

Processo de Desenvolvimento de Software

Aula 02

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

Agenda Aula 02

- Revisão RUP
- RUP estrutura estática: descrição do processo
- RUP estrutura dinâmica: desenvolvimento iterativo
- Exercício no Laboratório

RUP Revisão

O que é o RUP?

- Uma abordagem de desenvolvimento de software que é iterativa, centrada na arquitetura e dirigida por casos de uso
- Um bem definido e estruturado processo de software
- Um processo produto que provê um ambiente para customização de processo

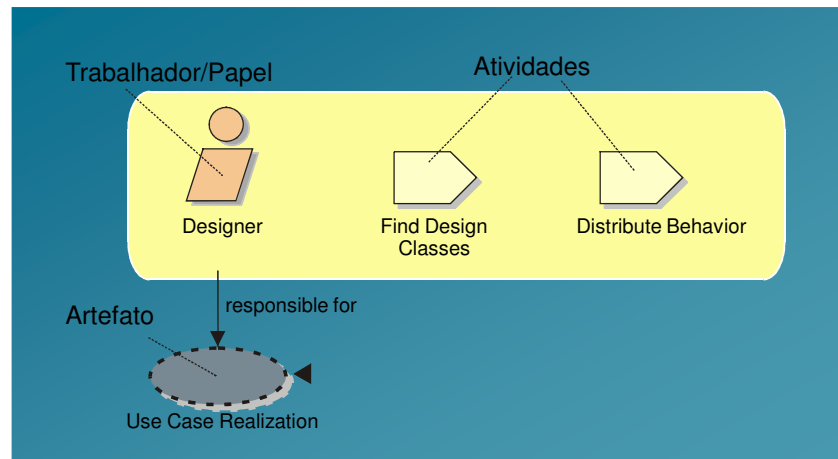
RUP Estrutura Estática

descrição do processo

- Um processo descreve quem está fazendo o quê, como e quando
- O RUP é representado usando quatro elementos primários de modelagem:
 - Trabalhadores (workers) ou Role: quem
 - Atividade: como
 - Artefato: o quê
 - Fluxos (workflows): quando

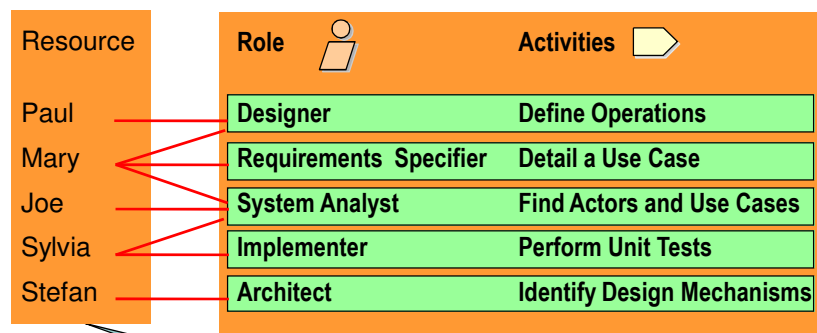
RUP Estrutura Estática

descrição do processo



RUP Estrutura Estática

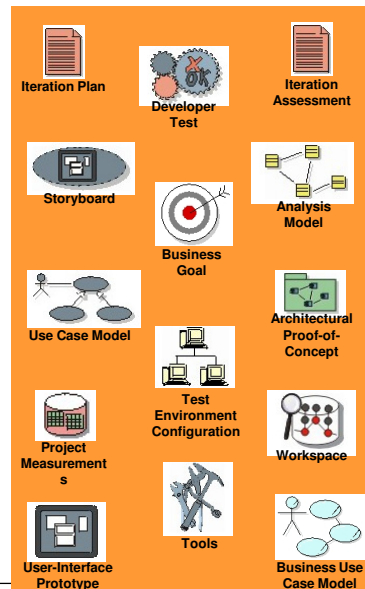
descrição do processo



Para cada indivíduo do projeto é atribuído um ou mais papéis.

