

OpenUP e Eclipse Process Framework

André Aziz (aaca@cin.ufpe.br)

Agenda

- OpenUP:
 - O que é?
 - Princípios e Conceitos;
 - Arquitetura;
- Eclipse Process Framework (EPF):
 - O que é?
 - EPF Composer.

Open Unified Process

O que é OpenUP?

- Processo de desenvolvimento de Software;
- Gratuito e *Open-source*;
- Processo ágil.



O que é OpenUP?

- Características essenciais do Processo Unificado:
 - Abordagem iterativa e incremental;
 - Baseado em casos de uso e cenários;
 - Gerenciamento de riscos;
 - Centrado na arquitetura.

O que é OpenUP?

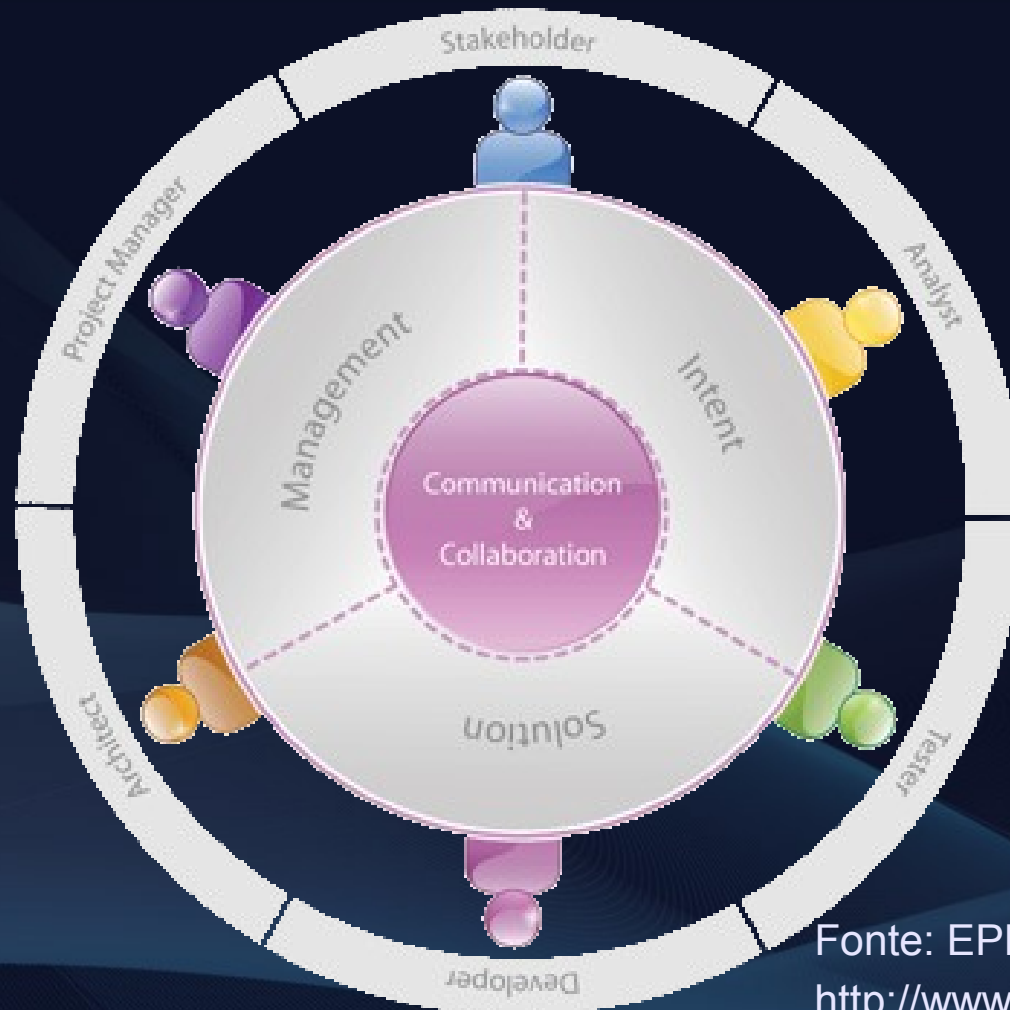
- Mínimo, completo e extensível:
 - Equipes pequenas e co-localizadas;
 - Pode ser utilizado como está;
 - Personalizável para atender propósitos específicos.

Quem Participa?



Fonte: NumberSix

Princípios



Fonte: EPF Wiki
<http://www.epfwiki.net/wikis/openuppt>

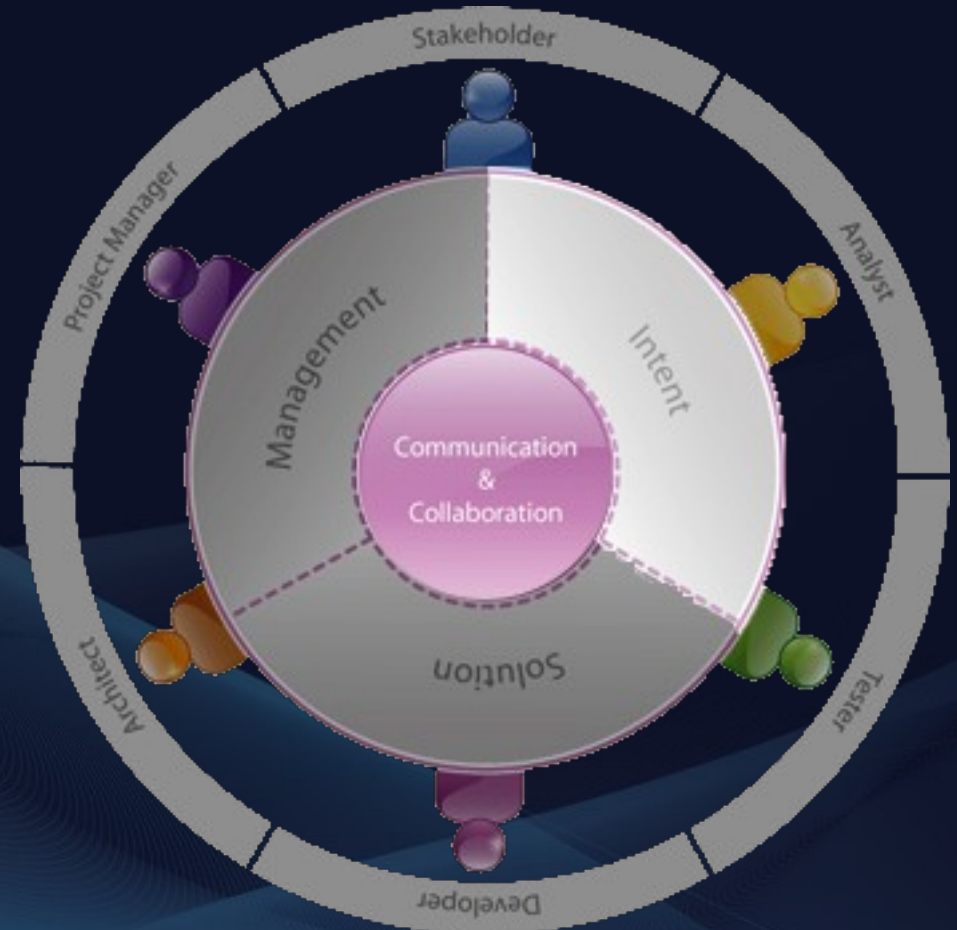
Princípios

- Colaborar para alinhar interesses e compartilhar entendimento.



