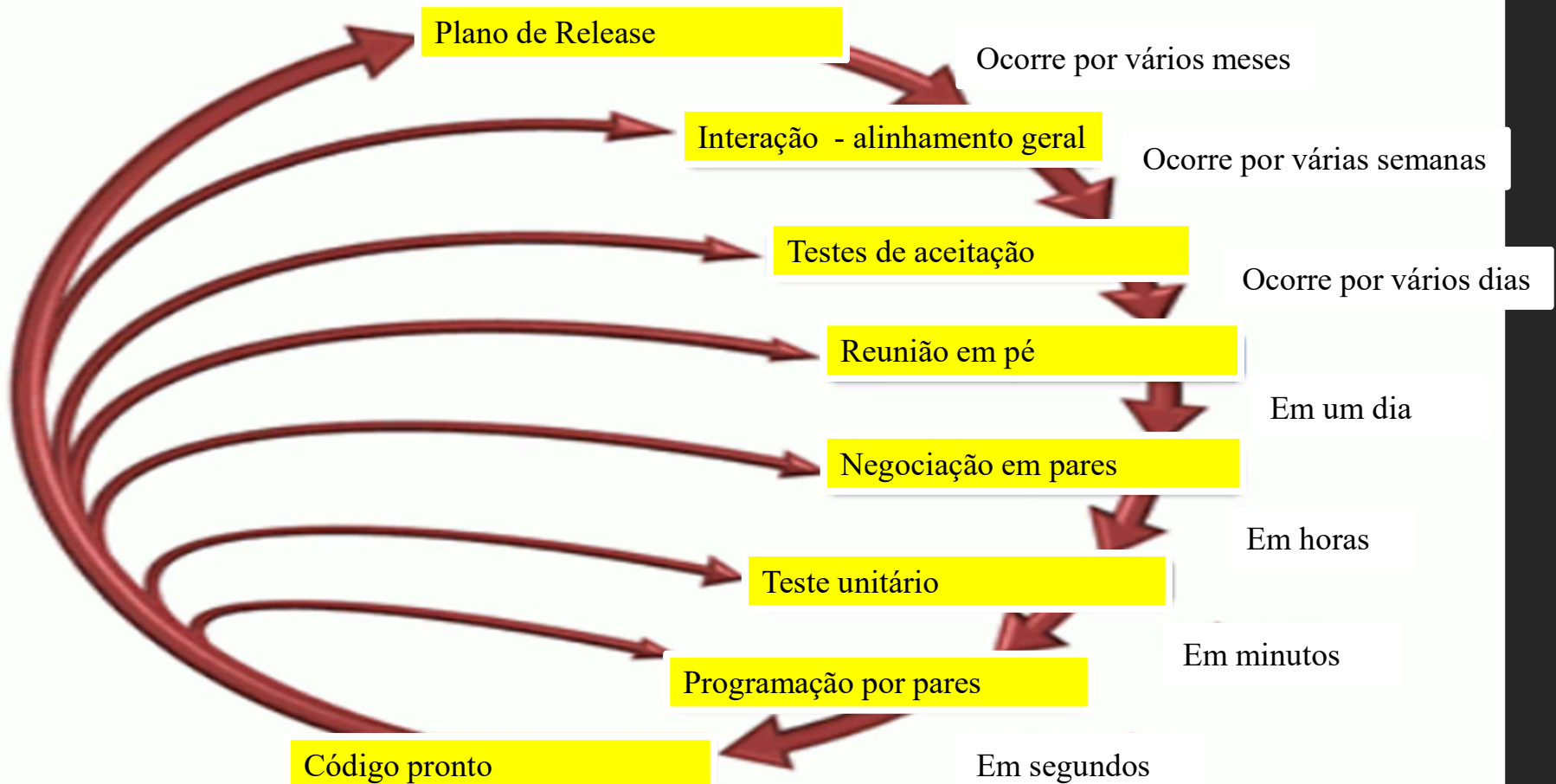


## EQUIPES DE TRABALHO NO XP

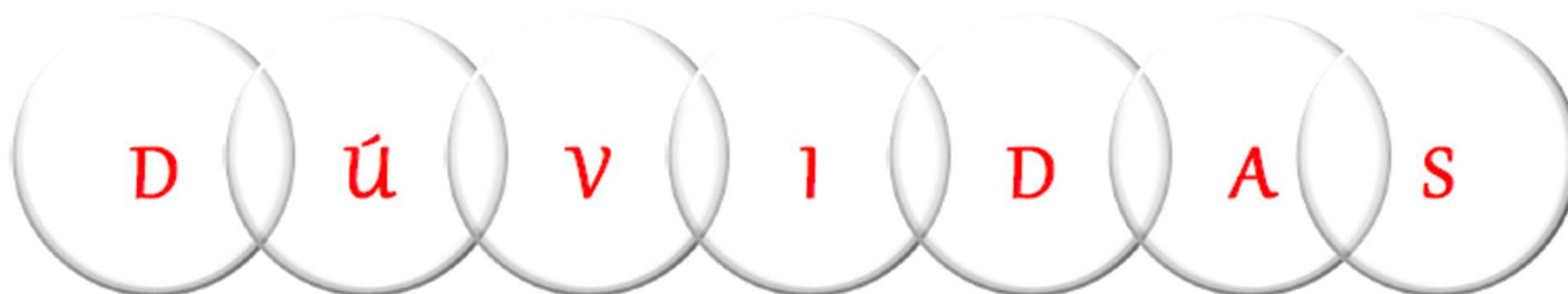
- **treinador** se preocupa principalmente com a execução técnica e evolução do processo segundo o XP (zela pela metodologia).
- **rastreador** (tracker) coleta métricas sobre o que está sendo desenvolvido e confronta com as métricas estimadas verificando possíveis divergências
- **programador** ocupa o principal papel em XP. Ele analisa, projeta, testa, codifica, e integra o sistema. Além disso, o programador estima a dificuldade das histórias e faz alterações nessas estimativas, caso necessário.
- **cliente** escolhe o que vai agregar valor ao seu negócio, escolhe o que deve ser feito primeiro e o que deve ser adiado. Além disso, o cliente define com a ajuda dos testadores, os testes funcionais que irão mostrar se o sistema realmente faz o que deve ser feito.
- **testador** irá ajudar o cliente na definição e escrita dos testes funcionais. Ele não precisa ser uma pessoa com apenas essa função, **pode desempenhar também o papel de programador.**
- **consultor** é necessário apenas em algumas situações onde se precisa de um especialista em uma determinada tecnologia ou sobre determinado assunto.