RUP Estrutura Estática

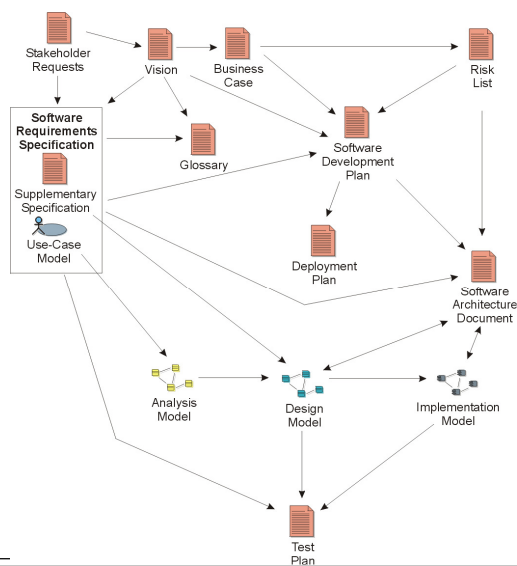
descrição do processo



RUP Estrutura Estática

descrição do processo

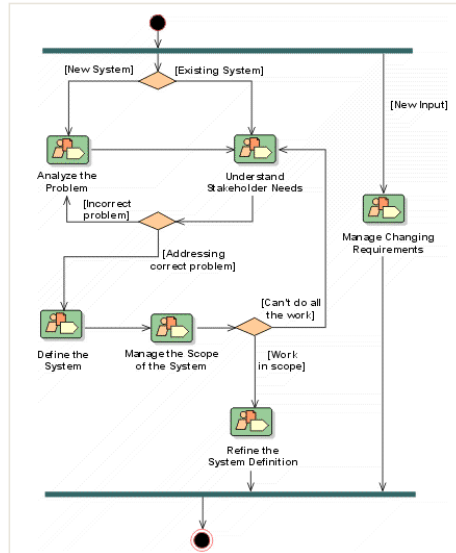
Principais Artefatos



RUP Estrutura Estática

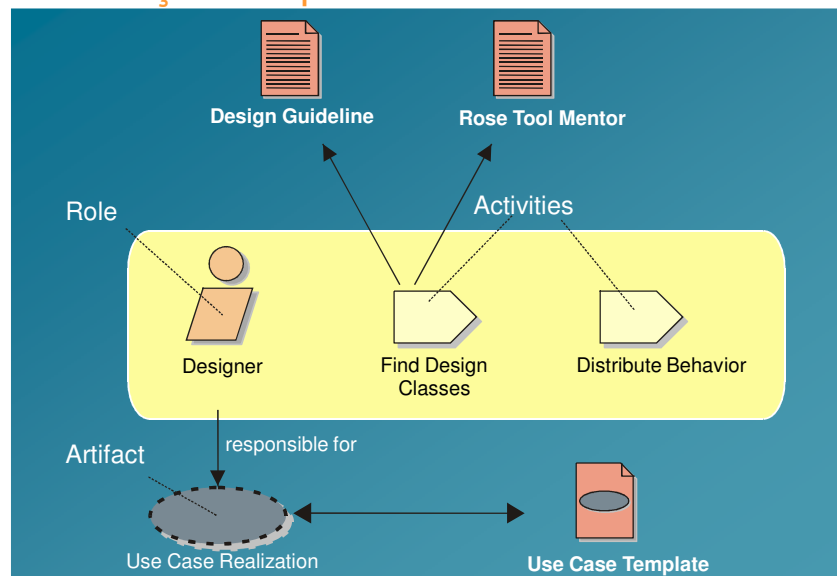
descrição do processo

Exemplo
Workflow
de Requisitos



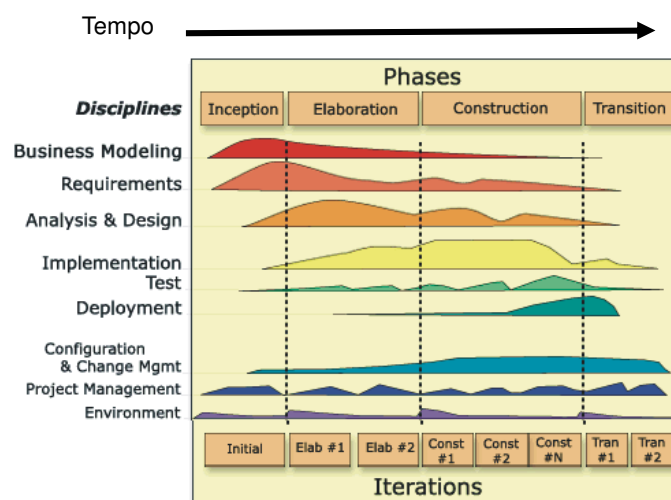
RUP Estrutura Estática

descrição do processo



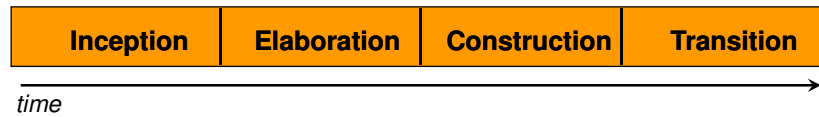
- RUP estrutura dinâmica
- Desenvolvimento Iterativo

Organização do RUP em Função do Tempo



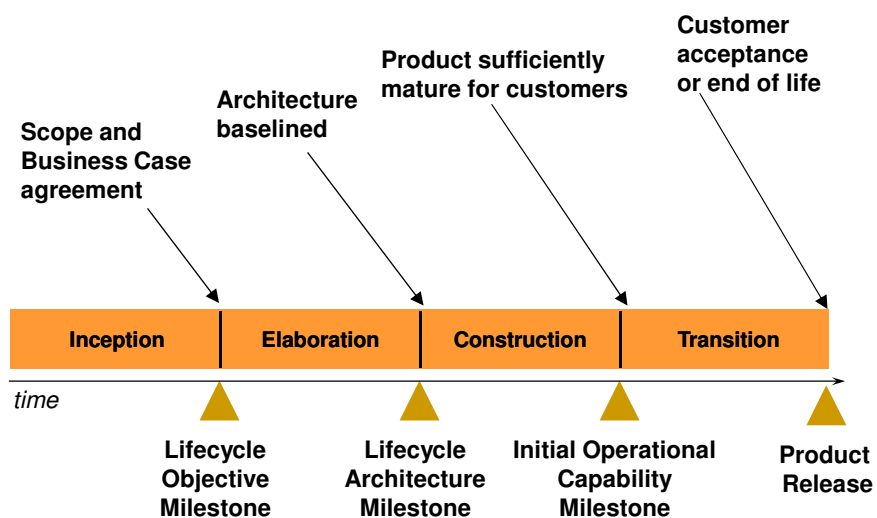
Organização por fases ajuda a minimizar os riscos de alocação de recursos.

