PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE

Processo de Desenvolvimento de Software

Aula 03

Prof. Carlos Eduardo de B. Paes
carlosp@pucsp.br

Agenda Aula 03

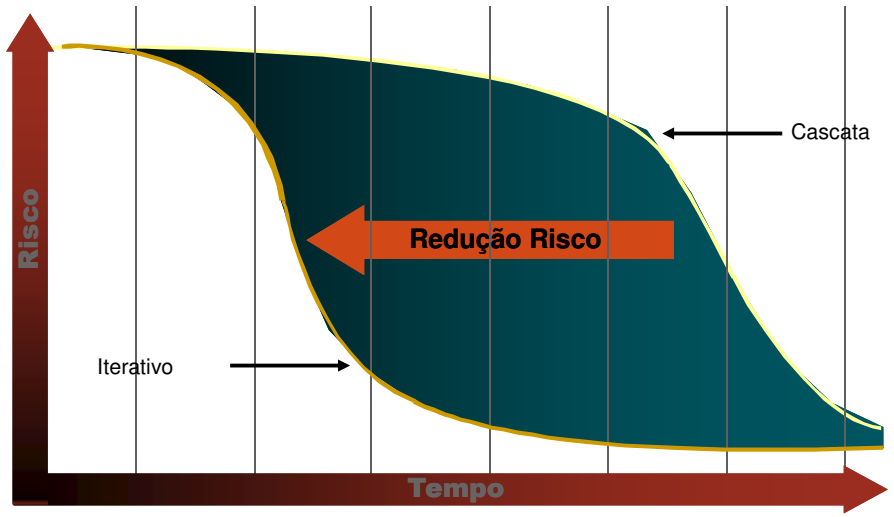
- As Filosofias do RUP
- Comparando processo de desenvolvimento de SW
- Estudo de caso: Projeto Deimos

As Filosofias do RUP

1. Atacar os maiores riscos o mais cedo possível e continuamente, ou eles irão atacar você
2. Assegurar que está sendo entregue algo de valor para seu cliente
3. Ficar focado em um software executável
4. Acomodar mudanças o mais cedo possível no projeto
5. Definir uma linha base de uma arquitetura executável o mais cedo possível
6. Construir o seu sistema usando componentes
7. Trabalhar em grupo como um time
8. Fazer a qualidade como um estilo de vida, não um preocupação postergada

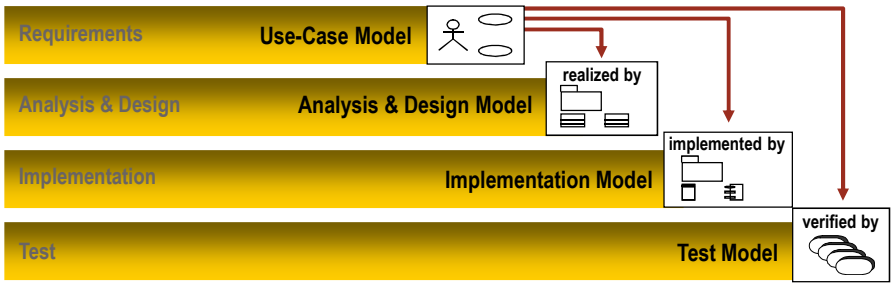
Filosofia 1

Atacar os maiores riscos o mais cedo possível e continuamente, ou eles irão atacar você



Filosofia 2

Assegurar que está sendo entregue algo de valor para seu cliente



Filosofia 3

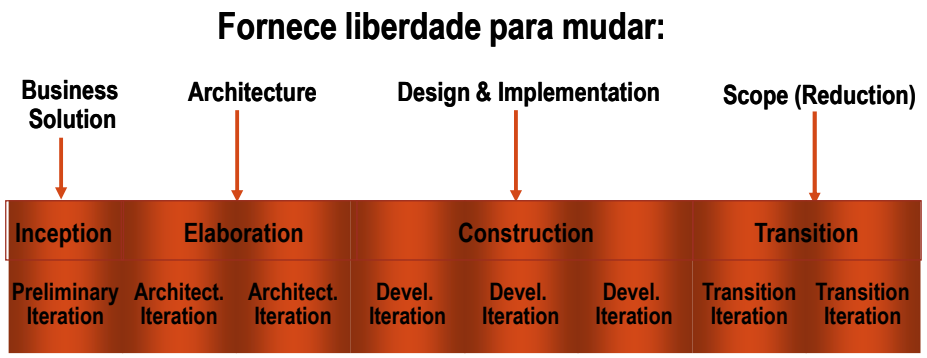
Ficar focado em um software executável

- Medir o progresso primariamente através da revisão do código executável e os resultados dos teste
  - Planos, requisitos, designs e outro produtos normalmente fornecem um falsa percepção do progresso e status
- Focar no produto final a ser entregue e nos artefatos que importam para que este objetivo seja atingido consistentemente
  - Agilize o processo
  - **Não use todo o RUP!**
    Apenas use que faz sentido para o seu projeto

