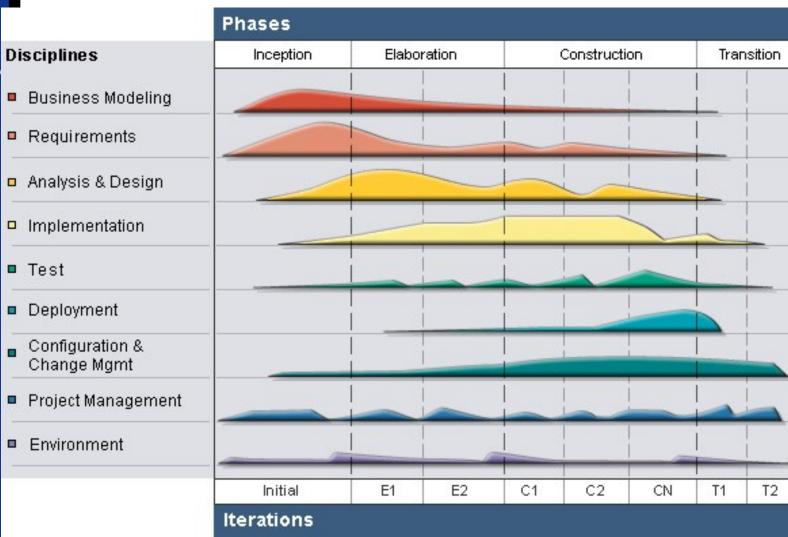


Rational Unified Process (RUP)

Visão Geral

http://wthreex.com/rup/







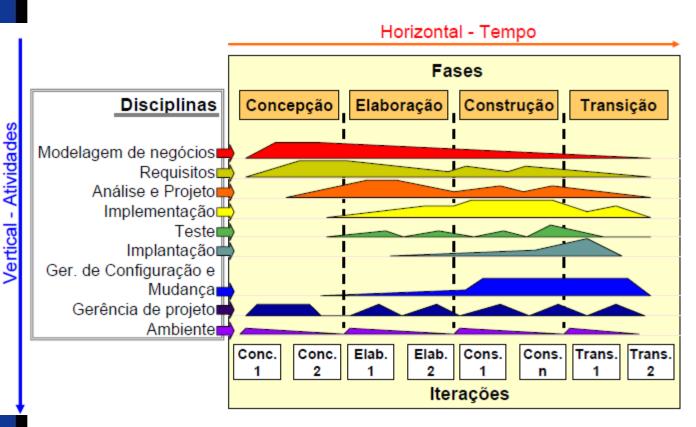
Estrutura de Processo

- Duas dimensões
 - Vertical
 - Estrutura estática do processo
 - Descreve como elementos do processo atividades, artefatos e papéis – são logicamente agrupados em disciplinas (ou fluxos de trabalho)
 - Horizontal
 - Estrutura dinâmica ou de tempo do processo
 - Expressa em termos de ciclos, fases, iterações e marcos ao longo do ciclo de vida de um projeto



Estrutura do Processo

Dimensões Horizontais e Verticais





- Estrutura estática: quatro elementos chave
 - Papéis (ou trabalhadores)
 - Quem?
 - Atividades
 - Como?
 - Artefatos
 - O que?
 - Fluxos de trabalho (workflows)
 - Quando? Em que seqüência?



- Papéis (ou trabalhadores)
 - Como um chapéu usado por um indivíduo (ou grupo) durante o projeto
- Um indivíduo pode usar vários chapéus...
- desempenhando vários papéis





- Atividades
 - Unidade de trabalho a ser realizado por um indivíduo que desempenha um determinado papel
 - Em geral, possui um propósito claro, expresso em termos da criação ou atualização de artefatos, tais como modelos, componentes ou planos.
 - De poucas horas a poucos dias
 - Em geral, envolve uma pessoa



- Artefatos
 - Pedaços de informação produzidos, modificados ou usados no processo
 - Podem ser:
 - Um modelo, como um diagrama de casos de uso
 - Um elemento de modelo, como uma classe
 - Um documento, como um guia de usuário
 - Código fonte
 - Executáveis, como um protótipo do software



- □ Fluxos de trabalho
 - Sequência de atividades e interação entre papéis que produzem artefatos
 - Fluxos de trabalho de alto nível são chamados de *Disciplinas*

Principais

- Modelagem de negócio
- Requisitos
- Análise e projeto (design)
- Implementação
- Teste
- Implantação (deployment)

Complementares

- Gerência de configuração e mudança
- Gerência de projeto
- Ambiente



- Disciplina: Modelagem de negócio
 - Entender a estrutura e a dinâmica da organização na qual um sistema deve ser implantado (a organização alvo).
 - Entender os problemas atuais da organização alvo e identificar as possibilidades de melhoria.
 - Assegurar que os clientes, usuários e desenvolvedores tenham um entendimento comum da organização alvo.
 - Derivar os requisitos de sistema necessários para sustentar a organização alvo.
 - Custo, tempo, esforço!!! Lucro????