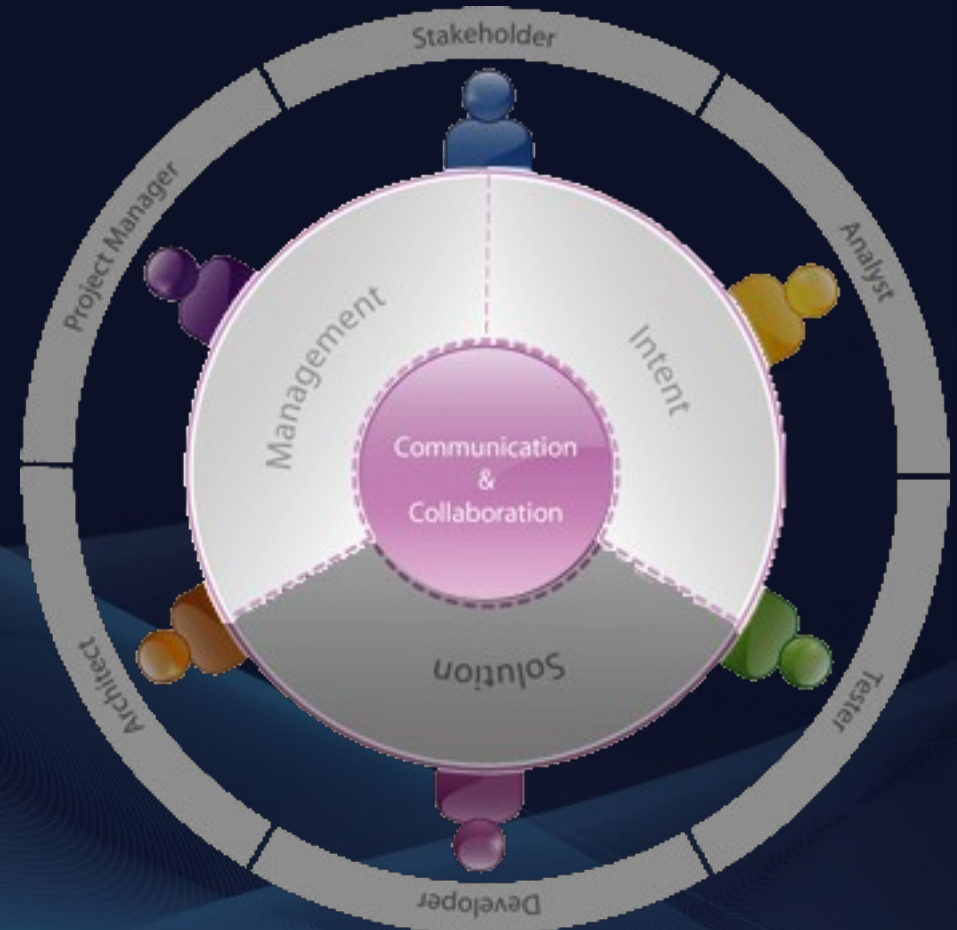
Princípios

- Balancear as prioridades (necessidades e custos técnicos) para maximizar o valor dos interessados.



Princípios

- Focar na articulação da arquitetura para facilitar a colaboração técnica, reduzir o risco, e minimizar o sucateamento e o retrabalho.



Princípios

- **Evoluir** continuamente para reduzir riscos, demonstrar resultados, e ganhar feedback do cliente.



OpenUP x Manifesto Ágil

- OpenUP:
 - *“Colaborar para alinhar interesses e compartilhar entendimento”;*
- Manifesto Ágil:
 - *“Indivíduos e interações a processos e ferramentas”.*

OpenUP x Manifesto Ágil

- OpenUP:
 - *“Balancear as prioridades para maximizar o valor dos interessados”;*
- Manifesto Ágil:
 - *“Colaboração do cliente à negociação de contrato”.*

OpenUP x Manifesto Ágil

- OpenUP:
 - *“Foco cedo na arquitetura para minimizar riscos e organizar o desenvolvimento”;*
- Manifesto Ágil:
 - *“Software funcionando à documentação compreensível”.*

OpenUP x Manifesto Ágil

- OpenUP:
 - *“Evoluir continuamente para ganhar feedback do cliente”;*
- Manifesto Ágil:
 - *“Responder à mudança ao invés de seguir um plano”.*

Público Alvo

- Profissionais de desenvolvimento de software;
- *Stakeholders*;
- Engenheiros de processo de software;
- Instrutores.

Organização

- Divididas em duas dimensões correlatas:
 - Metodologia:
 - Criada separadamente
 - Papéis, artefatos, tarefas e guias;
 - Processo:
 - Aplicação dos elementos da metodologia no tempo;

Organização

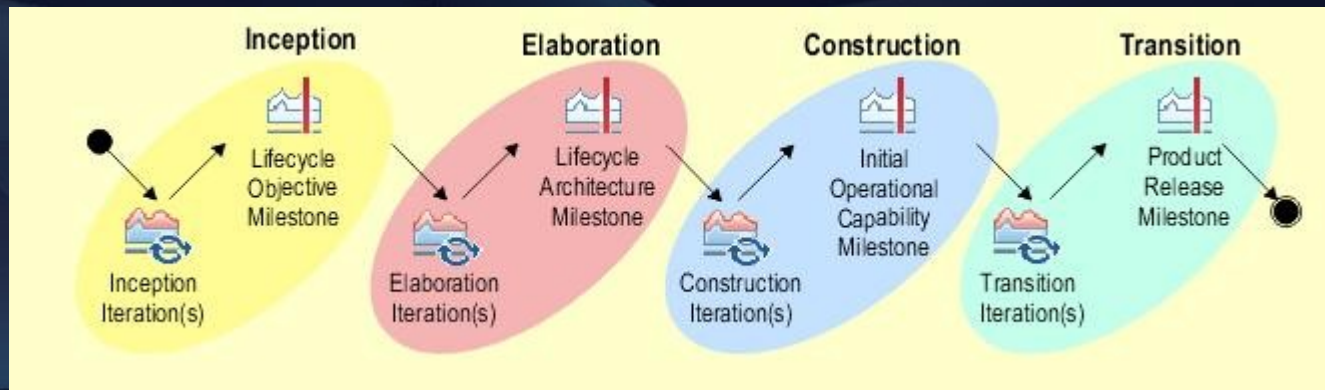
- Metodologia:
 - Provê explicações passo-a-passo;
 - Descreve como alcançar o objetivo do desenvolvimento;
- Processo:
 - Organiza e relaciona os elementos da metodologia em uma sequência semi-ordenada.

Organização

- Processo (cont.):
 - *Capability Patterns*:
 - Pedacos resuávies do processo;
 - Pequenos e focados;
 - blocos básicos.