Filosofia 4

Acomodar mudanças o mais cedo possível no projeto

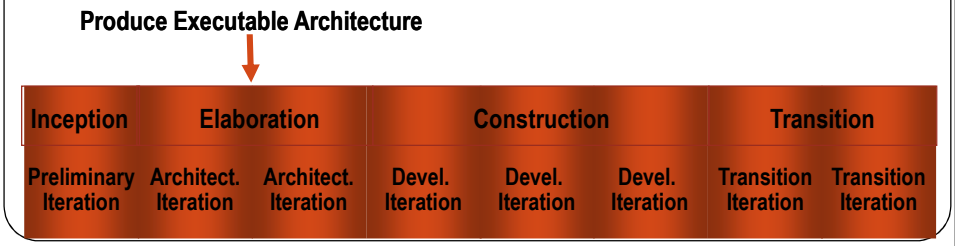
- Os sistemas de hoje são tão complexos para obter os requisitos, arquitetura, design e o escopo certo na primeira vez



Filosofia 5

Definir uma linha base de uma arquitetura executável o mais cedo possível

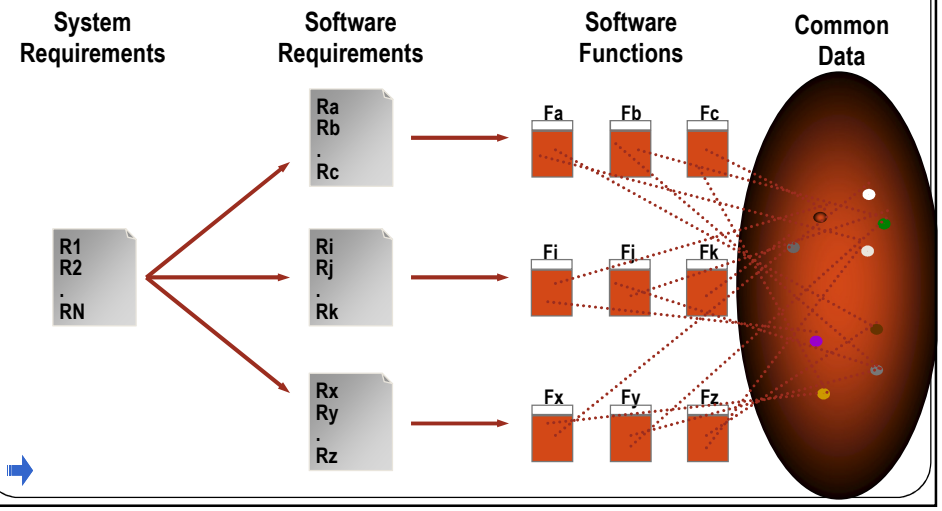
- Arquitetura fornece um estrutura reduzida do sistema
  - Subsistemas, componentes chaves,interface e mecanismos arquiteturais (soluções para problemas comuns, tais como persistência, comunicação entre processos, etc..)
- Implementar e testar a arquitetura mitiga as maioria dos riscos técnicos