RUP Estrutura Dinâmica



- Ciclo de vida ou dimensão de tempo de um projeto
- RUP fornece uma abordagem estruturada para o desenvolvimento iterativo → projeto é dividido em 4 fases

RUP Estrutura Dinâmica



RUP

Concepção: **saber o que construir**

- Preparar o documento de visão e os business case inicial
 - Incluem avaliação de riscos e estimativa de recursos
- Desenvolver os requisitos do projeto em alto nível
 - Casos de uso iniciais e os modelos do domínio (10-20% completo)
- Gerenciar o escopo do projeto
 - Reduzir riscos através da identificação dos requisitos chaves
 - Reconhecer que os requisitos irão mudar
 - Gerenciar mudanças e usar processo iterativo

Concepção

Elaboração

Construção

Transição

RUP

Concepção: **critérios de avaliação**

- Stakeholder concordam com a definição do escopo e das estimativas de custo/prazo
- Acordo que o conjunto de requisitos capturado está correto e que existe um entendimento compartilhado desses requisitos
- Acordo que as estimativas de custo/prazo, prioridades, riscos e processo de desenvolvimento estão apropriadas
- Todos os riscos foram identificados e as estratégias de mitigação existiram para cada um

Milestone: Lifecycle Objectives (LCO)

Concepção

Elaboração

Construção

Transição

RUP

Elaboração: **saber como construir**

- Detalhar os requisitos mais importantes (~80% completo)
 - Requisitos menos importantes pode ser deixados de lado
- Produzir um arquitetura estável e executável
 - Definir, implementar e testar as interfaces dos principais componentes
 - Identificar dependências com componentes e sistemas externos. Integrar shells/proxies
 - Alguns dos componentes chaves são parcialmente implementados
 - Aproximadamente 10% do código é implementado
- Dirigir a arquitetura com os casos de uso principais
 - 20% dos casos de uso dirigem 80% da arquitetura
 - Projeto, implementação e testes dos cenários chaves para os casos de uso

Concepção

Elaboração

Construção

Transição

RUP

Elaboração: **saber como construir**

- Verificar as qualidades arquiteturais
 - Confiabilidade: teste de stress
 - Escalabilidade de Desempenho: teste de carga
- Avaliar continuamente os business case, perfis de riscos e o plano de desenvolvimento

Concepção

Elaboração

Construção

Transição

RUP

Elaboração: critérios de avaliação

- Visão do Produto e requisitos estão estáveis
- Arquitetura está estável
- Testes chaves e abordagens de avaliação foram provadas;
- Elementos dos riscos principais foram endereçados e resolvidos
- Planos de iteração para a fase de Construção estão o suficientemente detalhados e fidedignos para proceder a realização do trabalho e são suportados por estimativas acreditáveis
- Todos os stakeholders concordam que a visão atual pode ser alcançada se o plano atual for executado para o desenvolvimento de todo o sistema, considerando o contexto da arquitetura atual
- Gastos atuais X gastos planejados são aceitáveis

Milestone: Lifecycle Architecture (LCA)

Concepção	Elaboração	Construção	Transição
-----------	------------	------------	-----------

RUP

Construção: construir o produto

- Completo requisito e modelo de projeto
- Projeto, Implementação e testes de cada componente
 - Protótipo do sistema e envolvimento dos usuários finais
 - Incrementalmente envolver a arquitetura executável para completar o sistema
- Construir diariamente ou semanalmente com processo de construção automatizado
- Testar cada build
 - Teste de regressão automático
 - Teste de carga e stress para assegurar a integridade arquitetural
- Entregar funcionalidade total do software (versão beta)
 - Incluindo material de treinamento e documento de usuário e implantação
- Produzir a descrição das entregas

Concepção	Elaboração	Construção	Transição
-----------	------------	------------	-----------

RUP

Construção: critérios de avaliação

- O critério de avaliação para a fase de Construção envolve as respostas para as seguintes questões:
- O produto entregue está maduro e estável o suficiente para ser disponibilizados para a comunidade de usuários?
- Todos os stakeholders estão prontos para a implantação do produtos na comunidade de usuários?
- Os gastos atuais x gastos previstos ainda estão aceitáveis?