- Disciplina: Requisitos
 - Estabelecer e manter concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer.
 - Oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos do sistema.
 - Definir as fronteiras do sistema (casos de uso).
 - Fornecer uma base para planejar o conteúdo técnico das iterações.
 - Fornecer uma base para estimar o custo e o tempo de desenvolvimento do sistema.
 - Definir uma interface de usuário para o sistema, focando nas necessidades e metas dos usuários.



- Disciplina: Análise e projeto
 - Transformar os requisitos em um projeto/desenho do sistema a ser criado.
 - Desenvolver uma arquitetura sofisticada para o sistema.
 - Adaptar o design para que corresponda ao ambiente de implementação, projetando-o para fins de desempenho



- Disciplina: Implementação
 - Definir a organização do código em termos de subsistemas de implementação organizados em camadas.
 - Implementar classes e objetos em termos de componentes (arquivos-fonte, binários, executáveis e outros).
 - Testar os componentes desenvolvidos como unidades.
 - Integrar os resultados produzidos por implementadores individuais (ou equipes) ao sistema executável.



- Disciplina: Teste
 - Localizar e documentar defeitos na qualidade do software.
 - Avisar de forma geral sobre a qualidade observada no software.
 - Validar as suposições feitas nas especificações de projeto e requisito através de demonstração concreta.
 - Validar as funções do software conforme projetadas.
 - Verificar se os requisitos foram implementados de forma adequada.



- Disciplina: Implantação
 - Descreve as atividades que garantem que o produto de software será disponibilizado a seus usuários finais.
 - Produzir versões de entrega (releases)
 - Empacotar, distribuir e instalar o software
 - Prover assistência ao usuário
 - Migração de software e dados



- Disciplina: Gerência de configuração e mudança
 - Controle de versão
 - Gerenciamento de mudanças
 - Equipes distribuídas???
 - Rastreamento de bugs
 - Notificação???
 - Existem várias ferramentas específicas
 - Fora do escopo do curso
 - Maven, CVS, SVN, Bugzilla, Cruise Control...



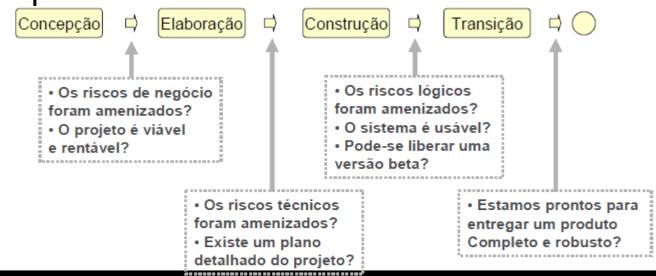
- □ **Disciplina:** Gerência de projeto
 - Fornecer um modelo para gerenciar projetos intensivos de software.
 - Fornecer diretrizes práticas para planejar, montar a equipe, executar e monitorar os projetos.
 - Fornecer um modelo de gerenciamento de risco.
 - Ferramentas:
 - IBM Rational, MS Project...



- □ **Disciplina:** Ambiente
 - A meta das atividades dessa disciplina é oferecer à organização o ambiente de desenvolvimento de software que dará suporte à equipe de desenvolvimento.
 - Ferramentas: IDEs (Eclipse, NetBeans), ferramentas de modelagem, gerenciadores de banco de dados...



