OpenUP: Processo Unificado Aberto

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Departamento Acadêmico de Gestão e Tecnologia da Informação
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Prof. Fellipe Aleixo (fellipe.aleixo@ifrn.edu.br)

O que é o OpenUP?

- OpenUP é um processo de desenvolvimento de software de código aberto projetado para equipes pequenas e centralizadas que querem ter uma abordagem ágil para desenvolvimento
- O OpenUP é um processo iterativo que é Mínimo, Completo e Extensível, valorizando a colaboração entre a equipe e os benefícios aos interessados ao invés da formalidade e entregáveis desnecessários
- Referências:
  - OpenUp – http://epf.eclipse.org/wikis/openup/
  - OpenUp/Basic – http://epf.eclipse.org/wikis/openuppt/

OpenUp e OpenUp/Basic

- OpenUp representa a família de *plugins* do processo unificado aberto – *Eclipse Proccess Framework*
- OpenUp/Basic é uma derivação simplificada do OpenUP voltada para equipes pequenas e centralizadas
- A principal fonte de informações será o site do OpenUP (em inglês)
  - O site do OpenUP/Basic (em português) pode ser usado em alguns casos para o esclarecimento de dúvidas

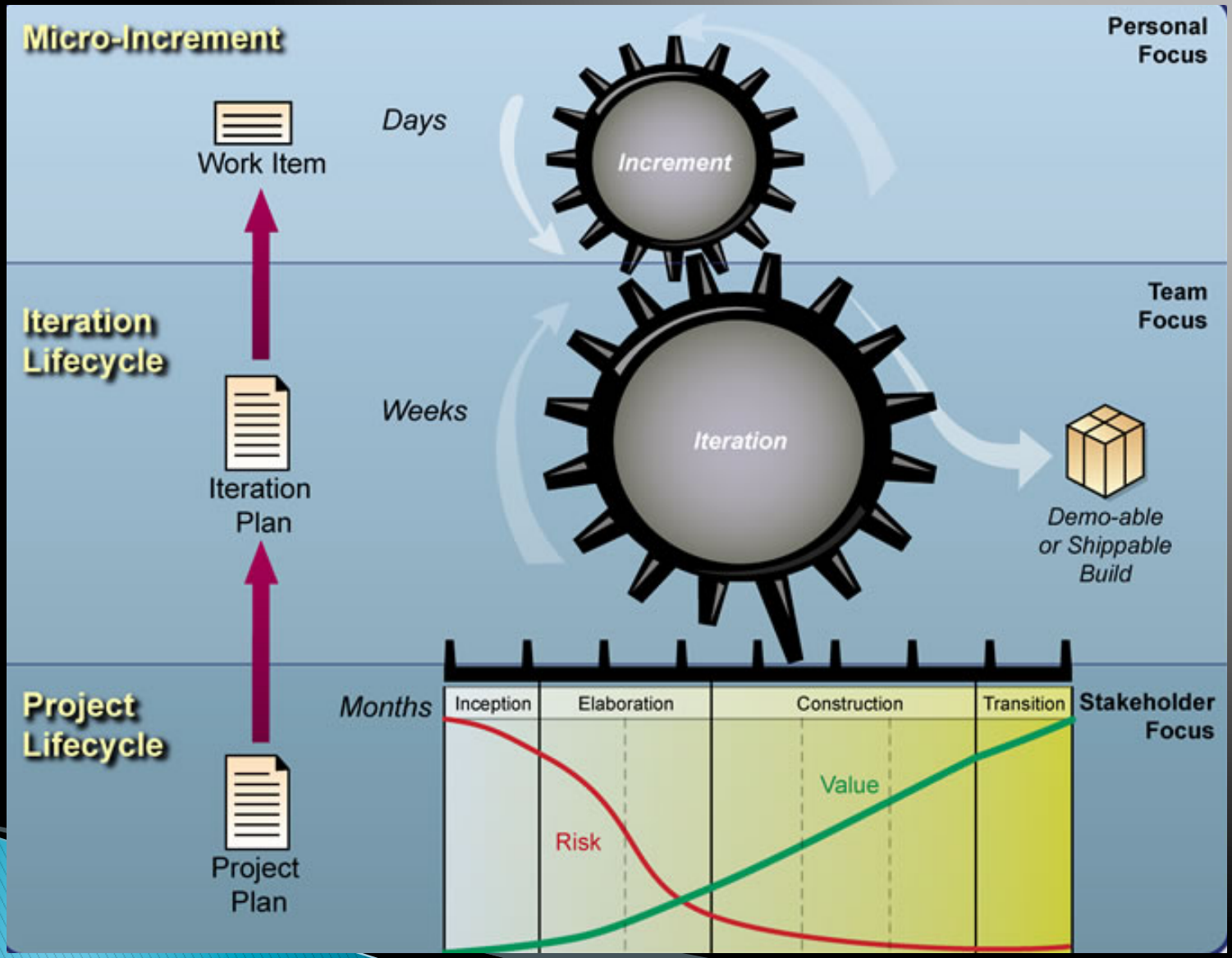
Princípios do OpenUP

- **Equilibrar as Prioridades concorrentes** para maximizar o valor para os *stakeholders*
- **Colaborar** para alinhar interesses e compartilhar entendimento
- **Focar na evidenciação da arquitetura**
  - Facilitar a colaboração técnica, reduzir o risco, e minimizar o sucateamento e o retrabalho.
- **Evoluir para continuamente obter feedback e melhorias**
  - Iterações curtas para antecipar o feedback

Quem Deve Usar?

- O OpenUP é mais útil para quatro grupos primários de usuários:
  - Profissionais de desenvolvimento de software (desenvolvedores, gerentes de projeto, analistas e testadores) que trabalham juntos como uma equipe de projeto
  - *Stakeholders* (interessados)
  - Engenheiros de processo de software
  - Instrutores

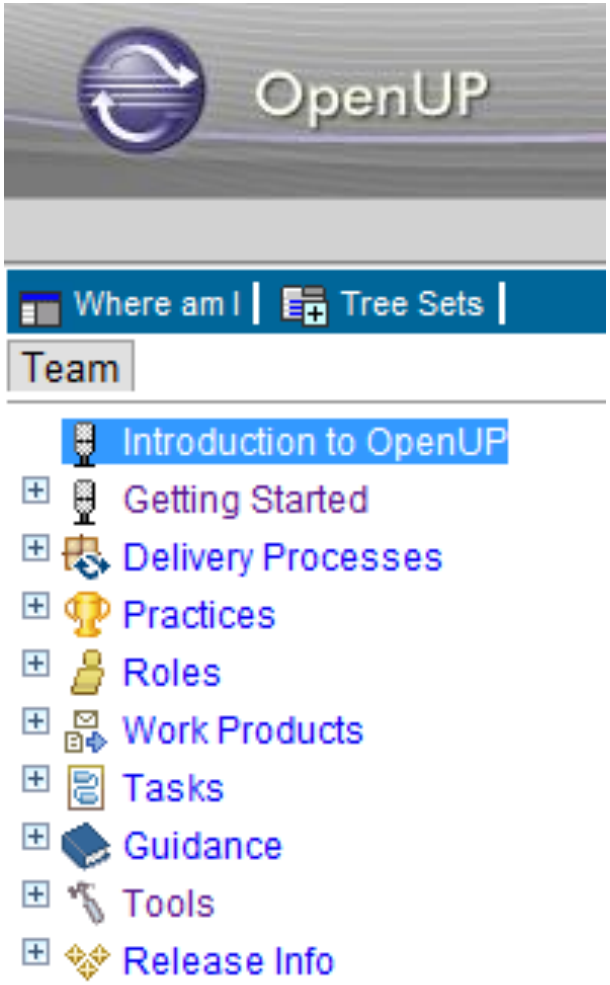
Visão Geral


Micro-Increment
Personal Focus
Work Item
Days
Increment
Iteration Lifecycle
Team Focus
Iteration Plan
Weeks
Iteration
Demo-able or Shippable Build
Project Lifecycle
Months
Inception
Elaboration
Construction
Transition
Stakeholder Focus
Value
Risk
Project Plan