Milestone: Initial Operational Capability (IOC)

Concepção	Elaboração	Construção	Transição
-----------	------------	------------	-----------

RUP

Transição: entregar o produto

- Produzir incrementalmente releases de correções
- Atualizar o manual do usuário e documentação de implantação
- Atualizar as descrições de releases
- Conduzir uma análise “post-mortem” do projeto

Concepção	Elaboração	Construção	Transição
-----------	------------	------------	-----------

RUP

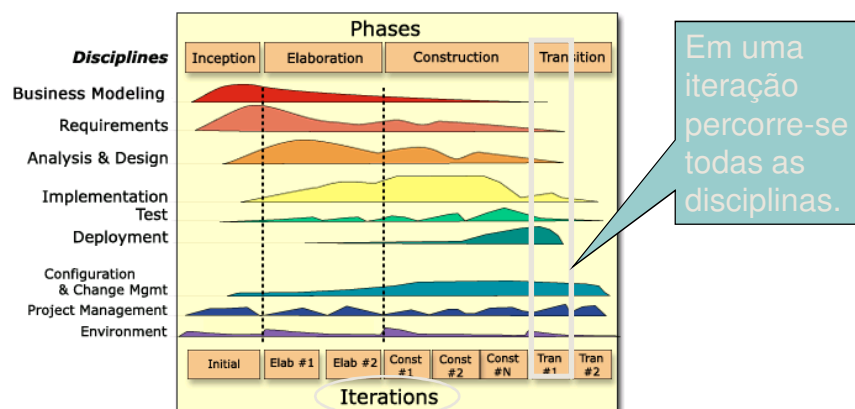
Transição: critérios de avaliação

- Os critérios de avaliação principais para a fase de Transição envolvem a resposta para as seguintes questões:
 - O usuário está satisfeito?
 - Os gastos atuais x gastos planejados são aceitáveis?

Milestone: Product release

Concepção	Elaboração	Construção	Transição
-----------	------------	------------	-----------

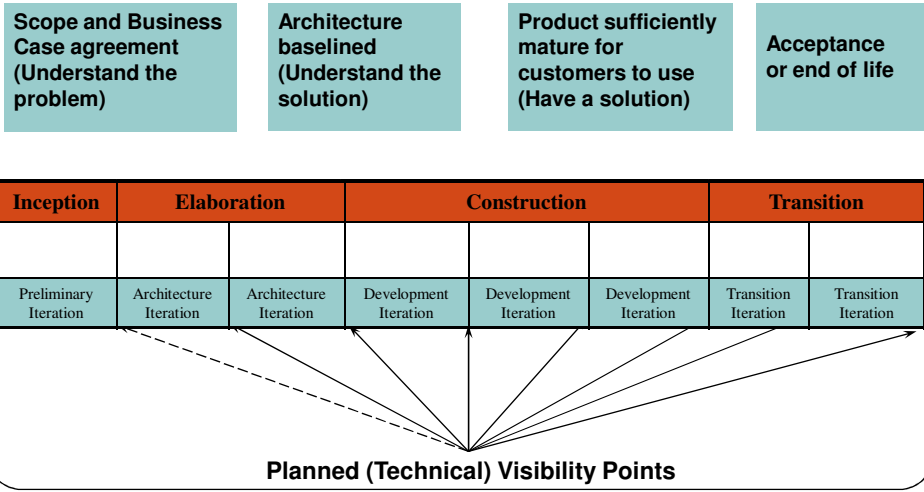
O que é uma Iteração?



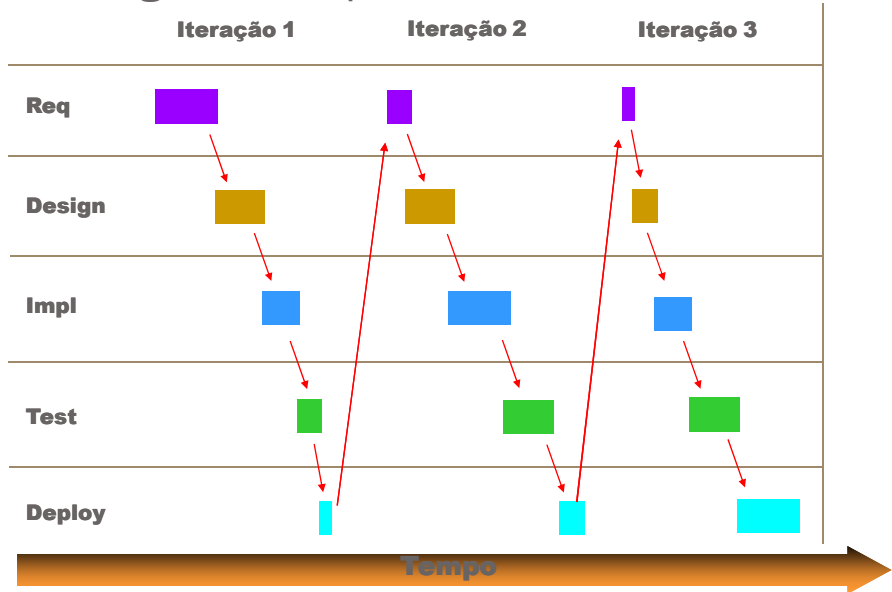
Iteração: uma sequência distinta de atividades com um plano delineado e critérios de avaliação que resultam em um release (interno ou externo)

