

ENGENHARIA DE SOFTWARE II

CAPÍTULO 1 – QUALIDADE, PROCESSO E O PRODUTO SOFTWARE, r3

Neste capítulo serão abordados os seguintes itens:

- 1.1 O produto *software*
- 1.2 Qualidade, processos, métodos e ferramentas
- 1.3 O processo de software
 - 1.3.1 Recursos do projeto e do produto
 - 1.3.2 Decomposição do Processo
 - 1.3.3 Características de um bom processo
- 1.4 Fusão do produto e do processo
 - 1.4.1 Gerenciamento da equipe de desenvolvimento com PMBOK

Referências: [PRE02] – Capítulo 1: O produto; [PRE07] – Capítulo 1: Software e Engenharia de Software; Desenvolvimento Ágil; [SOM05] – Capítulo 1: Introdução; Capítulo 2: Engenharia de sistemas com base em computadores; [STA06] – Parte 2: Conceitos de Tecnologia de Informações

1.1 O PRODUTO SOFTWARE

O *software* é o produto mais importante desta última era. E devido a sua dualidade com o hardware que com o passar do tempo melhora o desempenho, diminui o tamanho e reduz o custo, permite-se utilizar o software para gerar sistemas mais sofisticados.

O **software possui um duplo papel na produção**, não só pode constituir um produto completo, como também pode ser o veículo de melhora de outro produto.

Exemplo:

- **Como produto de software** - podemos citar toda a linha de software e que normalmente são comercializados, tais como software de sistema e de aplicação. Um exemplo simples seria o produto Word da Microsoft.
- **Como veículo de melhora dos produtos:**
 - **software** – podemos citar o software “Globalink”, um tradutor de línguas que ao instalar pode ser inserida como uma função de chamada no Word, bem como pode ser usado isoladamente como um produto completo.; e
 - **industrial** – quando o software esta associado a um *hardware* específico de uma máquina que pode assim ser automatizada.

1.2 QUALIDADE, PROCESSOS, MÉTODOS E FERRAMENTAS [Adaptado de Pressman, 2002]



Camadas da engenharia de software. Fonte: Pressman, 2002.

A engenharia de software é uma tecnologia em camadas (Figura abaixo) e que deve se apoiar num compromisso organizacional com o **foco na qualidade**. A **qualidade** leva à cultura de um processo de melhoria contínua e abordagens cada vez mais amadurecidas para a engenharia de software.

O fundamento da engenharia de software é a camada de **processos**. O processo é responsável por manter integradas as camadas da tecnologia, permitindo o desenvolvimento racional e oportuno do software.

Exemplos de processos:

- **Áreas-Chave de Processo - ACP (Key Process Areas – KPA) do CMMI - Capability Maturity Model Integration** – O CMMI define uma estrutura para um conjunto de 22 áreas-chave de processo distribuídas em 5 níveis de capacitação, que formam a base para o controle gerencial de projetos de software. Estabelecem o contexto no qual os métodos técnicos são aplicados, os produtos de trabalho (modelos, documentos, dados, relatórios, formulários) são produzidos, marcas de referência (**milestones**) são estabelecidas, qualidade é assegurada e modificações são adequadamente geridas. [PAU93]; [CMM00]; [PRE02]
- **Processo Unificado da Rational (RUP – Rational Unified Process)** – É um processo proprietário criado pela Rational Software Corporation, adquirida pela IBM. Considerado também uma metodologia é vendido na forma de um produto, podendo ser adaptado às necessidades da empresa. O RUP é desenvolvido, entregue e mantido como um software, recebendo atualizações regulares.

Os **métodos** fornecem a técnica de como fazer para construir software. Incluem um amplo conjunto de tarefas que abrangem análises de requisitos, projeto, construção de programas, teste e manutenção. Os métodos formam um princípio básico que regem cada área da tecnologia e incluem atividades de modelagem e outras técnicas descritivas.

As **ferramentas** da engenharia de software fornecem apoio automatizado ou semi-automatizado para o processo e para os métodos. Quando as ferramentas são integradas, e que podem ser reusadas, um sistema de apoio ao desenvolvimento de software pode ser estabelecido.

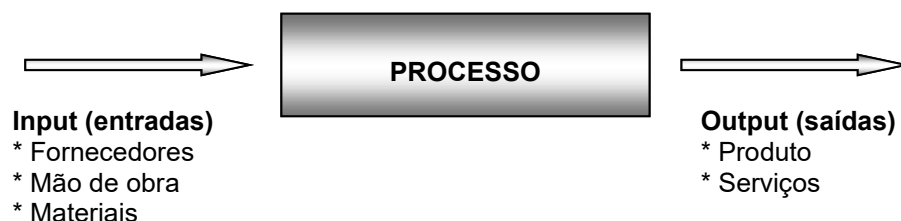
A ferramenta **CASE – Computer-aided Software Engineering** é um software de apoio que cria um ambiente de desenvolvimento de software. Esta ferramenta combina software hardware em uma base de dados de engenharia de software que contém informações sobre análise, projeto, construção de programas e teste.