- Orientadas a marcos (miles tones)
 - Não é uma partição de atividades: análise, projeto, codificação...
 - Cada fase possui necessidades dirigidas por *risco!!!*





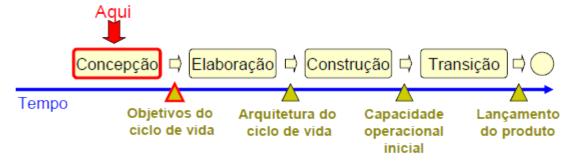
- Não há fluxos de trabalho fixos
 - RUP não é uma receita fixa
 - Deve ser adaptado pela equipe de desenvolvimento de acordo com suas necessidades
- Com base no contexto do projeto, definese o que é necessário para atingir os marcos de cada fase



- Não há artefatos "congelados"
 - Nem todos os artefatos são necessários
 - Depende do sistema e da equipe em questão
 - Deve ser adaptado a necessidades específicas



- □ Fase de Concepção
 - Entender o que construir
 - Identificar funcionalidades chave
 - Determinar pelo menos uma possível solução
 - Entender os custos, agenda e riscos
 - Decidir que processo seguir e que ferramentas usar





Entendendo as fases do RUP Fase de Concepção

- Entender o que construir
 - Produzir um Documento de Visão
 - Beneficios e oportunidades; o problema a ser resolvido; usuários-alvo; descrição de alto nível das funcionalidades; alguns requisitos não funcionais essenciais
 - Não tem formato específico!
 - Prover uma descrição superficial do sistema
 - Identificar e descrever resumidamente os principais atores e casos de uso
 - Detalhar atores e casos de uso
 - De acordo com a necessidade, detalhar os principais casos de uso e atores



Fase de Concepção

- Identificar funcionalidades chave
 - Cliente deve ser requisitado para elucidar os casos de uso
 - Deve-se ter em mente um conjunto de funcionalidades que formem um todo "entregável" ao usuário final
 - Não dá para entregar um sistema de cadastro de alunos, se não é possível adicionar um novo aluno no cadastro!



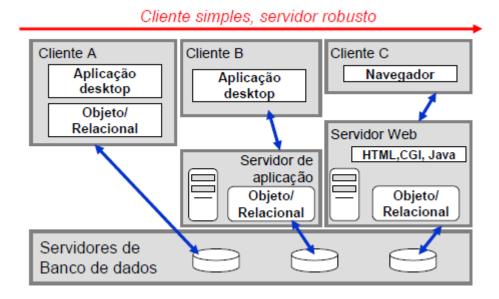
Fase de Concepção

- Determinar pelo menos uma possível solução...
 - Que outro sistema similar foi construído? Que tecnologia e arquitetura foi usada? Qual foi o seu custo?
 - No caso de sistema em evolução, a arquitetura ainda é satisfatória?
 - Que tecnologias deverão ser usadas no sistema? Alguma tecnologia deverá ser adquirida? Quais os custos e riscos associados?
 - Quais componentes de software são necessários no sistema? Eles podem ser comprados/obtidos?Podem ser reutilizados de outro projeto? Quais são os custos e riscos estimados?



Fase de Concepção

... de preferência, várias!!!





Fase de Concepção

- Entender os custos, agenda e riscos
 - Vale a pena continuar no projeto?
 - Quais os custos relacionados?
 - Quanto tempo você levará para concluir o projeto?
 - Qual o valor econômico do produto? Qual o retorno de investimento?





Entendendo as fases do RUP Fase de Concepção

- Revisão do projeto: Marco de objetivos do ciclo de vida
 - A equipe concorda com a definição de escopo do sistema e a estimativa inicial de tempo e custo?
 - Há um acordo de que o conjunto certo de requisitos foi capturado e compartilha-se o entendimento dos mesmos?
 - Há acordo de que as prioridades, riscos e processo são apropriados?
 - Há acordo sobre a estratégia de eliminação dos riscos?
- O projeto pode ser abortado ou reconsiderado se falhar neste marco
 - Reconsiderado = Nova iteração



- □ Fase de Elaboração
 - Detalhar os requisitos
 - Projetar, implementar e validar a arquitetura
 - Detalhar riscos, agenda e custos
 - Refinar processo e ambiente





- Detalhar os requisitos
 - Resultado da fase anterior: uma visão e 20% dos principais casos de uso
 - No fim da fase de elaboração, você deveria descrever de forma completa a maioria dos casos de uso (80%)
 - Protótipo de interface gráfica para os principais casos de uso



- Projetar, implementar e validar a arquitetura
 - Definir os blocos mais importantes e suas interfaces, assim como a decisão de construi-los, reutilizá-los ou comprá-los
 - Descrever como os blocos interagem para implementar as funcionalidades
 - Implementar e testar um protótipo da arquitetura... arquitetura executável



