

ALUNOS: TIAGO PEGORINI, JEAN RODRIGO SILVA, MAURICIO JUNIOR, GABRIEL GUSTAVO, LEONARDO DE FARIAS.

#### O que é OpenUP?

É baseado no Processo Unificado. Ele aplica uma abordagem iterativa e incremental dentro de um ciclo de vida estruturado e abraça uma filosofia pragmática e ágil que foca na natureza colaborativa do desenvolvimento de software. É um processo objetivo e independente de ferramenta, que pode ser estendido para direcionar uma grande variedade de projetos. É considerado um processo **Mínimo, Completo e Extensível**, valorizando a colaboração entre a equipe e os benefícios aos interessados, ao invés da formalidade e entrega de itens desnecessários (OpenUP, 2010). O processo pode ser facilmente entendido através das três camadas, **Microincrementos (Micro-increment)**, **Ciclo de Vida de Iteração (Iteration Lifecycle)**, **ciclo de vida do projeto (Project Lifecycle)**.

**Microincremento:** Poucas horas ou dias, progresso é visto diariamente e membros compartilham informações abertamente

**Ciclo de vida de iteração:** Planejadas em semanas e entrega incremental.

O esforço pessoal em um projeto OpenUP está organizado em **microincrementos**. Eles representam pequenas unidades de trabalho que produzem um passo do progresso do projeto, constante e mensurável (normalmente medido em horas ou dias). O processo aplica a colaboração intensiva à medida que o sistema é desenvolvido incrementalmente, por uma equipe comprometida e auto-organizada. Estes microincrementos fornecem um ciclo de *feedback* extremamente curto, que direciona decisões adaptativas durante cada iteração (OpenUP, 2010).

#### Similaridades e Diferenças com o RUP

- São métodos de desenvolvimento de software iterativo e incremental.
- O foco do RUP são requisitos Requisitos do Usuário(Casos de uso), Arquitetura e Fases, o OpenUp segue essa linha e adiciona diversas práticas dos métodos ágeis.
- O OpenUp é uma versão resumida do RUP, com características de métodos ágeis.
- 

#### Princípios do open UP:

Possui 4 princípios fundamentais, tendo grande importância para a interpretação dos conceitos de papéis, produtos de trabalho e as tarefas.

- **Balancear as prioridades concorrentes para maximizar os valores dos Stakeholders:**

É de extrema importância que todos os stakeholders tenham concordância sobre a solução proposta e o problema como um todo, levando em consideração o problema a ser resolvido, as restrições inerentes à equipe de desenvolvimento (custos, habilidades, recursos, etc.) e as restrições inerentes à solução proposta.

A definição do ponto de equilíbrio por ser um desafio constante, e existe um dinamismo que varia em conjunto com as características e requisitos da solução proposta, e isso implica também em mudanças nos riscos sobrepostos no processo de desenvolvimento. Durante o andamento do projeto, os desenvolvedores e os stakeholders devem estar preparados para reavaliar suas expectativas e ajustar seus planejamentos.

Existem boas práticas que podem ser utilizadas para facilitar algumas destas ações descritas:

- **Conhecer sua audiência:** tentar sempre manter um canal de comunicação entre a equipe de desenvolvimento e os stakeholders;
- **Separar o problema da solução:** garantir a compreensão dos problemas antes de mergulhar no desenvolvimento da solução, para facilitar a busca por soluções alternativas.
- **Documentar e Compartilhar a Compreensão do Domínio da Solução:** Os artefatos chaves para essa proposta são, Documento de Visão, Modelo de Caso de Uso, Especificação Suplementar e Glossário.
- **Utilizar Modelo de Casos de Uso Para Especificar os Requisitos Funcionais:** A aplicação de Modelos de Caso de Uso busca capturar os requisitos funcionais em um documento claro, escrito em uma linguagem própria do ambiente analisado.
- **Estabelecer e Manter a Concordância em Relação às Prioridades:** Priorizar requisitos para implementação através de atividades desenvolvidas em parceria com os Stakeholders pode garantir o sucesso do projeto.
- **Gerenciar o Escopo do Sistema:** O gerenciamento das mudanças solicitadas pode facilitar o processo de compreensão e concordância dos Stakeholders sobre os requisitos especificados.
- **Saiba Quando Parar:** Ao atingir todos os requisitos e a qualidade proposta, é importante parar o processo de desenvolvimento.