Mudança de Foco das Fases ao Longo do Tempo

Planned (Business) Decision Points



Mudança de Foco das Iterações ao Longo do Tempo



Duração de uma Iteração

- Uma iteração começa com planejamento e requisitos, e finaliza com um release interno ou externo.
- Idealmente uma iteração dever ser realizada em duas a seis semanas, dependendo do tamanho e complexidade do projeto
- Fatores que afetam a duração de um iteração:
 - Tamanho, estabilidade e maturidade da organização
 - Familiaridade com processo iterativo
 - Tamanho do projeto
 - Simplicidade técnica do projeto
 - Nível de automação usada para gerenciar o código, distribuir informações e realizar os testes

Número de Iterações

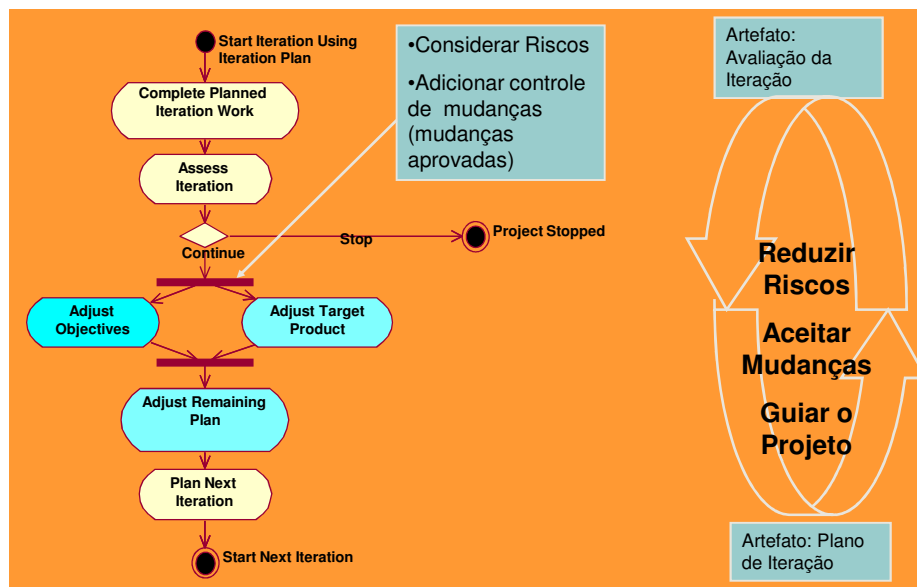
- Regra do “dedo polegar”: Usar 6 ± 3 iterações

Fase	Baixa	Média	Alta
Inception	0	1	1
Elaboration	1	2	3
Construction	1	2	3
Transition	1	1	2
Total	3	6	9

Condições que aumentam o número de iterações

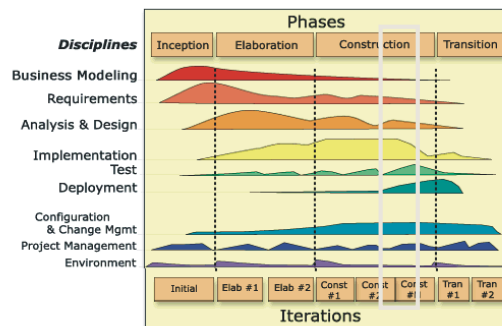
Inception <ul style="list-style-type: none"> ▪ Trabalhar com novas funcionalidades ▪ Desconhecer o ambiente do negócio ▪ Alta volatilidade do escopo ▪ Decisões fazer ou comprar 	Elaboration <ul style="list-style-type: none"> ▪ Trabalhar com nova tecnologia (novas características arquitetural) ▪ Elementos arquiteturais não testados ▪ Necessidade de protótipos do sistema
Construction <ul style="list-style-type: none"> ▪ Muito código para implementar e verificar ▪ Novas tecnologias ou ferramentas de desenvolvimento 	Transition <ul style="list-style-type: none"> ▪ Necessidades de Alfa e Beta ▪ Migração/Conversão da base do cliente ▪ Entrega incremental para o cliente

Uma Iteração



Artefato: Plano de Iteração

- Um tempo sequenciado de atividades e tarefas com recursos atribuídos e dependências entre tarefas. Um plano de fina granularidade, um por iteração.



Exemplo de Plano de Iteração

- Mostra a janela de tempo e recursos por disciplina

Cronograma de iteração para a disciplina de Requisitos

- Objectives
- Scope
- References
- Plan
 - Iteration Activities
 - Iteration Schedule
 - Iteration Deliverables
- Resources
 - Staffing Resources
 - Financial Resources
 - Equipment & Facilities Resources
- Use Cases
- Evaluation Criteria

Esboço de um plano de iteração

Requirements	40 days	Tue 12/1/98	Mon 1/25/99	
Develop Vision	25 days	Tue 12/1/98	Mon 1/4/99	System Analyst
Elicit Stakeholder Requests	4 days	Tue 1/5/99	Fri 1/8/99	System Analyst
Manage Dependencies	26 days	Tue 12/1/98	Tue 1/5/99	System Analyst
Capture a Common Vocabulary	10 days	Wed 12/23/98	Tue 1/5/99	System Analyst

Artefato: Avaliação da Iteração

- A avaliação da iteração captura os resultados de uma iteração, o grau na qual os critérios de avaliação foram atendidos, lições aprendidas e mudanças a serem feitas.