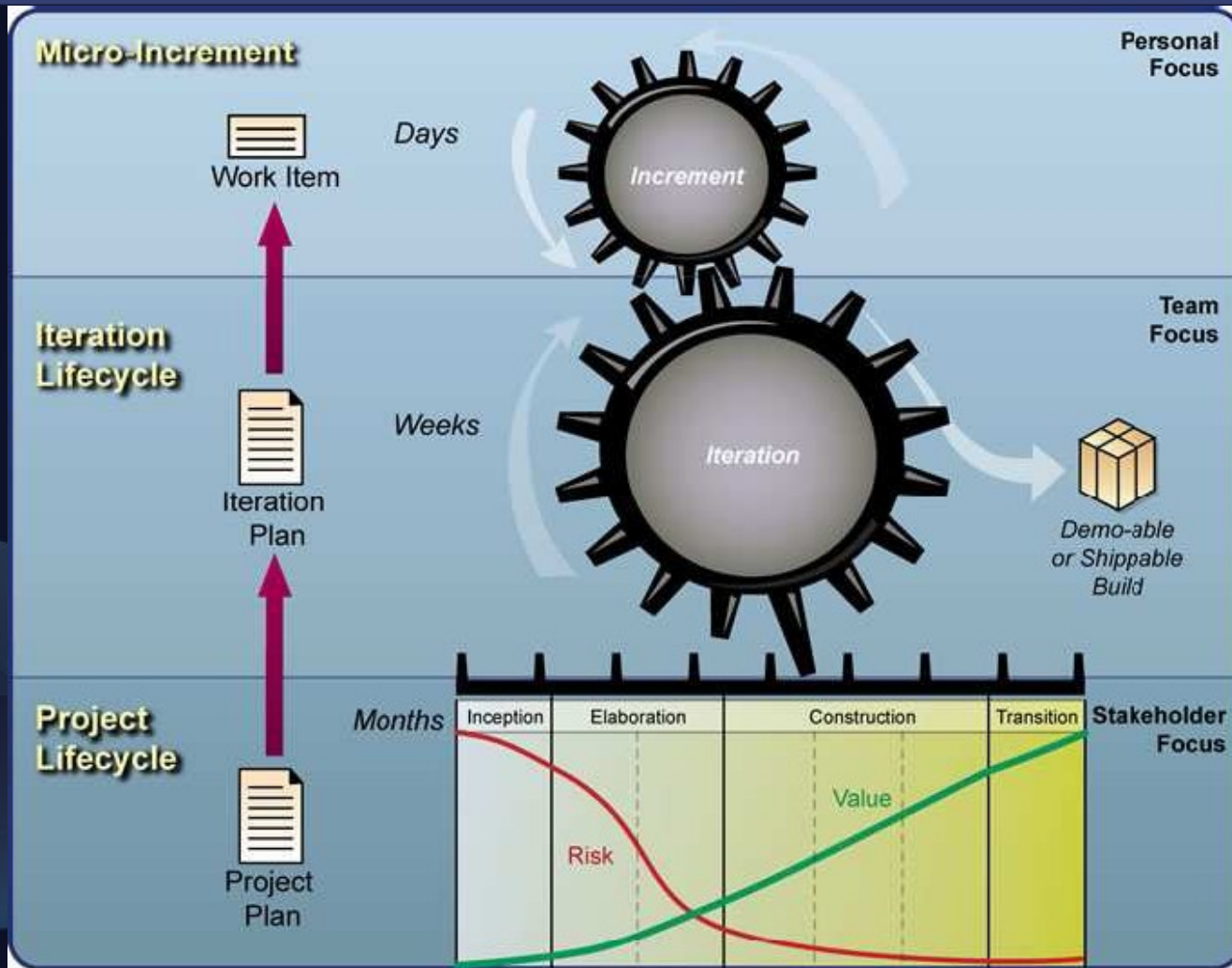
Organização

- Processo (cont.):
 - *Delivery Process*:
 - Sequência de *Iteration Template Patterns*:
 - Descreve o ciclo de vida completo de um projeto.



Fonte: EPF Wiki - <http://www.epfwiki.net/wikis/openuppt/>

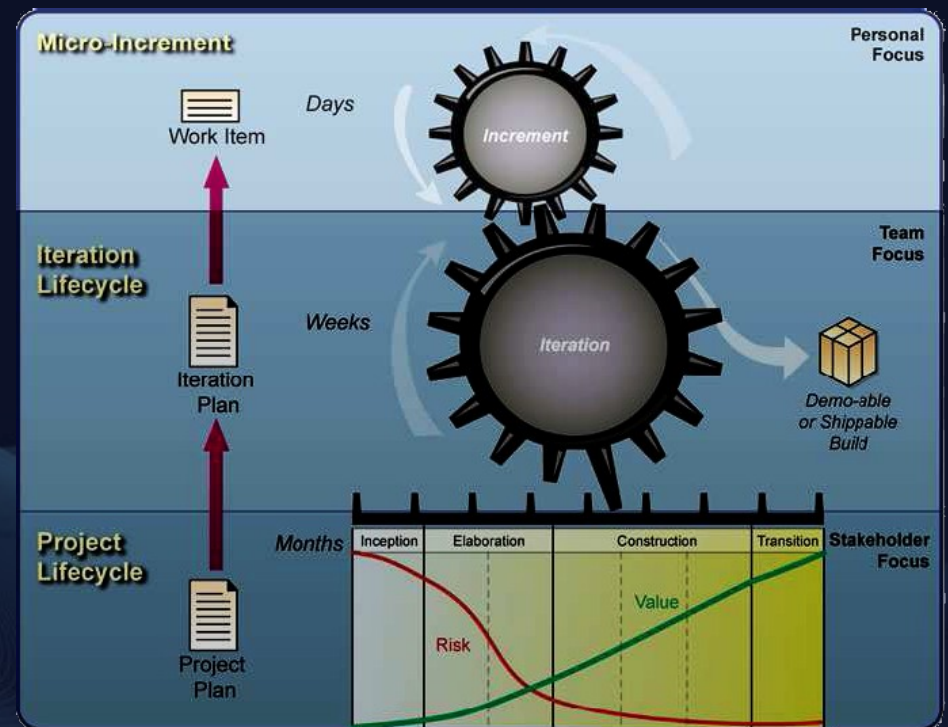
Ciclo de Vida



Fonte: Introduction
to OpenUP –
Ricardo Balduino

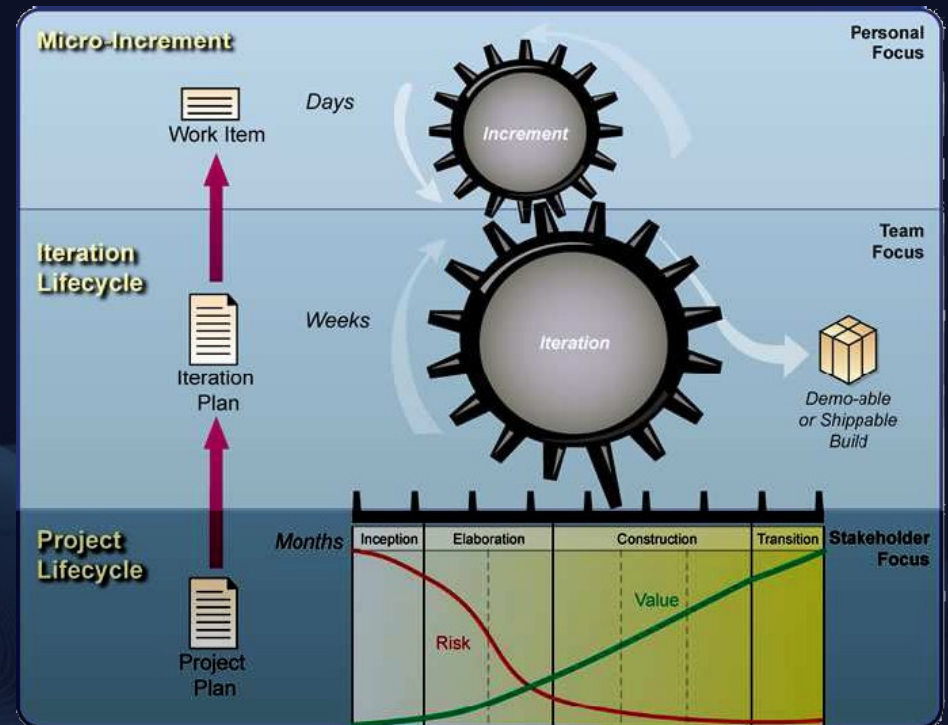
Micro-Increment

- Poucas horas ou dias;
- Progresso é visto diariamente;
- Membros compartilham informações abertamente;