Conceitos Básicos de Processo

- Os elementos do processo estão organizados no site em:
  - Processo – *Delivery Process*: usado para definir uma estrutura e um fluxo de trabalho
  - Práticas – Practices: práticas a serem incorporadas ao desenvolvimento
  - Papéis – *Roles*: quem executa o trabalho
  - Produtos de trabalho – *Work products*: (artefatos) resultados produzidos pela execução das tarefas
  - Tarefas – *Tasks*: ações a serem executadas
  - Guias – *Guidance*: templates, checklists, exemplos, orientações, conceitos, e assim por diante

Conceitos Básicos de Processo



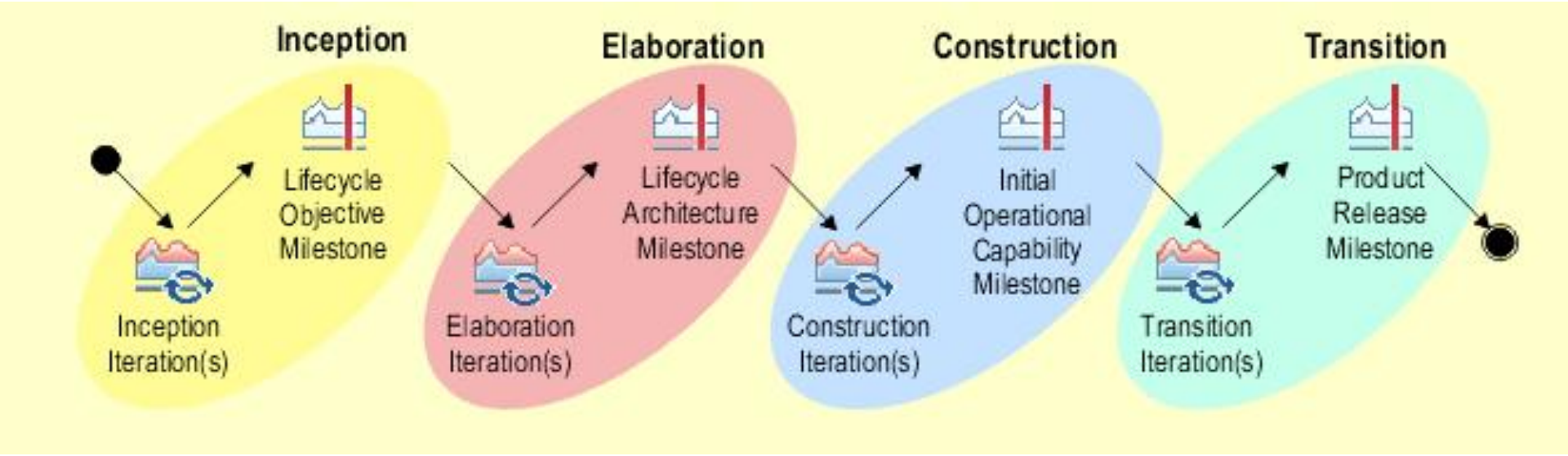
Glossário Básico

- Iteração
  - Uma iteração é um período de tempo definido dentro de um projeto em que você produz uma versão estável e executável do produto, junto com toda a documentação de apoio, scripts de instalação ou similares, necessários para usar a liberação
- Caso de Uso
  - Um caso de uso define uma seqüência de ações executadas pelo sistema que geram um resultado de valor observável para um ator em particular

Glossário Básico

- Risco
  - Na vida cotidiana um risco é uma exposição à perda ou dano; um fator, coisa, elemento ou um caminho que envolve perigo incerto. Similarmente, no desenvolvimento de software um risco é algo que pode comprometer o sucesso de um projeto
- Arquitetura de Software
  - A arquitetura de software representa a estrutura ou as estruturas do sistema, que consiste em componentes de software, propriedades externamente visíveis dos componentes e os relacionamentos entre eles