Filosofia 6

Construir o seu sistema usando componentes

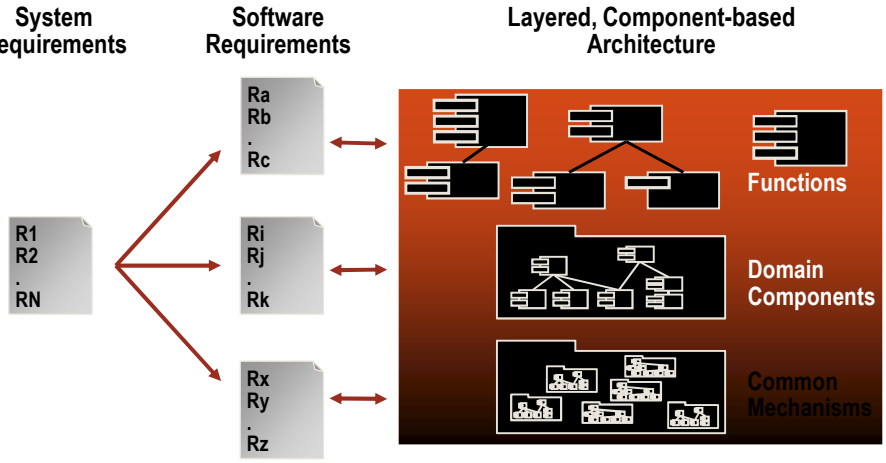
- Decomposição Funcional Tradicional
  - Dirigida por requisitos
  - Muitas dependências cria sistema não-flexíveis



Filosofia 6

Construir o seu sistema usando componentes

Arquitetura componentizadas fornece flexibilidade



Filosofia 7

Trabalhar em grupo como um time

- Capacitar e auto-gerenciamento
  - Visão clara
- Responsável pelos resultados do time
  - Expectativas claras
  - Todos por um, e um por todos (EVITAR)
  - "Meu projeto foi ótimo, seu código não funciona"
- Otimizar a comunicação
  - "Face-to-face" ao invés de e-mail
  - Processo efetivo (tamanho certo para o seu projeto)
  - Organizar em torno da arquitetura, não em torno das funções
  - Obter as ferramentas certas de suporte (fácil de acessar requisitos, áreas de trabalho privadas, fácil de acessar defeitos e etc)

Filosofia 8

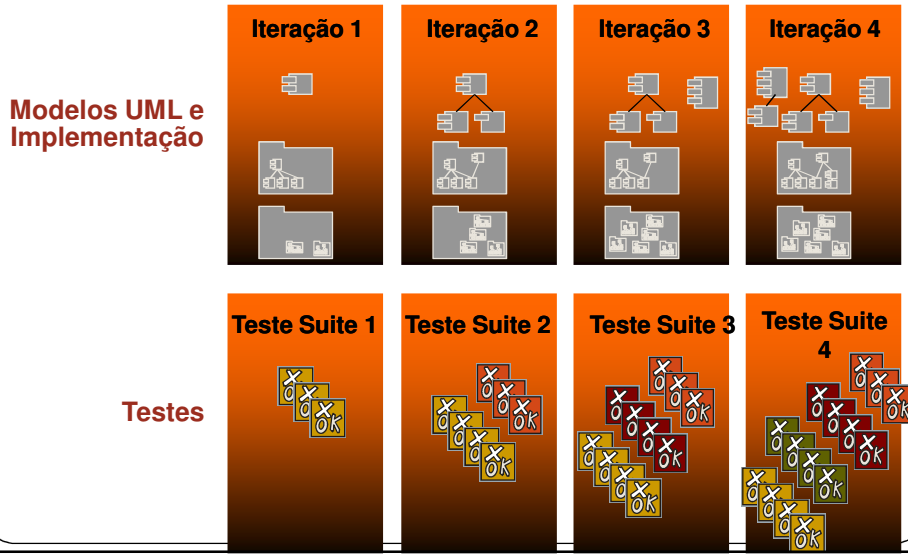
Fazer a qualidade como um estilo de vida, não um preocupação postergada

Problemas com software são de 100 a 1000 vezes mais caro para encontrar e reparar depois do desenvolvimento

- **Custo para reparar o software**
- **Custo de perda de oportunidades**
- **Custo de perda de clientes**