Conceitos que Dirigem o Desenvolvimento Iterativo

- Alguns conceitos importante que afetam o desenvolvimento iterativo são:
 - Mitigação dos riscos o mais cedo possível
 - Delinear uma arquitetura o mais cedo possível
 - Usar métricas objetivas
- *Vamos discutir rapidamente esses conceitos*

Definição de Risco

- É uma exposição à perda ou dano, ou um fator, coisa, elemento ou curso que envolve perigo incerto
- Risco em desenvolvimento de software → variável que, dentro da sua distribuição normal, pode levar um valor que arrisca ou elimina o sucesso do projeto

Alguns tipos de riscos:

Riscos arquiteturais/técnicos

Tecnologia não comprovada, escopo incerto

Riscos de recursos

Pessoas, skills, \$

Riscos de Negócio

Competição, ROI, fornecedores

Riscos de Prazo

Dependências do projeto

Apenas 24 horas em um dia!

Precisam ser identificados e priorizados no artefato Lista de Riscos

Qualificador de Risco

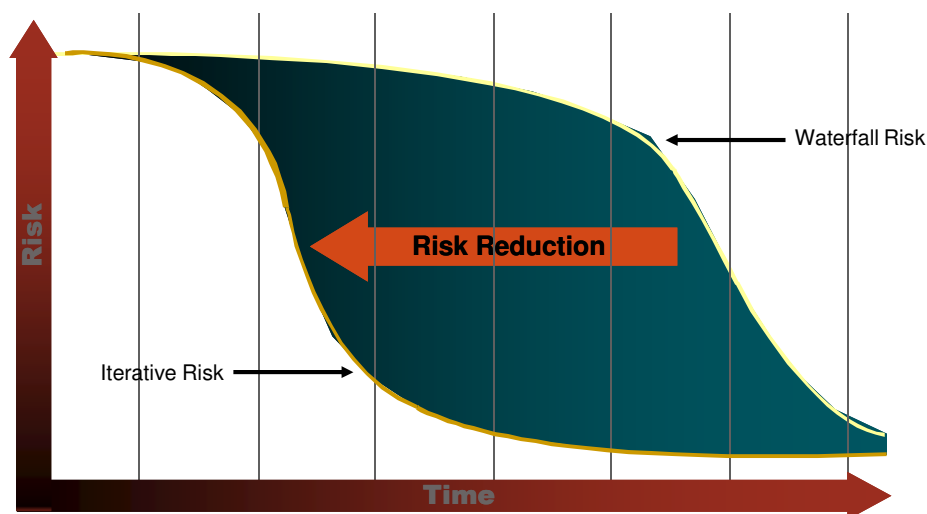
- **Risco Direto** – um risco sobre o qual o projeto tem um alto grau de controle
- **Risco Indireto** – um risco sobre o qual o projeto tem pouco ou nenhum controle
- **Magnitude do Risco** é usado para classificar os riscos. É um combinação de:
 - Probabilidade da ocorrência
 - Impacto sobre o projeto (severidade) ex: atrasos no projeto

Estratégias de Gerenciamento de Riscos

- Evitar o risco, transferir o risco ou aceitar o risco
- Mitigar os riscos aceitáveis através da:
 - Criação do artefato Lista de Riscos
 - Postura imediata e atitudes pro-ativa para reduzir a probabilidade ou o impacto de um risco
 - Definir um plano de contingência

A Lista de Riscos é criada na fase de Concepção e atualizada em toda iteração.