Ciclo de Vida



- Fases: Concepção (iniciação), Elaboração, Construção e Transição

Fases do OpenUP

- Uma visão geral das fases

Fase de Concepção

- O propósito desta fase é conseguir entendimento simultâneo entre todos os *stakeholders* dos objetivos de ciclo de vida para o projeto
- Objetivos:
  - Entenda o que construir. Determine a Visão, o escopo do sistemas, e seus limites. Identifique quem é *stakeholder* do sistema e porque
  - Identifique as funcionalidades chave do sistema. Decida quais requisitos são mais críticos
  - Determine pelo menos uma solução possível. Identifique pelo menos uma arquitetura candidata e sua aplicação prática
  - Entenda o custo, cronograma, e os riscos associados ao projeto

Fase de Concepção

- Os projetos podem ter uma ou mais iterações na fase de Concepção. Entre algumas razões para ter múltiplas iterações na fase de Concepção, você encontra:
  - O projeto é grande, e é difícil definir seu escopo
  - Sistema sem precedentes
  - Muitos *stakeholders* com necessidades conflitantes e relacionamentos complexos
  - Grandes riscos técnicos que demandam a criação de um protótipo ou prova de conceito

Fase de Elaboração

- O propósito desta fase é estabelecer uma linha de base da arquitetura do sistema e prover uma base estável para o volume de esforço de desenvolvimento na próxima fase
- Objetivos:
  - Obtenha um entendimento mais detalhado dos requisitos. Ter um bom entendimento dos principais requisitos permite a você criar um plano mais detalhado e obter comprometimento dos *stakeholders*. Tenha certeza de obter um entendimento detalhado dos requisitos mas críticos que serão validados pela arquitetura
  - Projete, implemente, valide, e estabeleça uma linha de base para a arquitetura. Projete, implemente, e teste um esqueleto da estrutura do sistema. Apesar da funcionalidade não estar completa ainda, a maior parte das interfaces entre os blocos sendo construídos é implementada e testada. Isto é conhecido como uma arquitetura executável
  - Mitigue os riscos essenciais e produza um cronograma e uma estimativa de custos precisos. Muitos riscos técnicos são resolvidos como resultado do detalhamento dos requisitos e do projeto, implementação e teste da arquitetura. Refine e detalhe o plano de projeto de alto nível

Fase de Elaboração

- O número de iterações na fase de Elaboração é dependente de, mas não limitada por, fatores como desenvolvimento de um novo produto versus ciclo de manutenção, sistema sem precedentes versus tecnologia e arquitetura conhecidas, e etc.
- Tipicamente, na primeira iteração, você deve projetar, implementar, e testar um pequeno número de cenários críticos para identificar que tipo de arquitetura e mecanismos de arquitetura você precisa, então você pode mitigar os riscos mais cruciais. Você também detalha os requisitos de alto risco que devem ser resolvidos antecipadamente no projeto. Você deve testar o suficiente para validar que os riscos arquiteturais estão mitigados
- Nas próximas iterações, você corrige o que não estiver correto na iteração anterior. Você projeta, implementa, e testa os cenários arquiteturalmente significantes que restaram, garantindo que você verifique todas as áreas principais do sistema (cobertura arquitetural), assim os riscos potenciais escondidos aparecem o mais cedo possível

Fase de Construção

- A finalidade desta fase é terminar o desenvolvimento do sistema baseado na arquitetura colocada na linha de base
- Objetivos:
  - Desenvolver de forma iterativa um produto completo que esteja pronto para ser entregue à comunidade de usuários. Descreva os requisitos restantes, preencha os detalhes do projeto, termine a implementação e teste o software. Libere a primeira versão operacional (beta) do sistema e determine se os usuários já estão prontos para que a aplicação possa ser implantada
  - Minimizar os custos de desenvolvimento e conseguir algum grau de paralelismo. Otimize os recursos e aumente o paralelismo de desenvolvimento entre os desenvolvedores ou as equipes de desenvolvimento, como por exemplo, atribuindo os componentes que podem ser desenvolvidos independentemente para desenvolvedores distintos