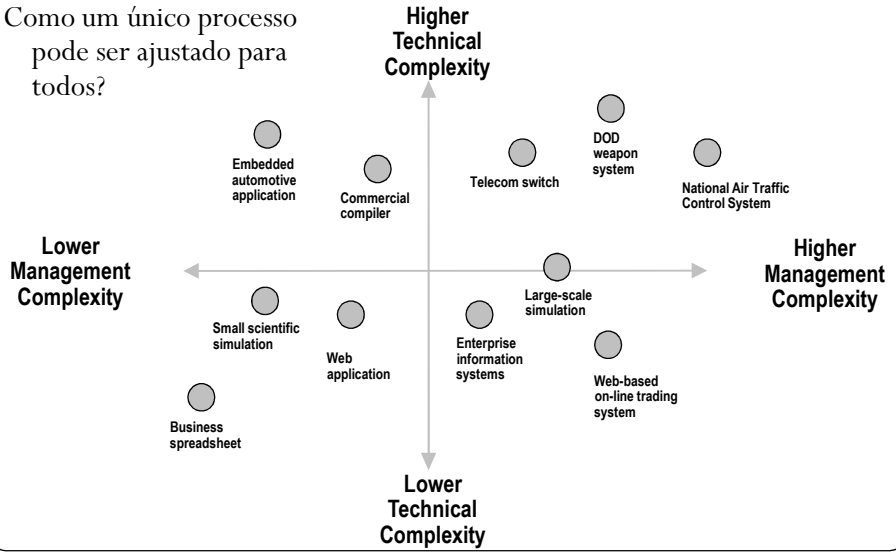
Filosofia 8

Fazer a qualidade como um estilo de vida, não um preocupação postergada



Comparando Processos de Software

Como um único processo pode ser ajustado para todos?



Comparando Processos de Software

- Objetivos:
  - Compreender melhor como aplicar o RUP
  - Discutir características e diferenças entre o RUP e outras abordagens
- Outras abordagens
  - **Abordagens ágeis**: Extreme Programming (XP),Scrum, Dynamic Systems Development Method(DSDM), Crystal e Adaptative Development

Comparando Processos de Software

- Outras abordagens
  - **Frameworks de avaliação de processo:** Software Engineering Institute (SEI) Capability Maturity Models (SEI CMM e SEI CMMI) e ISO/IEC 15504
  - **Desenvolvimento heavyweight**: DOD-STD-2167 e MIL-STD-498

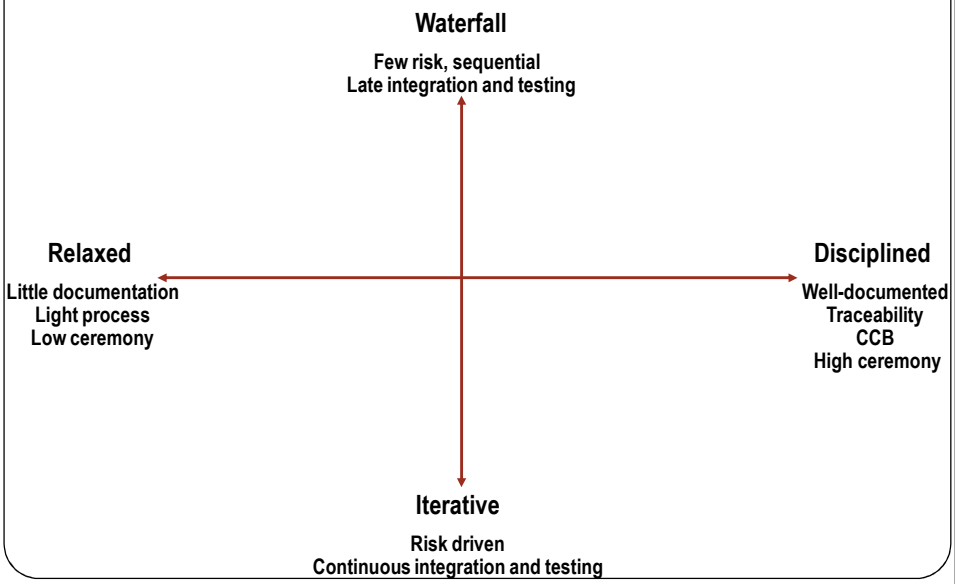
Comparando Processos de Software
Como comparar processos

- Podemos caracterizá-los da seguinte forma:
  - **Baixa/Alta cerimônia**
    - **Baixa cerimônia**: produz um mínimo de documentação de suporte e tem pouco formalismo nos procedimentos de trabalho
    - **Alta cerimônia**: tem uma ampla documentação de suporte e mantem rastreabilidade entre artefatos, controle de mudanças e etc..