Iteration Lifecycle

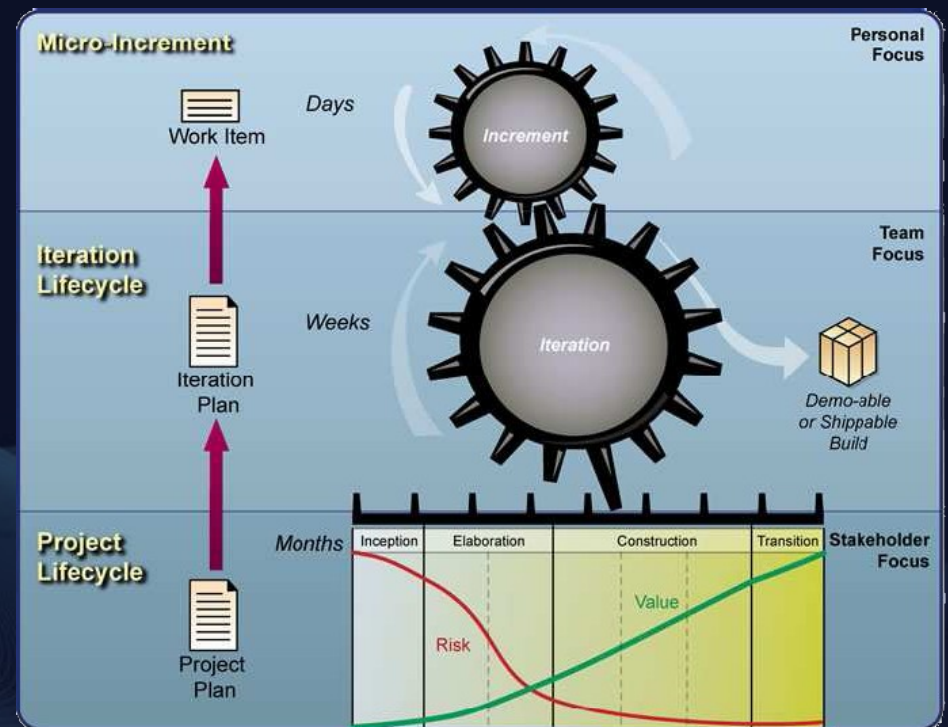
- Planejadas
 - Semanas;
- Entrega incremental;



Project Lifecycle

- Quatro fases:

- Concepção;
- Elaboração;
- Construção;
- Transição.



Papéis

- OpenUP define 7 papéis:
 - *Stakeholder*;
 - Analista;
 - Arquiteto;
 - Desenvolvedor;
 - Testador;
 - Gerente de Projeto;
 - Qualquer Papel.



Disciplinas

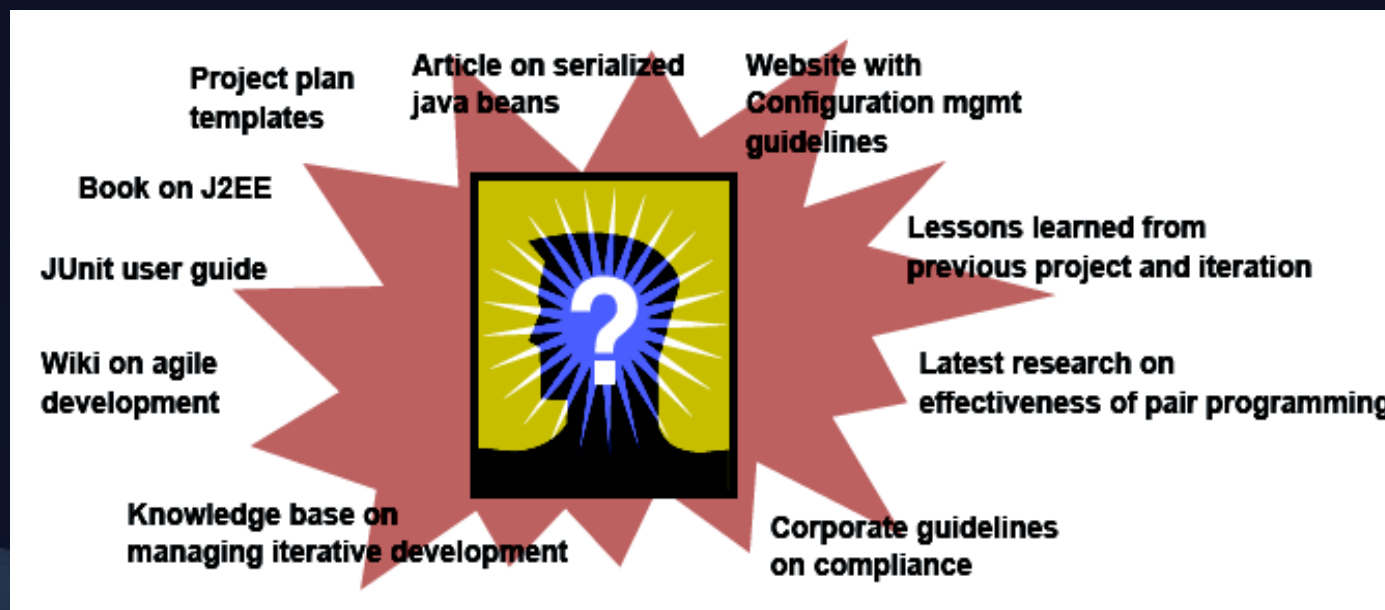
- Define 6 disciplinas:
 - Requisitos, Arquitetura, Desenvolvimento e Teste;
 - Gerenciamento de Projeto e Configuração & Gerenciamento de Mudanças;
- Demais disciplinas do UP são consideradas desnecessárias.

Tarefas e Artefatos

- Possui 18 tarefas:
 - Executores primários e adicionais;
- Possui 17 artefatos:
 - Considerados essenciais;
 - Não há obrigação pelo uso de artefatos formais.

Eclipse Process Framework

Quais são os desafios hoje?



Fonte:
NumberSix

- Falta de linguagem comum;
- Dificuldades de customização;
- Falta de um framework de comunicação.

Abordagem do EPF

Metodologia



Processo



Padronização da representação e gerenciamento de bibliotecas

Desenvolver e gerenciar processos para execução dos projetos

Configurar um processo coeso ajustado ao projeto

Criar templates para o plano de projeto para casar as definições do processo com as necessidades do projeto