- Detalhar riscos, agenda e custos
 - Com o detalhamento dos requisitos, aprimora-se a visão
 - Sabe-se com maior grau de certeza o que o software é!
 - Com a arquitetura executável, boa parte do trabalho está pronto
 - Você já sabe "o que falta fazer"
 - A maioria dos riscos foi eliminada
 - Isto reduz possíveis erros de estimativa de tempo e custo



- Refinar processo e ambiente
 - Na elaboração, você fez algum projeto, implementação e teste da arquitetura
 - O código já está sob gerência de configuração
 - Entende-se melhor as ferramentas, pessoas e o processo
 - Refinamento
 - Configuração mais precisa do ambiente



Entendendo as fases do RUP Fase de Elaboração

- Revisão do projeto: Marco de arquitetura do ciclo de vida
 - A visão e requisitos são estáveis? A arquitetura é estável?
 - Testes e avaliação demonstraram que os principais riscos foram eliminados?
 - Os planos para construção estão suficientemente detalhados para prosseguir? As estimativas têm credibilidade?
 - A equipe concorda que a visão pode ser alcançada?
 - A relação gasto ocorrido x planejado é aceitável?
 - Se falhar neste marco... nova iteração
 - O prejuízo de abortar já é bem maior... e a possibilidade bem menor



- □ Fase de Construção
 - Minimizar o custo de desenvolvimento
 - Desenvolver um produto pronto para a transição





Fase de Construção

- Minimizar o custo de desenvolvimento
 - Atingir algum grau de paralelismo
 - Organizar o desenvolvimento em torno da arquitetura e equipe de arquitetos
 - Gerência de configuração
 - Planejamento de integração
 - Progresso contínuo



Entendendo as fases do RUP Fase de Construção

- Desenvolver um produto pronto para a transição
 - Para ser implantado na comunidade do usuário final
 - Descrever os requisitos restantes
 - Completar o projeto
 - Projetar o banco de dados
 - Implementar código de testes de unidade
 - Integrar e testar o sistema
 - Preparar para a implantação

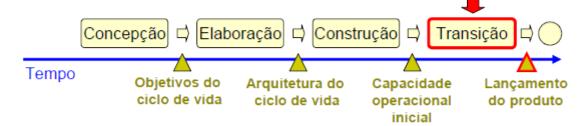


Entendendo as fases do RUP Fase de Construção

- Revisão do projeto: Marco de capacidade operacional inicial
 - O release do produto é maduro o suficiente para ser entregue ao usuário final?
 - Toda a equipe está preparada para a transição para a comunidade do usuário?
 - A relação gasto ocorrido x planejado é aceitável?
- Se falhar neste marco... nova iteração
 - O prejuízo de abortar já é absurdo... mas a possibilidade é mínima



- □ Fase de transição
 - Preparar empacotamento, produção e distribuição
 - Preparar local de implantação
 - Treinar usuários e mantenedores
 - Melhorar projetos futuros com lições aprendidas
 - □ Testes beta para avaliar aceitação do usuário





- Preparar empacotamento, produção e distribuição
 - Tornar o software um *produto*
 - Preparar o marketing do software
 - Preparar material auxiliar para venda
 - Distribuir o software para ser vendido



Entendendo as fases do RUP Fase de Transição

- Preparar local de implantação
 - Caso haja a troca de sistemas, a transição é mais complexa
 - Dados precisam ser transferidos dos sistemas antigos
 - Novos gastos podem ser necessários para implantar o sistema: desempenho, disponibilidade...



- Treinar usuários e mantenedores
 - Treinar todos os usuários, chefia operacional e equipes de manutenção apropriadamente
 - Treinamento diminui dependência da equipe de desenvolvimento
 - Usuários auto-sustentáveis



- Melhorar projetos futuros com lições aprendidas
 - Que refinamentos são indicados para o processo e as ferramentas?
 - Quais são as lições aprendidas?
 - Como melhor adaptar o RUP de acordo tais lições?



- Testes beta para avaliar aceitação do usuário
 - Testes de transição: ambiente alvo
 - Capturar, analisar e implementar requisições de mudanças
 - Patches



Entendendo as fases do RUP Fase de Transição

- Revisão do projeto: Marco de lançamento do produto
 - Os usuários finais estão satisfeitos?
 - A relação gasto ocorrido x planejado é aceitável? Se não, como melhorar no futuro para resolver o problema?
- Se falhar neste marco... nova iteração
 - Impossível abortar... já está pronto!