Comparando Processos de Software
Como comparar processos

- Podemos caracterizá-los da seguinte forma:
  - **Cascata ou Iterativo (ciclo de desenvolvimento):**
    - **Cascata**: abordagem linear com integração e testes realizados tardiamente
    - **Iterativo:** abordagem de desenvolvimento dirigida por riscos com implementação, integração e testes prematuros

Comparando Processos de Software
Como comparar processos



Comparando Processos de Software
Processos Ágeis

- Se tornou muito popular, especialmente entre desenvolvedores individuais e times pequenos
- Alguns processo ágeis:
  - **XP** (Kent Beck, 2000)
  - **Scrum** (Schwaber, 2002)
  - **DSDM** (Stapleton, 1998)
  - **Cristal** (Cockburn, 2002)
  - **Adaptative Development** (Highsmith, 2000)

Comparando Processos de Software
Processos Ágeis

- Desenvolvimento ágil faz alguns sacrifícios em termos da cerimônia e rigor
- Tem como retorno uma maior flexibilidade e habilidade para adaptação em ambientes em que o negócio evolui constantemente e muito rápido
- Sugiram na década de 90 (XP,Scrum,Crystal e Adaptative Development)
- Focado em pessoas (problema!!)

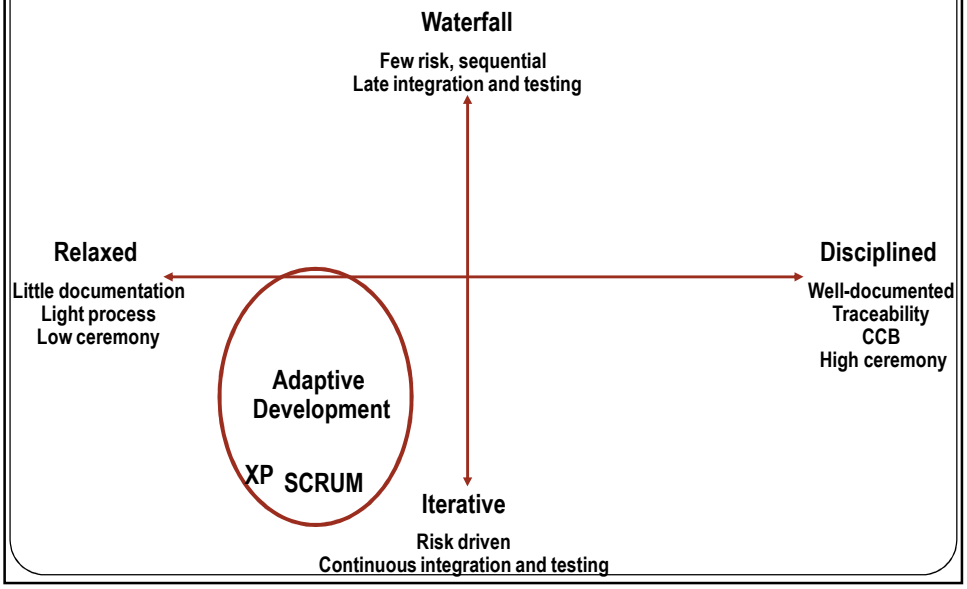
Comparando Processos de Software
Processos Ágeis

- Tem com base as boas práticas de engenharia de software (desenvolvimento iterativo, integração contínua, foco no software executável, refatoração, padrões de codificação e *users stories*
- Contribuições importantes:
  - **Promoção e popularização das boas práticas**
  - **Aceitação do conceito de processo**
  - Programação em pares
  - Redução da cerimônia
  - Foco no software e não nos artefatos de suporte

Comparando Processos de Software
Processos Ágeis

- Aspectos desfavoráveis:
  - Abordagens com pouca maturidade (alguns anos de idade)
  - Novas e não provadas
  - Não fornecem ainda muita orientação de uso (???)
  - Com usar esses processo em projetos grande e complexo → precisamos de orientação para ter sucesso no projeto

Comparando Processos de Software
Processos Ágeis



Comparando Processos de Software
Alta Cerimônia

- Em paralelo com os movimento ágil temos uma forte tendência as abordagens de alta-cerimônia para desenvolvimento de software
- Essa tendência inclui:
  - Frameworks para avaliação de processo: CMM, CMMI e ISO/IEC
  - Processos: DOD-STD e MIL-STD

Comparando Processos de Software
Alta Cerimônia

- Atualmente temos:
  - Empresas investindo em software como diferencial competitivo
  - Software se tornou um investimento estratégico de negócio
  - Os projetos de software não podem ser imprevisíveis e estourarem o orçamento
  - Problemas de qualidade!!!