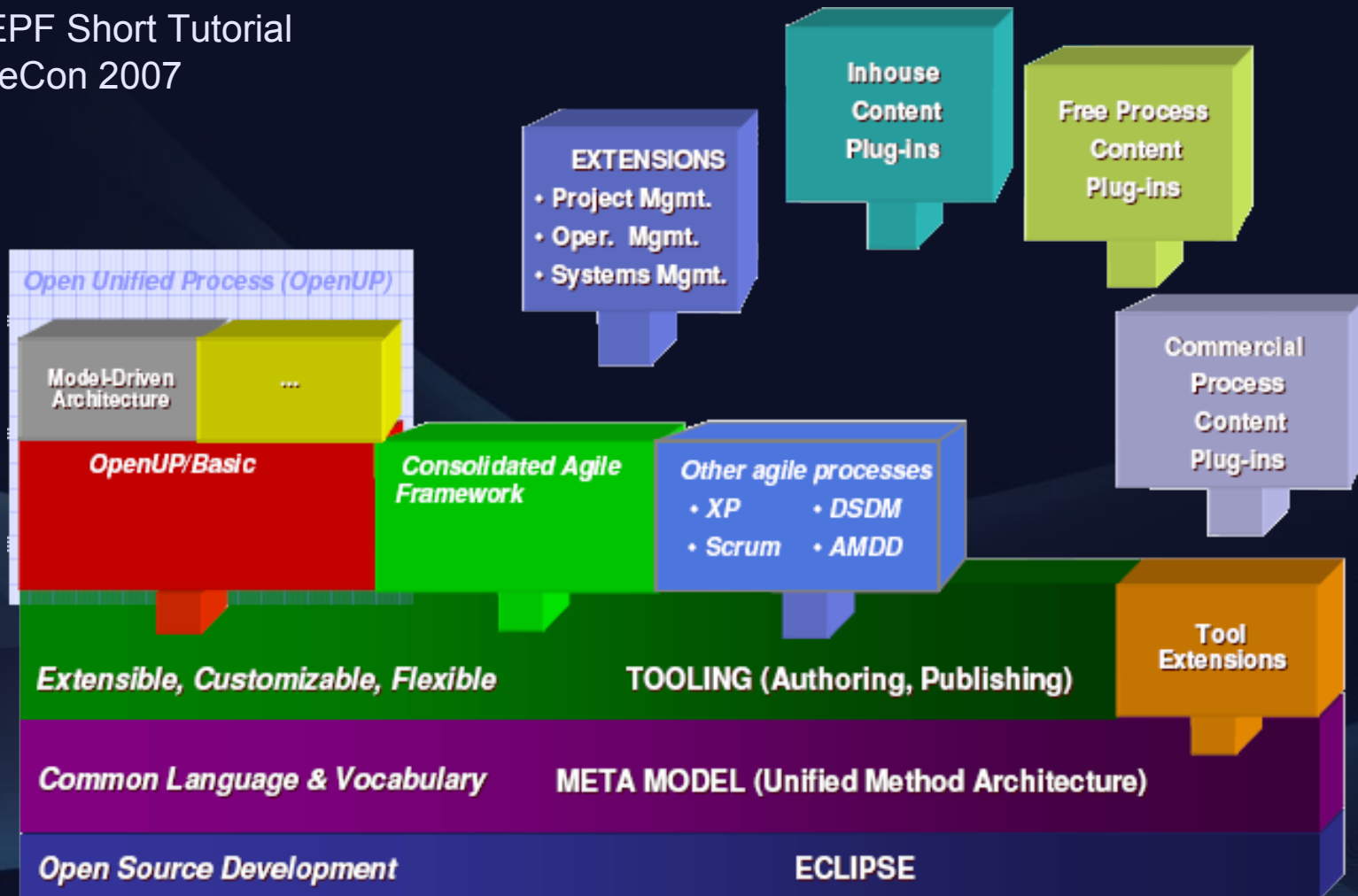
Fonte: An Overview to EPF Composer – Part 1

EPF Project

- Provê um ecossistema aberto e colaborativo para evolução dos processos de desenvolvimento de software;
- Disponibiliza práticas, processos e metamodelos;
- Usa a comunidade Eclipse para ganhar grande aceitação.

Ecosystema EPF

Fonte: EPF Short Tutorial
at EclipseCon 2007



EPF Composer

- Ferramenta para elaboração, manutenção e distribuição de processos de desenvolvimento;
- Baseada na plataforma Eclipse;
- Utiliza SPEM.

EPF Composer

- Propósitos:
 - Prover uma base de conhecimento à equipe de desenvolvimento de modo que eles possam buscar, gerenciar e distribuir o conteúdo;
 - Auxiliar engenheiros de processo e gerentes de projeto na seleção, extensão e implantação de processos para os seus projetos reais.

EPF Composer

- Projetado para ser um gerenciador de conteúdo:
 - Gerenciamento de documentos;
 - Processos pré-definidos;
 - *Capability Patterns*;
 - Material armazenado pode ser publicado e distribuído como Web Sites.

EPF Composer

The screenshot shows the OpenUP/Basic website in a Mozilla Firefox browser window. The address bar shows the file path: file:///C:/Home/Rational%20RUP%20team/RUPs/OpenUP_Basic-0.9-W-200703. The website has a blue header with the title "OpenUP/Basic" and links for "Glossary", "Feedback", and "About". A "Print" button is also visible. On the left, a "Tree Sets" sidebar lists the site's structure, including "Introduction to OpenUP/Basic", "Getting Started", "OpenUP/Basic Disciplines", "OpenUP/Basic Work Products", "Test", "OpenUP/Basic Roles", "OpenUP/Basic Lifecycle", and "About". The main content area is titled "Introduction to OpenUP/Basic" and features a "Contents" section with a bulleted list of topics: OpenUP and OpenUP/Basic, Minimal, Complete, and Extensible, Agile and Unified, Core Principles, and OpenUP Fundamental Concepts. Below this is a "Main Description" section with icons for "Getting Started", "Core Principles", "Roles", "Work Products", "Disciplines", and "Lifecycle". The "What is OpenUP/Basic?" section describes it as an open source software development process designed for small, co-located teams. It lists four core principles: Collaborate to align interests and share understanding, Balance competing priorities (needs and technical costs) to maximize stakeholder value, Focus on articulating the architecture to facilitate technical collaboration, reduce risk, and minimize scrap and rework, and Evolve continuously to reduce risk, demonstrate results, and gain feedback from the customer. The page concludes by stating that OpenUP/Basic is ready to use as-is and can be extended in large and small ways.

OpenUP/Basic - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/Home/Rational%20RUP%20team/RUPs/OpenUP_Basic-0.9-W-200703

IBM developerWorks

OpenUP/Basic

Glossary Feedback About

Print

Where am I | Tree Sets

OpenUP/Basic

- Introduction to OpenUP/Basic
- Getting Started
 - OpenUP/Basic Roadmap
 - Resources for Modifying Me
 - Resources for Contributing
- OpenUP/Basic Disciplines
- OpenUP/Basic Work Products
 - Architecture
 - Development
 - Project Management
 - Requirements
 - Glossary
 - Supporting Requirement
 - Use Case
 - Use-Case Model
 - Vision
 - Test
- OpenUP/Basic Roles
- OpenUP/Basic Lifecycle
 - Inception Iteration [1..n]
 - Elaboration Iteration [1..n]
 - Construction Iteration [1..n]
 - Transition Iteration [1..n]
 - Guidance
- About
- OpenUP Copyright

Introduction to OpenUP/Basic

Expand All Sections Collapse All Sections

Relationships

Contents	- OpenUP and OpenUP/Basic - Minimal, Complete, and Extensible - Agile and Unified - Core Principles - OpenUP Fundamental Concepts
----------	---

Back to top

Main Description

 Getting Started

 Core Principles

 Roles

 Work Products

 Disciplines

 Lifecycle

What is OpenUP/Basic?

OpenUP/Basic is an open source software development process designed for small, co-located teams who want to take an agile approach to development. OpenUP/Basic is an iterative process that is Minimal, Complete, and Extensible. It values collaboration and stakeholder value over unnecessary deliverables and formality.

OpenUP is characterized by four mutually supporting core principles:

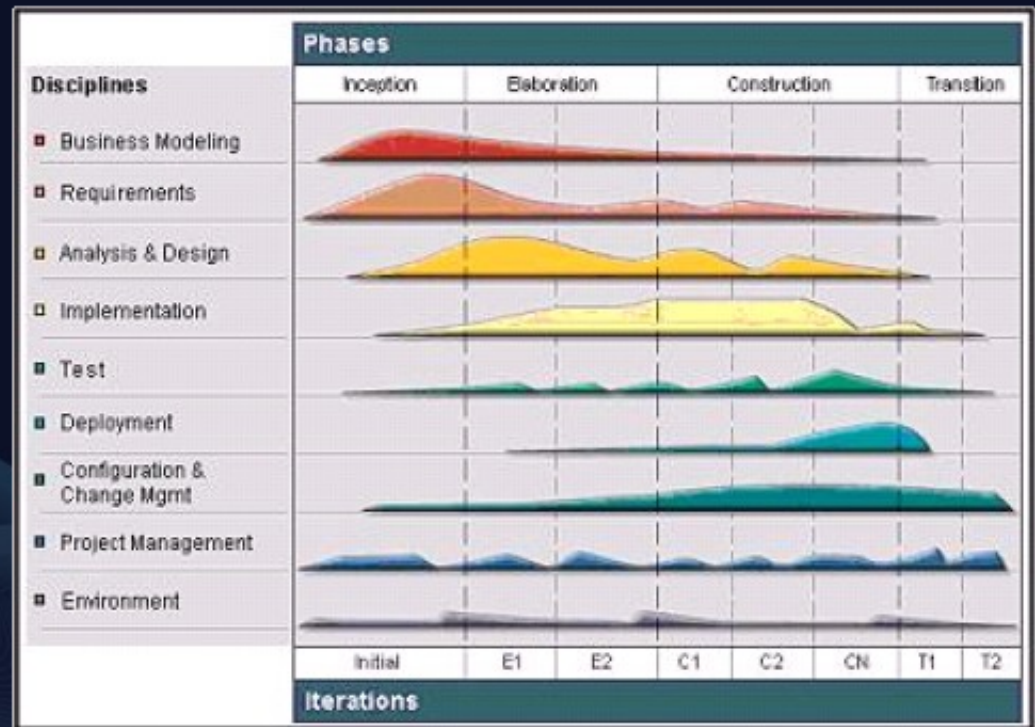
- Collaborate to align interests and share understanding
- Balance competing priorities (needs and technical costs) to maximize stakeholder value
- Focus on articulating the architecture to facilitate technical collaboration, reduce risk, and minimize scrap and rework.
- Evolve continuously to reduce risk, demonstrate results, and gain feedback from the customer