## VISÃO GERAL DO PROCESSO DE SOFTWARE COM XP

### Processo de Software em LOOPS

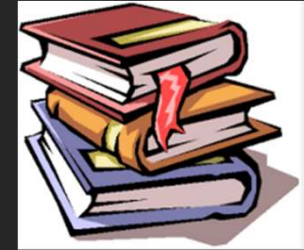








## Referência bibliográficas



### BIBLIOGRAFIA :

- SURJAN, Jakov Trofo - Stelutti, Danilo - Fonseca, Douglas Henrique da - Oliveira, José Fabio Saraiva de. Sistema de gestão de projetos baseado na metodologia ágil scrum , 2011.
- **PRESSMAN**, Roger S.. Engenharia de software. - Uma abordagem profissional, 7ª edição. São Paulo, AMGH.
- **SOMMERVILLE**. Engenharia de software. Pearson.

Material de aula estará no site após a aula.

BONS ESTUDOS!

## Bibliografia

- Guide to the Scrum Body of Knowledge (SBOK™ Guide), 2013.
- KAWAMOTO Júnior, Luiz Teruo - Gimenes Jr., Carlos - Pinho, Luciane Cristina Magalhães, Análise das boas práticas de gerenciamento de projetos no processo de desenvolvimento de software utilizando metodologia PMI e SCRUM em uma empresa multinacional, 20'0.
- SURJAN, Jakov Trofo - Stelutti, Danilo - Fonseca, Douglas Henrique da - Oliveira, José Fabio Saraiva de. Sistema de gestão de projetos baseado na metodologia ágil scrum , 2011.
- VARGAS, Ricardo Viana. Gerenciamento de projetos: estabelecendo diferenciais competitivos. 7. ed Rio de Janeiro Brasport, 2010.