Comparando Processos de Software
Alta Cerimônia

- Conseqüências:
  - Muitas organizações tem se preocupado com a importância de usar um bem-definido e documentado processo de desenvolvimento de software para facilitar o sucesso de projeto de software
- Características de projeto de alta cerimônia
  - Planos completos são cuidadosamente documentados
  - Muitos artefatos relacionados a gerenciamento, requisitos, design e testes são produzidos

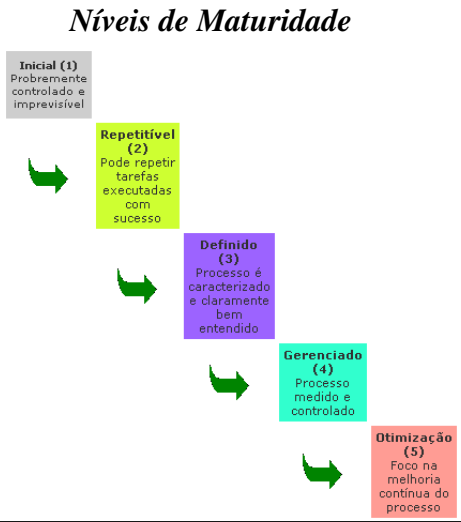
Comparando Processos de Software
Alta Cerimônia

- Características de projeto de alta cerimônia
  - Artefatos de gerenciamento, requisitos, design e testes são detalhados, documentado e colocados sob controle de versão
  - Links de rastreabilidade entre requisitos, elementos de design e artefatos de testes são criados e mantidos
  - ***Change Control Board*** aprovado é necessário para mudanças
  - Inspeção dos resultados são cuidadosamente registrados e localizados

Comparando Processos de Software
SEI CMM

- Modelo de maturidade de software (Capability Maturity Model)
- Referência: http://www.sei.cmu.edu/cmm
- Desenvolvido pela Software Engineering Institute (uma instituição vinculada à Carnegie-Mellon University)
- CMM não é um processo
- CMM é um framework para avaliação usado para determinar o nível de maturidade do processo de um organização

Comparando Processos de Software
SEI CMM



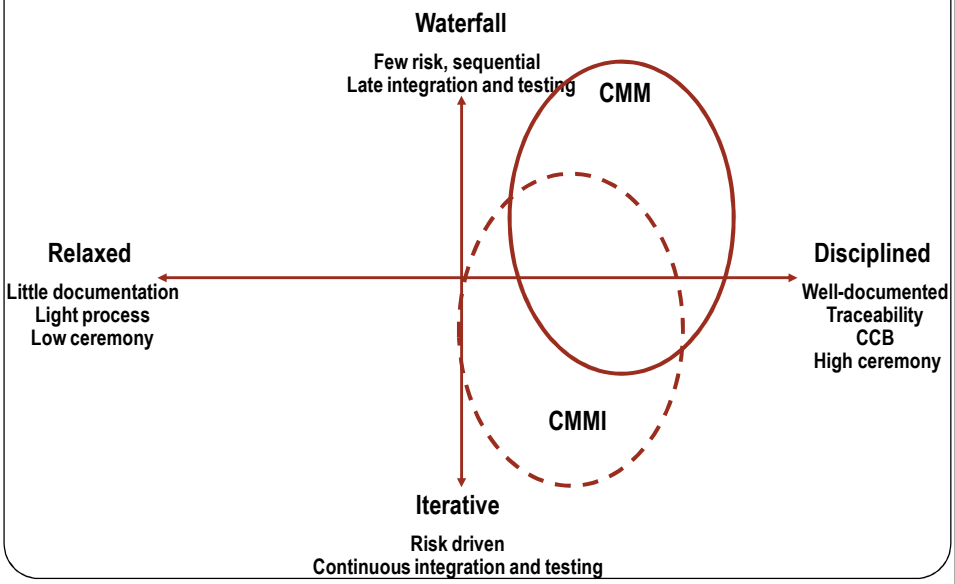
Comparando Processos de Software
SEI CMM

- Infelizmente algumas organizações e consultores de CMM interpretam de forma errada o framework CMM →leva a organização a adicionar artefatos e atividades desnecessárias
- O processo acaba ficando muito pesado!
- Outro efeito indesejado → encoraja ao desenvolvimento em cascata (waterfall)

Comparando Processos de Software
SEI CMMI

- Endereça os problemas do CMM
- Acomoda de forma mais efetiva as modernas boas práticas tais como:
  - Desenvolvimento dirigido a riscos
  - Desenvolvimento iterativo