- **Colaborar para alinhar os interesses e compartilhar os conhecimentos:**

Promover um ambiente saudável para o processo de boas práticas tendem a alinhar os interesses dos participantes do projeto auxiliando a compartilhar conhecimentos adquiridos.

Sabe-se que um ambiente de desenvolvimento saudável possibilita uma colaboração efetiva entre os membros da equipe e permite aos participantes do projeto a criar uma visão compartilhada. Existem boas práticas que podem ser utilizadas:

- **Manter uma visão compartilhada dos conhecimentos:** por mais que o processo de desenvolvimento seja “enxuto” gera um banco de conhecimento sobre o processo de desenvolvimento e os membros da equipe e entre os demais Stakeholders.

Produto de trabalho da disciplina de Requisitos para manter o alinhamento entre os Stakeholders e a equipe de desenvolvimento.

- **Promover um ambiente de autoconfiança:** de um modo geral as pessoas não se sentem seguras no desempenho de suas atividades tendo dificuldades para expor suas ideias. O gerente de Projeto tem uma responsabilidade maior que é

acompanhar todas as atividades de forma minuciosa como gerenciar por objetivos, derrubar as barreiras, manter diálogos pertinentes, etc.

Comentado [TP1]:

- **Compartilhe Responsabilidades:** um membro que se isole dos demais gera uma desvantagem. Muitas pessoas tendem a se isolar não conseguindo se alinhar com os demais membros. A responsabilidade não é exclusiva de um único membro da equipe os membros mais experientes da equipe devem trabalhar em conjunto com os membros menos experientes.

- **Aprenda Continuadamente:**

Desenvolver continuamente tanto habilidades como técnicas é um fundamento importante buscando sempre aprender com os exemplos de seus colegas.

- **Gerenciar em Torno da Arquitetura:**

Da maneira que os projetos crescem aumenta também a comunicação entre os membros ao mesmo momento que há uma compreensão do sistema em um todo podem focar suas ações em um ou mais subsistemas na qual tem responsabilidade. mentais onde passam a visualizar -se tomar cuidado para que não aja casos isolados de modelos mentais onde passam a visualizar e compreender apenas funcionalidades nas quais estão focadas.

- **Focar inicialmente na arquitetura para minimizar riscos e organizar o desenvolvimento:**

Sem uma arquitetura bem especificada ou mal estruturada, o próprio desenvolvimento de um sistema pode se tornar ineficiente e com também haverá dificuldades nos processos de manutenção, reuso e integração.

E com uma arquitetura evoluída é possível identificar o nível de complexidade do sistema, auxiliando também na redução de riscos e no processo de organização do desenvolvimento.

A partir disso, existem boas práticas que podem ser utilizadas:

- **Elaborar uma Arquitetura Considerando seus Conhecimentos:** para promover flexibilidade e velocidade no processo de desenvolvimento, é necessário a criação de uma arquitetura que atenda somente os requisitos definidos.

- **Influência da Arquitetura Como uma Ferramenta Colaborativa:** Uma boa arquitetura facilita a colaboração e a sinergia entre os membros da equipe, criando um vocabulário comum e beneficiando a comunicação.

- **Tratar a Complexidade do Sistema Incrementando o Nível de Abstração do Projeto:** A elaboração da arquitetura prega o uso de modelos (Diagramas de Classe, Padrão MVC, Design Patterns, etc.) para incrementar o nível de abstração do projeto.

- **Organize a Arquitetura em Componentes Coesos e Fracamente Acoplados:** Essa ação permite aumentar a compreensão e incrementar a flexibilidade e o reuso.

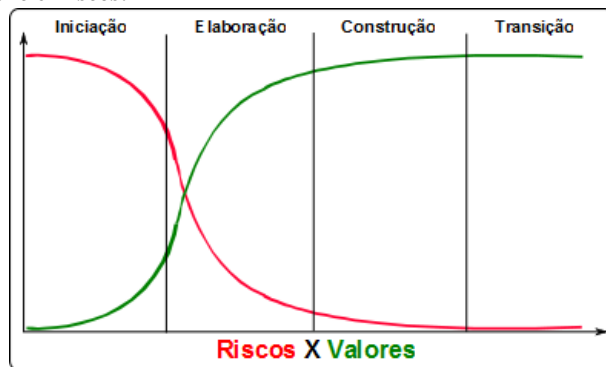
- **Reutilize Componentes Existentes:** Já é mais do que posto e sabido que o reuso de componentes reduz o custo e o cronograma do processo de desenvolvimento de software.