1.3 O PROCESSO DE SOFTWARE [Adaptado de Pressman, 2002]

Um **processo** é “uma sequência de fatos, atividades ou operações que apresentam certa unidade e/ou que se reproduzem com certa regularidade”.

Em **cada processo deve se definir os pontos de início (input) e fim (output)** do processo, chamados de **marcas de referência (ou milestones)**. Veja a figura abaixo

- **input:** o ponto inicial, normalmente marca os insumos necessários para o processo, tais como: fornecedores, recurso humano, materiais e energia.
- **output:** o ponto final, é a entrega do produto ou serviço ao cliente.



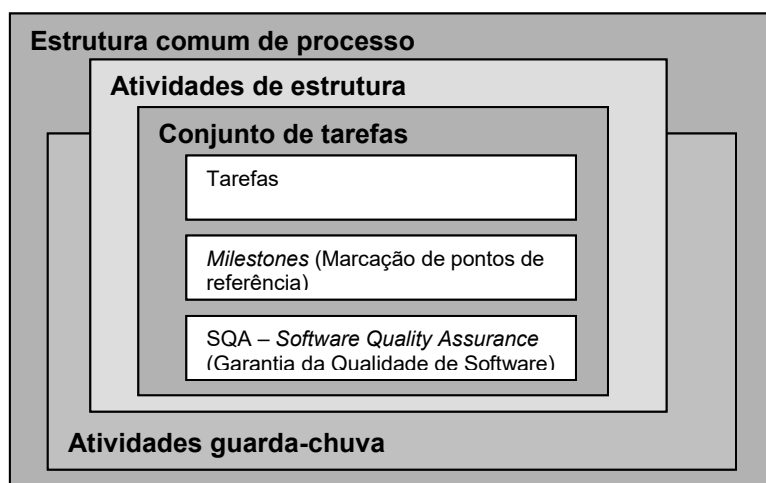
Modelo básico do processo.

1.3.1 Recursos do Projeto e do Produto

Os processos do projeto recaem em uma das duas categorias:

- **Processo de gerenciamento de projeto** descreve, organiza e completa o trabalho do projeto. No gerenciamento do projeto os principais recursos a serem tratados são:
 - Recursos Humanos;
 - Recursos Materiais;
 - Recursos Tecnológicos;
 - Recursos Financeiros.
- **Processo de gerenciamento do produto** especifica e cria projeto do produto. O produto como vimos anteriormente, pode ser um componente de outro produto, serviços a serem executados, resultados da produção ou do comércio, ou produto produzido em fábrica. Processos de gerenciamento de produto são tipicamente definidos em uma determinada etapa do ciclo de vida do projeto e variáveis pela área de aplicação. Se considerarmos o gerenciamento do produto (no caso gerenciamento da infraestrutura da TI) no desenvolvimento de sistemas de informação, os principais recursos a serem tratados na seguinte sequência:
 - Recursos de *Software*;
 - Recursos de Dados;
 - Recursos de Rede de Computadores;
 - Recursos Humanos (*Peopleware*);
 - Recursos de *Hardware*;

O processo de software ou processo de Engenharia de Software, normalmente pode representar uma determinada funcionalidade do software. É uma sequência coerente de práticas, que objetiva o desenvolvimento ou evolução de sistemas de software. Estas práticas englobam as atividades de especificação, projeto, implementação, testes e caracterizam-se pela interação de ferramentas, pessoas e métodos. A estrutura comum de um processo de software pode ser caracterizada como mostrada na Figura abaixo.