Comparando Processos de Software
SEI CMM e SEI CMMi



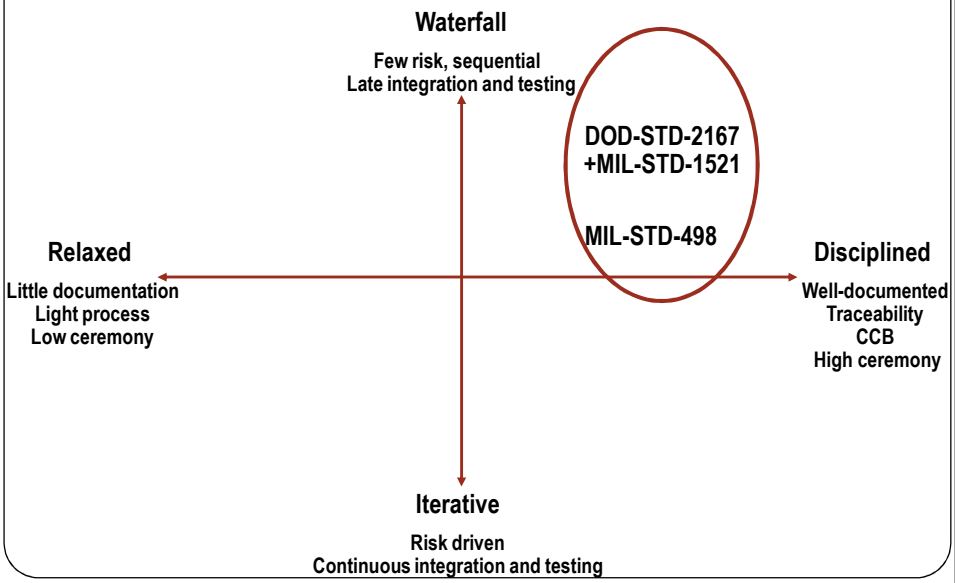
Comparando Processos de Software
ISO/IEC 15504

- Framework que avalia a maturidade do software em uma escala de 1 a 6
- Foi derivado do projeto SPICE (*Software Process Improvement and Capability Determination*)
- Muito similar ao SEI CMMI
- Por ser colocado o mesmo quadrante do CMMI

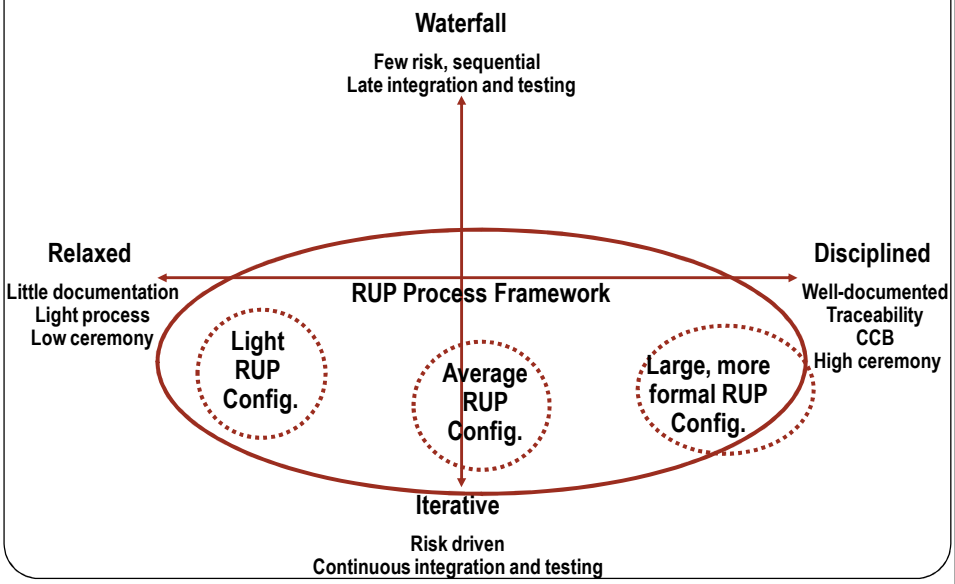
Comparando Processos de Software
Padrões DoD

- Desenvolvidos pelo U.S. Departament of Defense (DOD)
- Padrões que fornecem um desenvolvimento de software com alta cerimônia
- Tem como objetivo minimizar custos e melhorar a previsibilidade
- Alta cerimônia: DOD-STD-2167, DOD-STD-2167A, MIL-STD-1521B e MIL-STD-498
- Média cerimônia: MIL-STD-498