- **Envolver os Stakeholders para obter contínuo feedback do desenvolvimento:**

É de extrema importância que a equipe de desenvolvedores promova práticas que permitam demonstrar a agregação incremental de valor ao produto e obter feedback contínuo dos stakeholders, para evitar que as necessidades deixem de ser atendidas. Quando se oferece uma estrutura estável e confiável de desenvolvimento que possibilite o feedback contínuo, as mudanças ocorrem de forma menos traumática e se acomodam mais facilmente.

A partir disso, existem boas práticas que podem ser utilizadas:

- **Desenvolva seus Projetos de Forma Iterativa:** Dividir seu processo de desenvolvimento em pequenas etapas cíclicas (iterações) habilita sua equipe a incrementar a capacidade de entrega de produtos de trabalho continuamente.
- **Foque as Iterações de Forma a Atingir os Próximos Marcos:** as ações devem ser focadas de forma a tornar visível todos os pontos fundamentais do projeto.
- **Gerencie Riscos:**



- **Gerencie Mudanças:** Os Stakeholders também devem ter plena consciência de que mudanças geram custos. Mais uma vez, destaca-se a importância do processo de análise de requisitos.

- **Mensure o Progresso do Projeto de Forma Objetiva:** Uma forma objetiva de se avaliar o progresso de um projeto é avaliar a quantidade, qualidade e estabilidade dos Produtos de Trabalho que são entregues.

#### Relação dos princípios do OpenUP com o manifesto ágil

Um dos princípios do openup que visa ser dirigido pelos casos de usos, mostra semelhança com outros princípios muito importantes do manifesto ágil como:

- Software em funcionamento mais que documentação abrangente, no sentido de incentivar a produção de software pelas iterações;
- Responder as mudanças mais que seguir um plano, essa ideia se alinha também com essa direção por casos de uso.
- OpenUP: "Colaborar para alinhar interesses e compartilhar entendimento", Manifesto Ágil: "Indivíduos e interações e processos e ferramentas"
- OpenUP: "Balancera as prioridades para maximizar o valor dos interessados", Manifesto Ágil: "Colaboração do cliente à negociação de contrato"

- OpenUP: “Foco cedo na arquitetura para minimizar riscos e organizar o desenvolvimento, Manifesto Ágil: “Software funcionando à documentação compreensível”
- OpenUP: “Evoluir continuamente para ganhar feedback do cliente”, Manifesto Ágil: “Reponder à mudança ao invés de seguir um plano”

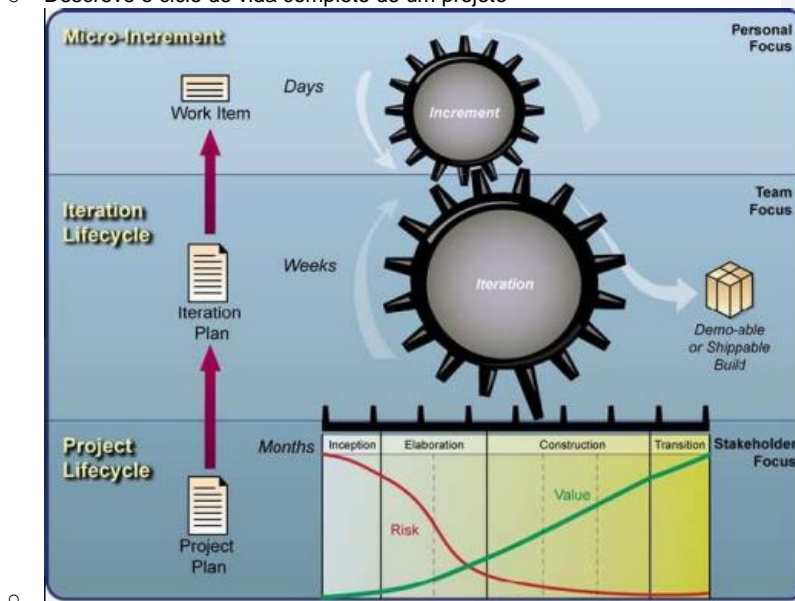
#### Dimensões: conteúdo de métodos e conteúdo de processos