OpenUP/Basic is ready to use as-is; nothing needs to be added or removed. It can also be extended in large and small ways

Terminologia e Conceitos

- EPF Composer é baseado na separação entre metodologia e processo;

Fonte: An Overview to EPF Composer – Part 1



Metodologia

- Normalmente descrita em livros, artigos, padrões e regulações;
 - Documentada de diferentes maneiras;
- EPF Composer permite a estruturação deste material:
 - Unified Method Architecture (UMA).

Metodologia



IC / Best Practices



Books / Publications



Standards / Regulations



Homegrown Methods

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastructure\Samples\rup_20050708-playing

File Edit Search Configuration Window Help

config_for_PH_basic

Configuration

Content

Task: Detail a Use Case

Discipline: Requirements

Expand All Sections Collapse All Sections

Purpose

- To describe one or more of the use case's flow of events in sufficient detail to enable software development to begin on it.
- To describe the use case specification to the understanding and satisfaction of the actor representative or customer.

Back to top

Relationships

Roles	Primary Performer:	Additional Performers:
	Requirements Specifier	
Inputs	Mandatory:	Optional:
	- Use Case - Iteration Plan	- Glossary - Stakeholder Requests - Use-Case Model - Supplementary Specifications - Requirements Management Plan - Vision - Storyboard
Outputs	- Use Case - Supplementary Specifications	

Back to top

Steps

Expand All Steps Collapse All Steps

Review and Refine the Scenarios

Start by reviewing and refining the scenarios that you will be dealing with in the development cycle. These may have already been initially identified in the Task: Find Actors and Use Cases. Use these enumerated scenarios as a starting point in determining the scope of what flows will need to be described.

Storyboards will help you in understanding and detailing the use case flows. Another input to consider is the User-Interface Prototype, if one has already been developed.

Detail the Flow of Events

Structure the Flow of Events

Illustrate Relationships with Actors and Other Use Cases

Describe any Special Requirements

Requirements

- Capture a Common Vocabulary
- Detail a Use Case
- Detail the Software Requirements
- Develop Requirements Management
- Develop Vision
- Elicit Stakeholder Requests
- Find Actors and Use Cases
- Manage Dependencies
- Prioritize Use Cases
- Review Requirements
- Structure the Use-Case Model

Test

- Uncategorized Tasks

Domains

- Work Product Kinds
- Code
- Deliverables
- Document
- Environment Infrastructure
- Management Data
- Model
- Analysis Model
- Architectural Proof-of-Concept
- Business Analysis Model
- Business Use Case Model
- Data Model
- Deployment Model
- Design Model
- Implementation Model
- Navigation Map
- Software Requirement
- Storyboard
- Use-Case Model
- Workload Analysis Model

Model Element

- Plan
- Record
- Uncategorized

Role Sets

- Roles
- Managers
- Analysts
- Business Designer
- Business-Process Analyst
- Requirements Specifier
- Stakeholder
- System Analyst
- Developers
- General Roles
- Production & Support
- Testers

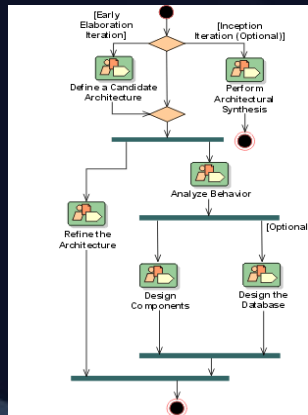
Processo

- Define a seqüência de como:
 - O trabalho está sendo desenvolvido pelos papéis;
 - Os artefatos estão sendo produzidos e evoluídos;
- Tipicamente expressos como workflows e breakdown structures.

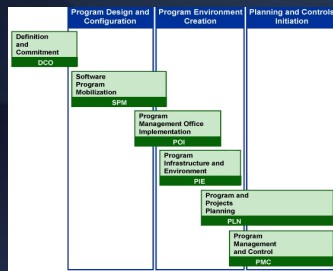
Processso



Workflow



Workflow



Breakdown / Iterations /
Increments / Sprints / Waterfall

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastructure\Samples\rup_20050708-playing

File Edit Search Configuration Window Help

config_for_PH_basic

Library

- basic_unified_process
 - Method Content
 - Processes
 - PH_basic_business_integration
 - RUP extension

Configuration

config_for_PH_basic

- Capture a Common Vocabulary
- Detail a Use Case
- Detail the Software Requirements
- Develop Requirements
- Develop Vision
- Elicit Stakeholder Requirements
- Find Actors and Use Cases
- Manage Dependencies
- Prioritize Use Cases
- Review Requirements
- Structure the Use-Case Model
- Test
- Uncategorized Tasks
- Domains
 - Analysis and Design
 - Business Modeling
 - Configuration & Change Management
 - Deployment
 - Environment
 - Implementation
 - Project Management
 - Requirements
 - Glossary
 - Requirements Attributes
 - Requirements Management
 - Software Requirement
 - Software Requirements
 - Stakeholder Requests
 - Storyboard
 - Supplementary Specifications
 - Use-Case Model
 - Vision
 - Test
 - Uncategorized
 - Work Product Kinds
 - Role Sets
 - Managers
 - Analysts
 - Business Designer

Small Use Case-based Process

Presentation Name	Index	Model Info	Type	Pred
Small Use Case-based Process	0		Capability P...	
Inception	1		Phase	
Understand Stakeholder Needs	2		Activity	
Find Actors and Use Cases	3		Task Descri...	
Define the System	4		Activity	2
Detail a Use Case	5		Task Descri...	
System Analyst		Primary Performer	Role Descri...	
Requirements Specifier		Secondary Performer	Role Descri...	
Use Case		Mandatory Input	Artifact Des...	
Glossary		Optional Input	Artifact Des...	
Use-Case Model		Optional Input	Artifact Des...	
Vision		Optional Input	Artifact Des...	
Storyboard		Optional Input	Artifact Des...	
Stakeholder Requests		Optional Input	Artifact Des...	
Use Case		Output	Artifact Des...	
Prioritize Use Cases	6		Task Descri...	
Other Work...	7		Activity	4
Elaboration	8		Phase	
Refine the System Definition	9		Activity	
Detail a Use Case	10		Task Descri...	
Requirements Specifier		Primary Performer	Role Descri...	
Use Case		Mandatory Input	Artifact Des...	
Iteration Plan		Mandatory Input	Artifact Des...	
Glossary		Optional Input	Artifact Des...	
Stakeholder Requests		Optional Input	Artifact Des...	
Use-Case Model		Optional Input	Artifact Des...	
Supplementary Specifications		Optional Input	Artifact Des...	
Requirements Management Plan		Optional Input	Artifact Des...	
Vision		Optional Input	Artifact Des...	
Storyboard		Optional Input	Artifact Des...	
Use Case		Output	Artifact Des...	
Supplementary Specifications		Output	Artifact Des...	
Analyze Behaviour	11		Activity	
Use-Case Analysis	12		Task Descri...	