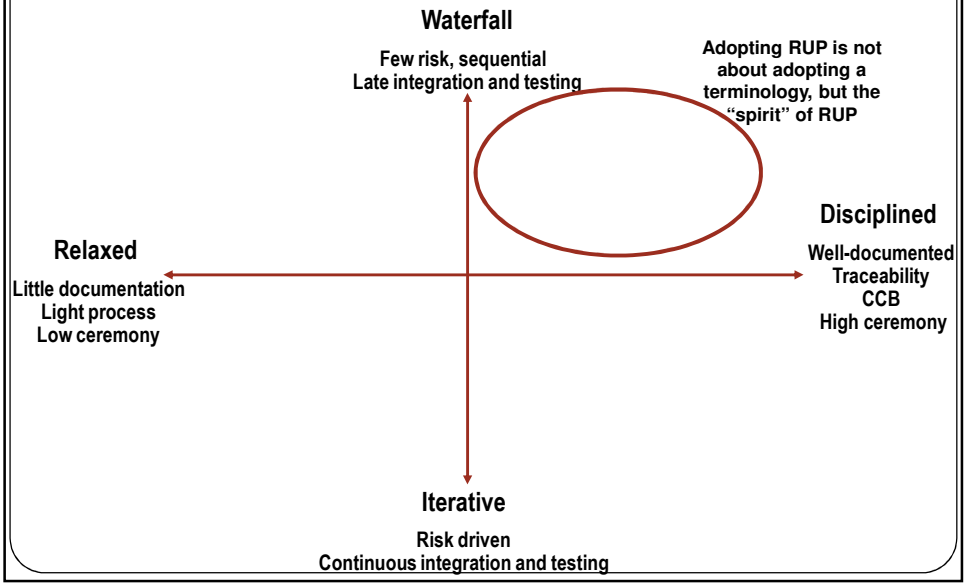
Comparando Processos de Software
Padrões DoD



Comparando Processos de Software
Framework RUP



Comparando Processos de Software
Uso Errado do RUP



O quanto iterativo você deve ser?

- Desenvolvimento iterativo e dirigido a riscos traz vários benefícios (diminuição nos custos, fidelidade no cronograma e etc)
- Fatores que impedem um organização a usar uma abordagem altamente iterativa e dirigida a riscos com integração e testes contínuos

O quanto iterativo você deve ser?

- Alguns fatores:
  - Time inexperiente para adotar uma abordagem iterativa
  - Falta de um bom suporte de processo
    - Desenvolvimento iterativo introduz vários novos desafios, especialmente para gerentes de projetos e arquitetos
    - É necessário um orientação de um processo para mitigar riscos o mais cedo possível
  - Falta de uma boas ferramentas de suporte
    - É muito difícil realizar um desenvolvimento iterativo sem uma boa ferramenta de suporte para automatizar testes e gerenciamento de mudanças e configuração

Quanta cerimônia você deseja?

- Prof. Barry Boehm →"Ágil" versus "Disciplinado"
  - Ler o artigo: "Get Ready for Agile Methods,with Care", 2002
- Fatores para o uso de uma abordagem com baixa cerimônia:
  - Mercados com rápidas mudanças
  - Time co-alocados
  - Times pequenos
  - Baixa complexidade técnica

Quanta cerimônia você deseja?

- Característica de baixa cerimônia: Pouca documentação (por exemplo, não documentar e acompanhar mudanças nos requisitos, design e testes)
- Fatores para o uso de uma abordagem com alta cerimônia:
  - Desenvolvimento de larga escala
  - Desenvolvimento distribuído
  - Projetos com complexidade técnica
  - Relacionamento complexos com stakeholders

Quanta cerimônia você deseja?

- Característica de alta cerimônia: muita documentação
- Fator crucial → usar ferramentas para automatizar
  - Gerenciamento de mudanças e configuração
  - Modelagem visual
  - Coleta de métricas
  - Testes
  - Etc…

Projeto Deimos

- O RUP para um time de uma pessoa
- Este exemplo nos mostrará o seguinte:
  - Como usar o processo
  - Seus princípios
  - Desenvolvimento iterativo
- ***A proposta de um processo de engenharia de software não é tornar a vida dos desenvolvedores um inferno ou esmagar a criatividade através de uma quantidade enorme de documentos***

Projeto Deimos
Exercício

- Mostrar como o RUP e sua filosofia são aplicados para o desenvolvimento deste projeto
- Identificar o ciclo de desenvolvimento em termos de fases e iterações
- Identificar os artefatos produzidos e a sua importância