##### Metodologia:

- Provê explicações passo a passo
- Descreve como alcançar o objetivo do desenvolvimento

##### Processo:

- Organiza e relaciona os elementos da metodologia em uma sequência semi-ordenada.
- Padrões de Capacidade:
  - Pedacos reusáveis do processo
  - Pequenos e focados
  - Blocos básicos
- Processo de Entrega
  - Sequência de Iteration Template Patterns
  - Descreve o ciclo de vida completo de um projeto



#### Disciplinas, Papéis, tarefas, artefatos e orientação

**Artefato:** Produto de Trabalho, possui 17 artefatos: Considerados essenciais e não há obrigação pelo uso de artefatos formais.

**Tarefa:** Como executar o trabalho, são claras e focadas em resultados, o texto normalmente é curto e objetivo, descrevendo como as pessoas devem colaborar para alcançar os objetivos. Os passos são descrições resumidas do que deve ser realizado, e apontam para orientações externas a respeito de como fazê-los. Possui 18 tarefas: Executores primários e adicionais.

**Papel:** Quem executa o trabalho, possui 7 papéis:

- Stakeholder
- Analista
- Arquiteto
- Desenvolvedor
- Testador
- Gerente de Projeto
- Qualquer Papel

**Disciplina:** É uma coleção de tarefas, define 6 disciplinas:

- Requisitos
- Arquitetura
- Desenvolvimento
- Teste
- Gerenciamento de Projeto e Configuração
- Gerenciamento de mudanças

**Orientação:** São usados para definir os fluxos de trabalho.

#### Fases (principais atividades e objetivos)

O OpenUP estrutura o **ciclo de vida do projeto** em quatro fases:

- **Concepção:** primeira fase do processo, onde os *stakeholders* e os membros da equipe colaboram para determinar o escopo e os objetivos do projeto, e determinar se o projeto deve ou não continuar.
- **Elaboração:** segunda das quatro fases no ciclo de vida do projeto, quando riscos arquiteturais significantes são tratados.
- **Construção:** terceira fase do processo, a qual foca no detalhamento dos requisitos, na implementação e nos testes da maior parte do software.
- **Transição:** quarta fase do processo, focada na transição do software para o ambiente do cliente e na obtenção da concordância dos *stakeholders* de que o desenvolvimento do produto está completo

#### Padrões de capacidade e processos de entrega

Os padrões de capacidade são padrões pré-definidos de processos.

Este processo de entrega define um processo de desenvolvimento de software completo que suporta os princípios fundamentais do OpenUP. Foi projetado para suportar equipes pequenas e co-localizadas, em suas atividades diárias.

#### Customização (adaptação ou tailoring)

O OpenUp é um tailoring do RUP, porem misturando com agile, Scrum e XP.

<https://www.tiespecialistas.com.br/openup-uma-visao-geral/>

<http://diego-pacheco.blogspot.com/2009/03/as-diferencas-do-openup-para-o-rup.html>

<https://medium.com/@LucianeS/openup-um-processo-integrado-e-agil-a4400c17ce62>

[http://www.hectordufau.com.br/openup/openup/guidances/supportingmaterials/minimal\\_complete\\_extensible\\_3AF278BB.html](http://www.hectordufau.com.br/openup/openup/guidances/supportingmaterials/minimal_complete_extensible_3AF278BB.html)

[http://www.cin.ufpe.br/~processos/TAES3/slides-2007.2/OpenUP\\_e\\_EPF-aaca.pdf](http://www.cin.ufpe.br/~processos/TAES3/slides-2007.2/OpenUP_e_EPF-aaca.pdf)

<https://o.aolcdn.com/images/dims3/GLOB/crop/2693x1350+0+130/resize/630x315!/format/jpg/quality/85/http%3A%2F%2Fo.aolcdn.com%2Fhss%2Fstorage%2Fmidas%2F7a8d8270cbd79e98dabb203ce9195a2a%2F206694451%2FRTS22ZT2.jpeg>

<https://en.wikipedia.org/wiki/OpenUP>

<http://open2up.blogspot.com/p/principios-basicos-do-open-up.html>