Perfis dos Riscos

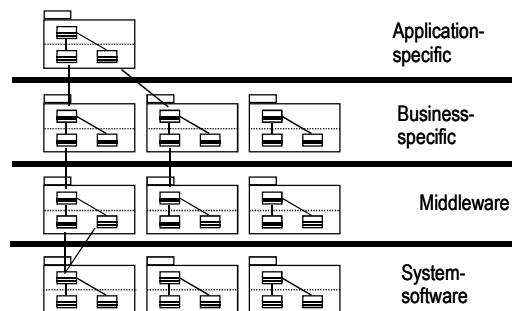


Redução de Riscos “dirige” o ciclo de vida Iterativo

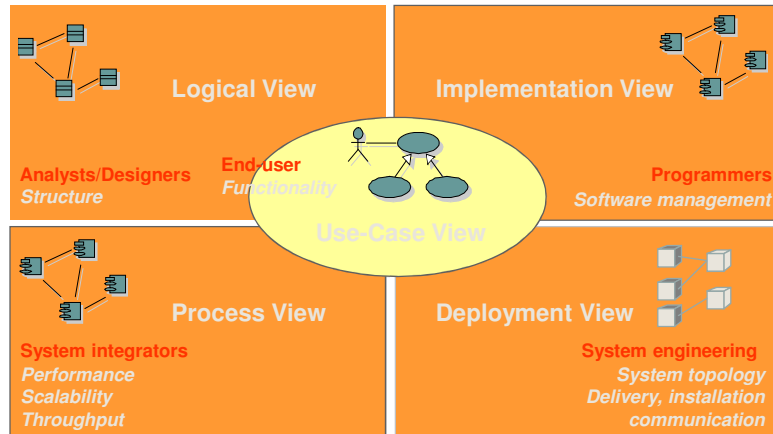
- Primeira iteração devem endereçar os riscos mais importantes
- Avaliação de riscos é um processo contínuo; riscos mudam ao longo do tempo
- A Lista de Riscos atualizada é usada como entrada para a atividade Desenvolver o Plano de Iteração

Uma Arquitetura Executável

- Uma validação (testada) da arquitetura
 - Testada através dos casos de uso que tem o maior impacto na arquitetura
- A linha base para o resto do desenvolvimento
- Fornece mitigação para diversos riscos



Controle Intelectual: Visões Arquiteturais 4+1



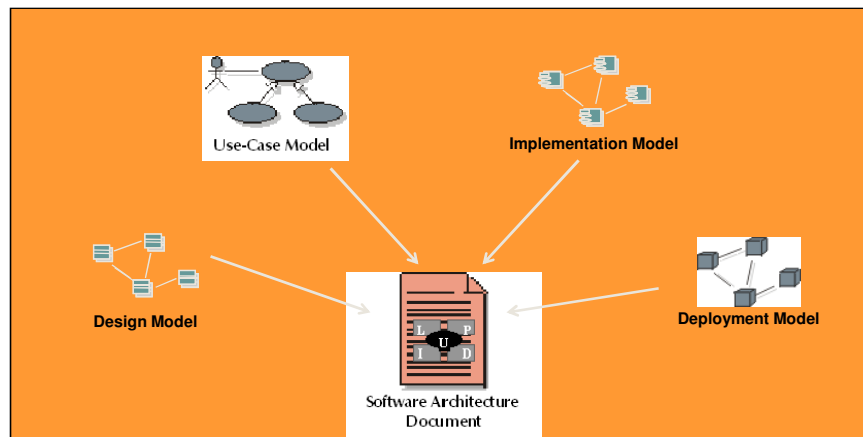
Descrição da Arquitetura



- Documento da Arquitetura do Software
 - Representa a visão geral da arquitetura de um sistema de software
 - Inclui:
 - Visões Arquiteturais
 - Objetivos e Restrições
 - Requisitos que a arquitetura deve suportar
 - Restrições técnicas
 - Casos de mudanças
 - Características de tamanho e desempenho
 - Qualidade, extensibilidade e portabilidade

Descrição da Arquitetura

- Visões são extraídas de outros artefatos



Arquitetura com Base para Gerência de Projeto

- Um arquitetura estável representa um importante milestone do projeto
- Arquitetura pobre é frequentemente a razão para o insucesso do projeto
- Arquitetura é um pré-requisito para a previsão do projeto
- Arquitetura é a fonte de numerosas decisões

Arquitetura Delineada o mais cedo possível

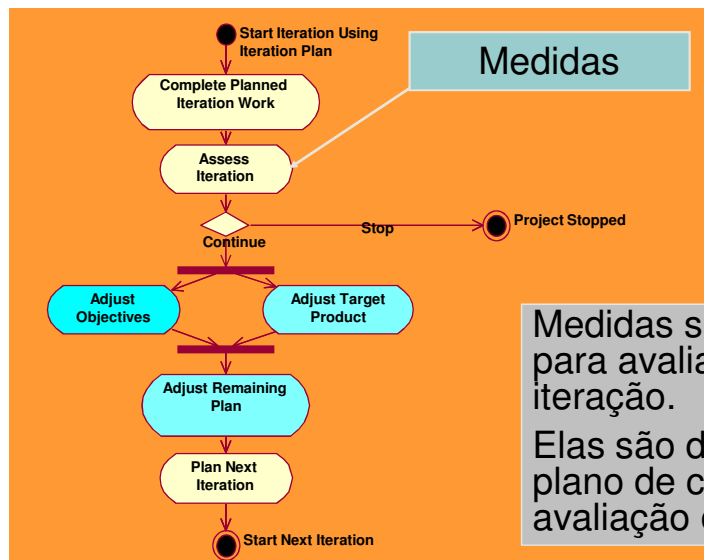
- Arquitetura
 - Dirige a definição das fases
 - Dirige o conteúdo das iterações
 - Dirige a organização do time de desenvolvimento
- No RUP, a arquitetura é delineada até o final da fase de elaboração e refinada nas iterações seguintes

Uso de Medidas Objetivas

- Medidas são usadas para fornecer o seguintes para o time de desenvolvimento:
 - Uma avaliação precisa da progresso
 - Percepção da qualidade que envolve o produto software
 - A base para estimativa de custos e prazo para finalizar o produto com aumento da precisão ao longo do tempo

Medidas devem refletir o estado “real” do que é medido.
Medidas subjetivas tem pouco valor.