Fase de Construção

- Normalmente, a fase de Construção tem mais iterações (de duas a quatro) do que as outras fases, dependendo dos tipos de projetos:
  - Projeto simples: Uma iteração para construir o produto (para uma liberação beta)
  - Projeto mais substancial: Uma iteração para expor um sistema parcial e uma para amadurecê-lo para o teste beta
  - Projeto grande: Três ou mais iterações, dependendo do tamanho do projeto (quantidade de requisitos à implementar para uma liberação beta)

Fase de Transição

- A finalidade desta fase é assegurar que o software esteja pronto para ser entregue aos usuários
- Objetivos:
  - Executar o teste Beta para validar se as expectativas dos usuários foram atendidas. Isto normalmente requer algumas atividades de ajuste fino, tais como reparação de erros e melhorias no desempenho e na usabilidade
  - Obter a concordância dos *stakeholders* de que a distribuição está completa. Isto pode envolver vários níveis de testes para a aceitação do produto, incluindo testes formais, informais e testes beta
  - Melhorar o desempenho de projetos futuros com as lições aprendidas. Documente as lições aprendidas e melhore o ambiente de processos e ferramentas para o projeto

Fase de Transição

- A fase de Transição pode incluir a execução paralela de sistemas antigos e novos, migração de dados, treinamento de usuários e ajustes nos processos de negócio
- A quantidade de iterações na fase de Transição varia de uma iteração para um sistema simples que necessita primeiramente de reparos de pequenos erros, até muitas iterações para um sistema complexo, envolvendo a adição de características e a execução de atividades para fazer a transição, no negócio, do uso do sistema antigo para o sistema novo
- Quando os objetivos da fase de Transição são alcançados, o projeto está pronto para ser encerrado. Para alguns produtos, o fim do ciclo de vida atual do projeto pode coincidir com o começo do ciclo de vida seguinte, conduzindo à nova geração do mesmo produto

Disciplinas

- Agrupamentos de tarefas que compartilham de um mesmo propósito

(1) Arquitetura

- Esta disciplina explica como criar a arquitetura do sistema a partir dos requisitos significantes para a mesma
- A arquitetura é implementada na disciplinas de desenvolvimento
- Tarefas:
  - Definir o esboço da arquitetura
  - Refinar a arquitetura

(2) Desenvolvimento

- Esta disciplina explica como projetar e implementar uma solução técnica que esteja conforme a arquitetura e atenda aos requisitos
- Propósitos:
  - Transformar os requisitos em um projeto do que será o sistema
  - Adaptar o projeto para se adequar ao ambiente de implementação
  - Construir o sistema de forma incremental – gerando "*builds*"
  - Verificar se as unidades técnicas usadas para a construção do sistema trabalham de acordo como foram especificadas

(2) Desenvolvimento

- Tarefas:
  - Integrar e criar um build
  - Projetar a solução
  - Implementar os teste de desenvolvedor
  - Implementar a solução
  - Executar os testes de desenvolvedor

(3) Gerência de Projeto

- Esta disciplina explica como um "treinador" atua como facilitador e suporte para a ajudar a equipe a lidar com os riscos e obstáculos encontrados durante a construção do software
- O propósito desta disciplina é:
  1. Manter a equipe focalizada na entrega contínua do produto de software testado para a avaliação dos *Stakeholders*
  2. Ajudar a priorizar à seqüência de trabalho
  3. Ajudar a criar um ambiente de trabalho eficaz para maximizar a produtividade da equipe
  4. Manter os *Stakeholders* e a equipe informados sobre o progresso do projeto
  5. Fornecer uma estrutura para controlar o risco do projeto e para adaptar-se continuamente às mudanças

(3) Gerência de Projeto

- O gerenciamento de projeto age como uma ponte entre os *Stakeholders* e a equipe de desenvolvimento
- As atividades da gerenciamento de projeto devem adicionar valor ao criar um ambiente de trabalho de elevado desempenho onde
  1. Os *Stakeholders* tenham confiança na habilidade da equipe de conhecer as capacidades e restrições da plataforma técnica e de entregar com sucesso algo valoroso
  2. Os membros da equipe de projeto compreendam os desejos dos *Stakeholders* e confirmem que compreenderam, produzindo continuamente um produto de software para avaliação