Task Descriptor : detail_a_use_case

Steps

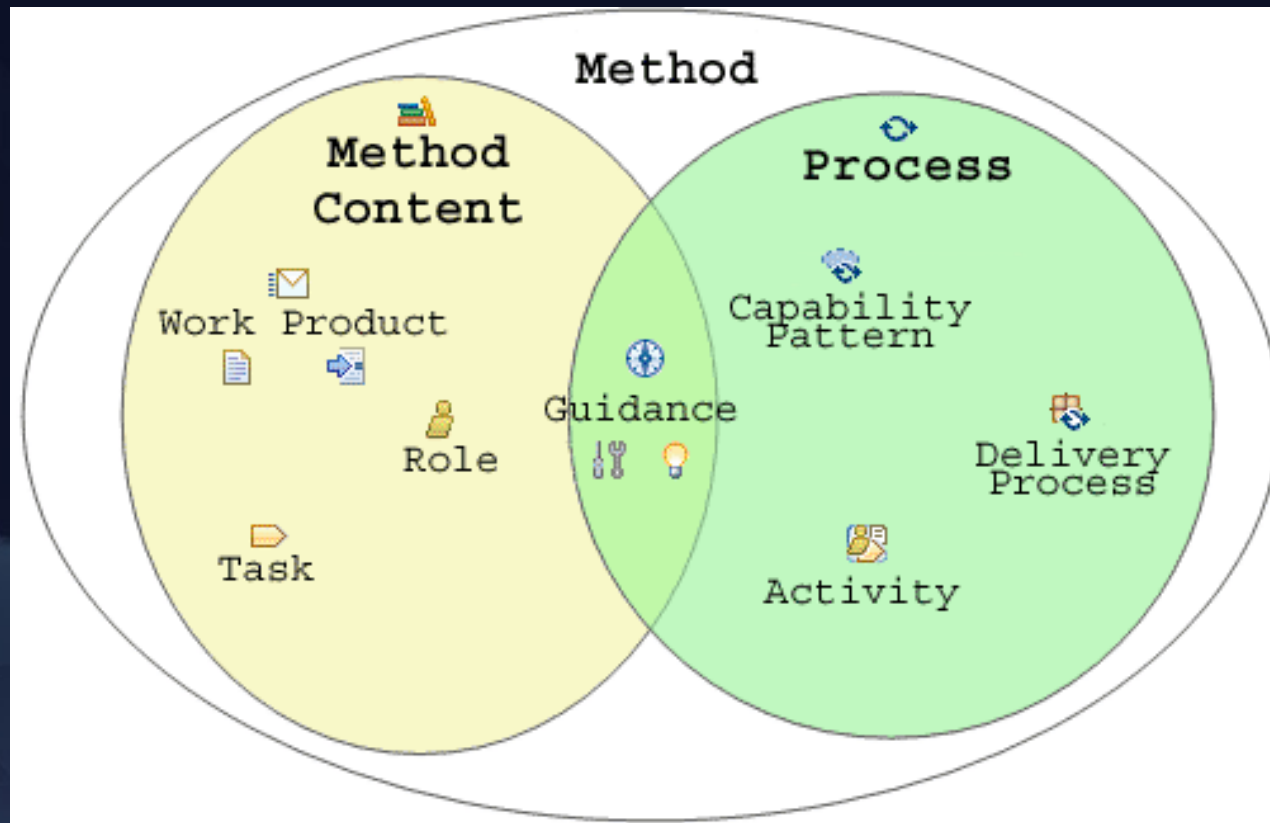
Specify the steps to perform this task descriptor.

Steps:

- Review and Refine the Scenarios
- Detail the Flow of Events
- Illustrate Relationships with Actors and Other Use Cases
- Evaluate Your Results

Add...

EPF Composer

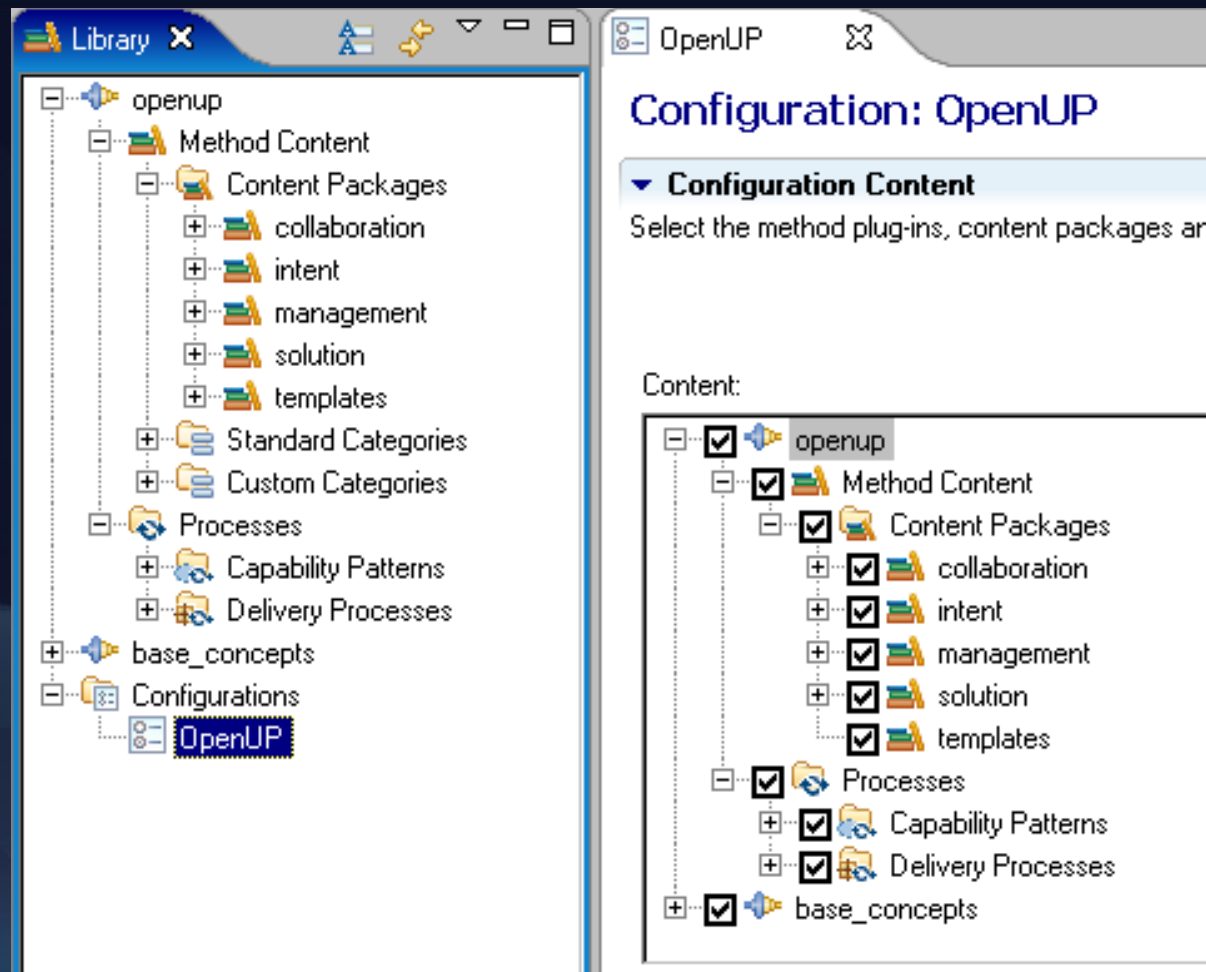


Fonte: An Overview to EPF Composer – Part 1

Cenários

- Configurando uma metodologia:
 - É o cenário mais simples;
 - Selecionar a metodologia;
 - Marcar o conteúdo da biblioteca que casa com o seu interesse;

Cenários



Fonte: An Overview to EPF Composer – Part 1

Cenários

- “Costurando” um processo:
 - Selecionar, adicionar, remover e editar partes de processos, sem influenciar o processo original;

Cenários

- Criar um processo novo:
 - Criar um processo totalmente novo;
 - Reusar blocos de processos menores (capability patterns);
 - Definir o seu próprio modelo de ciclo de vida.