Uso de Medidas em Uma Iteração



Medidas são usadas para avaliar uma iteração.

Elas são definidas no plano de critérios de avaliação da iteração.

Coleção de Medidas

- Medidas devem:
 - Ser simples e objetivas
 - Ser fáceis de coletar
 - Ser fáceis de interpretar
- Coleção de medidas deve:
 - Ser automatizada e não evasivo ao desenvolvedor
 - Colabora para a avaliação da qualidade o mais cedo possível no ciclo de vida, onde os esforços para a melhoria da qualidade são mais efetivos
- Medidas de valores absolutos e tendências devem:
 - Ser ativamente usadas para comunicar o progresso e qualidade em um formato consistente
- A seleção de um conjunto de medidas dependerão das características do projeto e do contexto

As Sete Principais Métricas

- Existem sete principais métricas que devem ser usadas no desenvolvimento iterativo:
- Indicador de Gerenciamento:
 - Trabalho e progresso (trabalho realizado ao longo do tempo)
 - Orçamento e gastos (custos recorrentes ao longo do tempo)
 - Pessoal e dinâmica do time (mudanças de pessoas ao longo do tempo)
- Indicadores de Qualidade:
 - Mudança de tráfego e estabilidade (mudanças de tráfego ao longo do tempo)
 - Ruptura ou modularidade (média de ruptura por mudanças ao longo do tempo)
 - Retrabalho e adaptabilidade (média de retrabalho por mudança ao longo do tempo)
 - Média de tempo entre as falhas (MTBF) e maturidade (taxa de defeitos ao longo do tempo)

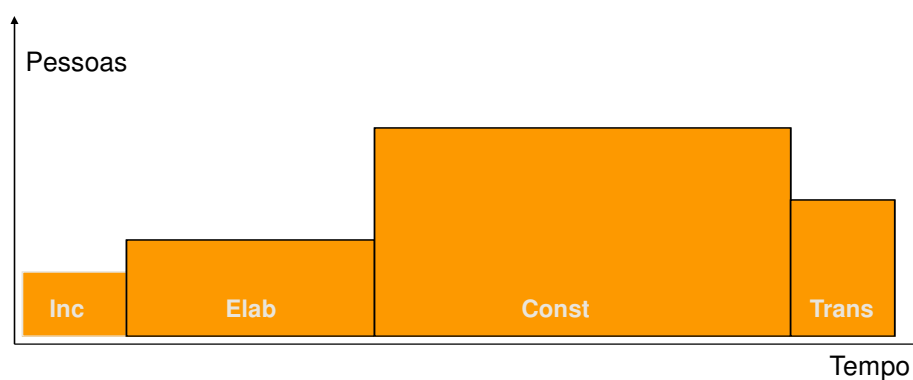
Propostas das sete principais Métricas

Métricas	Proposta
Trabalho e progresso	Planejamento da iteração, plano x atual, indicadores de gerenciamento
Orçamento e gastos	Percepção financeira, plano x atual, indicadores de gerenciamento
Dinâmica do pessoal e do time	Plano de recursos x atual, taxas escondidas, taxas em atrito
Mudanças de tráfego e Estabilidade	Plano de iteração, indicadores de gerenciamento da convergência do cronograma
Ruptura e modularidade	Convergência, esmagamento do software, indicador de qualidade
Retrabalho e adaptabilidade	Convergência, retrabalho do software, indicador de qualidade
MTBF e maturidade	Adequação e cobertura do teste, robustez para o uso, indicador de qualidade

Características do Desenvolvimento Iterativo

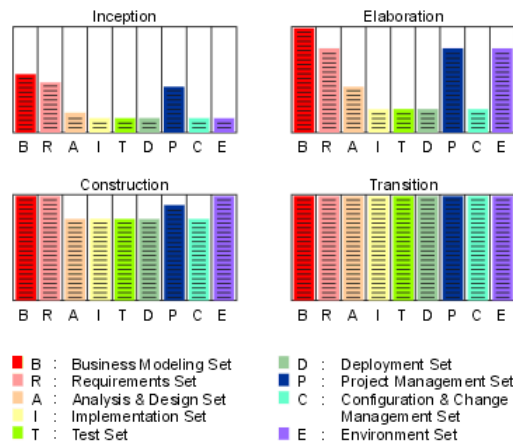
- Você pode controlar alocação de recursos por fase
- Você pode evoluir os artefatos conforme requerido por fase
- Você pode aumentar a precisão das estimativas de custo de fase para fase

Porcentagem de Tempo e Esforço por Fase



	Inc	Elab	Const	Trans
Esforço	5%	20%	65%	10%
Tempo/Prazo	10%	30%	50%	10%

Evolução do Artefatos ao Longo do Tempo



Information set evolution over the development phases.

- Com a abordagem iterativa, os artefatos amadurecem ao longo do tempo

Fidelidade da Estimativa de Custo