(3) Gerência de Projeto

- O Gerente de Projeto trabalha com os *Stakeholders* para criar um Plano de Projeto inicial baseado na Visão. Este plano defini o tamanho e objetivo das quatro Fases e das Iterações de cada fase
- No começo de cada iteração, o gerente de projeto trabalha com os *Stakeholders* e com a equipe de desenvolvimento para priorizar os requisitos, as solicitações de mudança e os defeitos na Lista de Itens de Trabalho e aloca-los na iteração corrente
- O gerente de projeto trabalha então com a equipe de desenvolvimento para criar um Plano de Iteração mais refinado, baseado nos objetivos descritos no plano de projeto e nos itens de trabalho atribuídos à iteração
  - Os membros da equipe oferecem-se para executar os itens de trabalho alocados e fornecer ao gerente de projeto as tarefas e as estimativas de tempo necessárias para entregar tais itens

(3) Gerência de Projeto

- A equipe demonstra que entendeu os desejos dos *Stakeholders* durante cada iteração pela construção de um produto de software que é demonstrado aos *Stakeholders* para confirmar o entendimento e incentivar o feedback
- Ao final de cada iteração, a avaliação da Construção final deve incluir os resultados dos testes, deve ser registrada em uma Avaliação de Estado e deve ser comunicada a todos os *Stakeholders* e membros da equipe
- A equipe de desenvolvimento demonstra o progresso contínuo aos *Stakeholders* reportando os itens de trabalho terminados em cada iteração através do *Project Burndown*
  - A equipe pode usar o *Iteration Burndown* para demonstrar o progresso durante uma iteração

(3) Gerência de Projeto

- O gerenciamento de projeto precisa considerar as incertezas que o projeto enfrentará (isto é os Riscos) e trabalhar de forma proativa com os *Stakeholders* e a equipe para adaptar-se continuamente às mudanças nos requisitos do negócio, nos requisitos de sistema, e nas capacidades técnicas
- O gerenciamento de projeto é uma disciplina tipo guarda-chuva que impacta e é impactada por todas as outras disciplinas

(4) Requisitos

- Esta disciplina explica como elicitar, analisar, especificar, validar e gerenciar os requisitos para o sistema a ser desenvolvido
- O propósito desta disciplina é:
  - Entender o problema a ser resolvido
  - Entender as necessidades dos *Stakeholders*
  - Definir os requisitos para a solução
  - Definir os limites (escopo) do sistema
  - Identificar interfaces externas ao sistema
  - Identificar restrições técnicas na solução
  - Fornecer a base para o planejamento das iterações
  - Fornecer a base inicial para a estimativa de custo e cronograma

(4) Requisitos

- Para conseguir os objetivos, é importante compreender a definição e o escopo do problema que estamos tentando resolver
- Os *Stakeholders* são identificados e o problema a ser resolvido é definido
- Concordando com o problema a ser resolvido, os Requisitos para o sistema são eliciados, organizados, analisados, validados e especificados
- Durante todo o ciclo de vida, as mudanças nos requisitos são gerenciadas

(5) Teste

- Esta disciplina explica como prover um feedback sobre a maturidade do sistema através da avaliação do projeto, implementação, execução e dos resultados dos testes
- O propósito desta disciplina é:
  - Encontrar e documentar defeitos
  - Validar e provar as suposições feitas no projeto e requisitos especificados através de demonstrações concretas
  - Validar que o produto de software foi feito como projetado
  - Validar que os requisitos estão apropriadamente implementados

(5) Teste

- Um esforço de teste bom está baseado na filosofia de testes breves e testes freqüentes
- Orientações:
  - Como este software poderia quebrar?
  - Em que possíveis situações poderia estar este software para trabalhar de maneira previsível?
- Testar desafia as suposições, riscos e incertezas inerente no trabalho de outras disciplinas, e trata dessas preocupações que usam demonstração concreta e avaliação imparcial