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastructure\Samples\Beacon

File Edit Search Configuration Window Help

Library

- basic_unified_process
 - Method Content
 - Processes
 - Capability Patterns
 - Basic Building Blocks
 - assess_and_close_out_project
 - assess_and_plan_iteration
 - define_architecture
 - determine_architectural_feasibilit
 - develop_solution
 - initiate_project
 - manage_changes
 - manage_iteration
 - manage_requirements
 - test
 - Phase Iteration Templates
 - Delivery Processes
 - basic_unified_process
- scrum
 - Method Content
 - Processes
 - Capability Patterns
 - Scrum Building Blocks
 - assess_and_plan_sprint
 - manage_product_backlog
 - manage_sprint
 - Delivery Processes
 - scrum_and_bup
 - base_concepts
 - Configurations

Presentation Name	Index	Model Info	Type	Predecessors	Repeat...	Ong
Basic Unified Process	0		Delivery Pro...		false	false
Inception Phase - Iteration n	1	extends 'inception_p...	Activity		true	false
Elaboration Phase - Iteration n	19	extends 'elaboration...	Activity		true	false
Manage Iteration	20	extends 'manage_ite...	Activity		false	true
Initiate Iteration	21		Task Descriptor		false	false
Conduct daily meetings	22		Task Descriptor		false	false
Manage Requirements	23	extends 'manage_re...	Activity		false	false
Define the Architecture	26	extends 'define_arch...	Activity		false	false
Develop Solution (for requirement)	29	extends 'develop_sol...	Activity		false	false
Validate Build	35	extends 'test, basic_...	Activity		false	false
Create Test Cases	36		Task Descriptor		false	false
Implement Tests	37		Task Descriptor		false	false
Run Tests	38		Task Descriptor		false	false
Evaluate Test Results	39		Task Descriptor		false	false
Assess and Plan Next Iteration	40	extends 'assess_and...	Activity	20, 23, 26, 29, 35, 45	false	false
Manage Changes	45	extends 'manage_ch...	Activity		false	true
Construction Phase - Iteration n	48	extends 'constructio...	Activity		true	false
Transition Phase - Iteration n	77	extends 'transition_p...	Activity		true	false

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

Presentation Name	Index	Model Info	Type	Predecessors	Repeat...	Ong
Scrum and BUP	0		Delivery Process		false	false
Sprint 0	1		Iteration		false	false
Initiate Project	2	extends 'initiate_proj...	Activity		false	false
Assess and Plan Sprint	6	extends 'assess_and...	Activity		false	false
Sprint n	11		Iteration		false	false
Manage Sprint	12	extends 'manage_sp...	Activity		false	true
Conduct Daily Scrums	13		Task Descriptor		false	false
Manage Product Backlog	14	extends 'manage_pr...	Activity		false	true
Develop Solution (for requirement)	18	extends 'develop_sol...	Activity		false	false
Validate Build	24	extends 'test, basic_...	Activity		true	false
Create Test Cases	25		Task Descriptor		false	false
Implement Tests	26		Task Descriptor		false	false
Run Tests	27		Task Descriptor		false	false
Evaluate Test Results	28		Task Descriptor		false	false
Assess and Plan Sprint	29	extends 'assess_and...	Activity		false	false

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

Cenários

- Desenvolver a metodologia e criar ou estender processos:
 - Escolher entre criar ou estender uma metodologia existente;
 - Adicionar, modificar ou substituir artefatos, papéis, passos em tarefas, etc.

Cenários

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastructure\Samples\Beacon

File Edit Search Internal Configuration Window Help

Classic RUP (for large projects)

Library

- rup
 - Method Content
 - Content Packages
 - Architecture
 - Assessment
 - Design
 - Database Design
 - Design with Use-Case
 - GUI Design
 - Operation Design
 - Real-Time Design
 - Roles
 - Tasks
 - Work Products
 - rup_analysis_mod
 - rup_analysis_
 - rup_design_mode
 - Guidance

Configuration

Classic RUP (for large projects)

- Disciplines
- Domains
 - RUP Domains
 - Analysis and Design
 - Data Migration Specifica
 - Analysis Model
 - Analysis Class
 - Architectural Proof-of-C
 - Data Model
 - Design Model
 - Deployment Model
 - Navigation Map
 - Reference Architecture
 - Software Architecture C
 - User-Interface Prototyp
 - Service Model
 - Business Modeling
 - Configuration & Change Ma

Work Product (Artifact): rup_analysis_class

General Information

Provide general information about this artifact.

Name: rup_analysis_class

Presentation name: Analysis Class

Unique ID:

Brief description: This work product specifies elements of an early conceptual model for 'things in the system which have responsibilities and behavior'.

Detail Information

Provide detailed information about this artifact.

Purpose: Analysis classes are used to capture the major system.

Main description: Analysis Classes specify elements of an early o the system which have responsibilities and beha prototypical classes of the system, and are a fir abstractions that the system must handle. Anal maintained in their own right, if a "high-level", co is desired. Analysis classes also give rise to the

Key considerations:

Notation

Provide notation information about this artifact.

Brief outline:

Representation options:

Tailoring

Provide tailoring information about this artifact.

Impact of not having:

Work Product (Artifact): rup_analysis_class

Representation options:

CRC Card technique - see [VIR90] for details of this technique). On the front side of the card, capture the name and description of the class. An example for a Course in a course registration system is listed below:

Class Name	Description
Course	The Course is responsible for maintaining information about a set of course sections having a common subject, requirements and syllabus.

Responsibilities		
To maintain information about the course.		
Attributes		
Name	Description	Type
Course Title	The name of the course	string
Description	A short description of the course	string

On the back of the card, draw a diagram of the class:

```
graph LR
    Course[Course] -- "0..*" --> Section[Section]
    Section -- "1..*" --> Professor[Professor]
    Section -- "1..*" --> Student[Student]
    Section -- "0..*" --> Textbook[Textbook]
    Section -- "1..*" --> Room[Room]
    Course -- "pre-requisite" --> Textbook
    Course -- "readingList" --> Textbook
```

Conclusão

- OpenUP:
 - Processo mínimo, completo e extensível;
 - Princípios e técnicas ágeis;
 - Ciclo de vida estruturado;
- EPF Composer:
 - Permite criação, manutenção e distribuição de processos;
 - Fácil de usar.

Referências

- **Agile Manifesto** - <http://www.agilemanifesto.org>
- **Eclipse Process Framework project** - <http://www.eclipse.org/epf>
- **Lyons, B.: The Open Unified Process: A Brilliant, Collaborative March into Open**
- **EPF Wiki** - <http://www.epfwiki.net/>
- **OpenUP** - <http://www.epfwiki.net/wikis/openuppt/>

Referências

- **NumberSim** -
<http://www.numbersix.com/news/n6articles/openUp.html>
- **Wikipedia - OpenUP** -
<http://en.wikipedia.org/wiki/OpenUP>
- **EclipseCon 2007** -
<http://www.eclipsecon.org/2007/index.php?page=susb/&id=3621>

Dúvidas?

André Aziz (aaca@cin.ufpe.br)