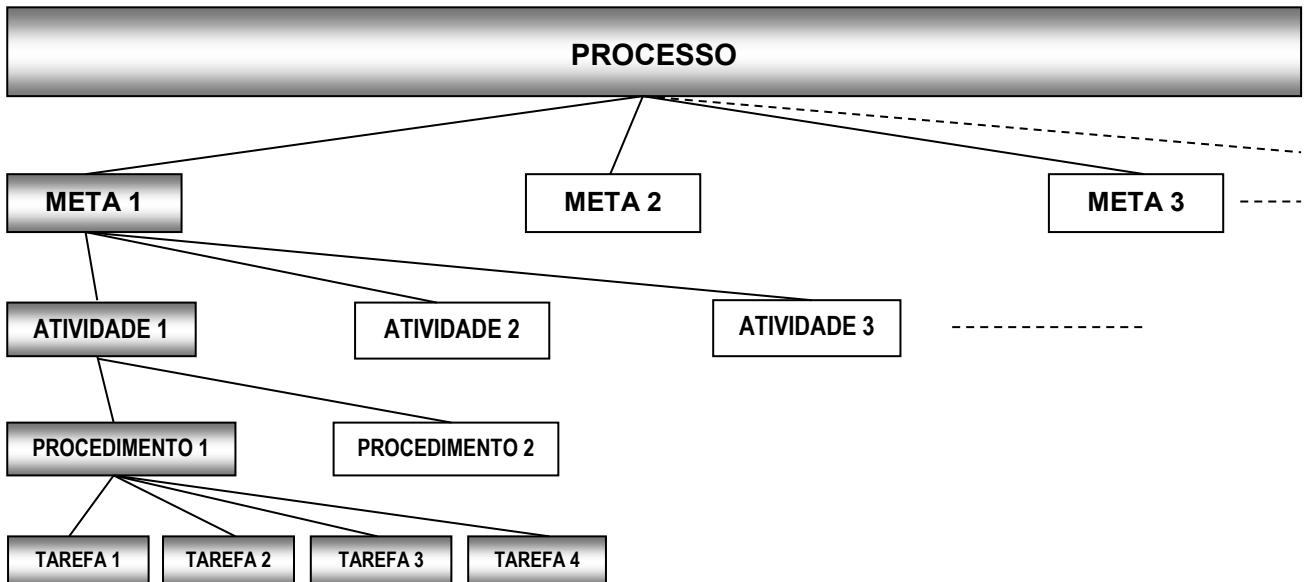


O processo software. Fonte: Pressman, 2002.

1.3.2 Decomposição do Processo

Aplicando o conceito de modularidade e de acordo com o escopo do projeto, o processo pode ser estruturado em vários módulos, com base no seu tamanho e complexidade. A Figura abaixo mostra como seria a estrutura genérica de um processo nos moldes do CMMI.

Estrutura do processo



Estrutura modular genérica do processo.

A **meta** é um marco de referência que traça o objetivo de uma parte do processo. Normalmente associada a um prazo de cumprimento da meta (*baseline*). Tem como objetivo manter os planos, artefatos e atividades de software consistentes com os requisitos alocados.

A **atividade** é a realização de uma função específica do processo, normalmente compreendida na meta.

O **procedimento** é a forma de como se deve agir, fazer ou cumprir uma atividade. Os procedimentos normalmente são caracterizados por meio de uma lista que apresenta passo a passo e de forma sequencial as tarefas que deverão ser executadas.

As **tarefas** são os trabalhos que devem ser realizados. Representa uma empreitada de serviços, ou seja, uma quantidade de trabalhos realizados ou a realizar dentro de um prazo determinado.

1.3.3 Características de um bom processo

A criação de software profissional deriva da satisfação dos processos e da finalização do produto. Um pequeno número de quadro de atividades é aplicado para todos os projetos de software, indiferentemente do tamanho, complexidade ou qualidade exigida.

O bom processo de software fornece o **arcabouço** com o qual pode ser estabelecido um plano abrangente para o desenvolvimento do software. O arcabouço do dicionário Houaiss significa "estrutura para uma construção". São vários conjuntos de diferentes tarefas, marcas de referência, entregas e pontos de garantia de qualidade, que permitem que as atividades do arcabouço sejam adaptadas às características do projeto de software e às necessidades da equipe de projeto.

As **fases genéricas que caracterizam o processo de software (definição, desenvolvimento e manutenção)** são aplicáveis a todo software. Estas fases são discutidas na modelagem do processo, que compõe um conjunto de paradigmas de engenharia de software (modelos: cascata, incremental, prototipação, espiral, RAD e alguns outros).

As principais características de um bom processo são:

- Configurável para diferentes organizações.
- Adaptável para diferentes tamanhos e tipos de projetos.
- Bem definido, gerenciável e repetível.
- Com nomenclatura universal e métricas para planejamento e gerenciamento do projeto.
- Integrado com ferramentas que o suportem.

1.4 FUSÃO DO PRODUTO E DO PROCESSO

A fusão do produto e do processo ocorre a partir das atividades de efetivação do processo. Cada atividade inicia com a definição do **arcabouço do processo**, que é a escolha dos membros da equipe de desenvolvimento, escolha do modelo de desenvolvimento de software a ser empregado, métodos a serem aplicados, definição das ferramentas e técnicas para o desenvolvimento.

Para iniciar um método sugerido pelo PMBOK para fazer parte do arcabouço é a **MTR – Matriz de Responsabilidades (MR)**, que é dada como um método eficiente para distribuição de responsabilidades entre os membros da equipe.

1.4.1 Gerenciamento da equipe de desenvolvimento com PMBOK

Para gerenciar o processo, o PMBOK orienta o uso da **Matriz de Responsabilidades**. É um método bem abrangente e muito útil no processo organizacional. A matriz é um arranjo de vários elementos num quadro retangular que comporta colunas e linhas. Observe a matriz abaixo.

Existem basicamente três tipos de matrizes no processo organizacional.

- Distribuição de Atividades (ou Tarefas) do processo, relacionadas com os integrantes da equipe de desenvolvimento.
- Mapeamento do processo
- Acompanhamento do processo

1.4.1 Distribuição de Atividades (ou Tarefas) relacionadas com os Participantes do projeto

Nas linhas ficam as atividades (ou tarefas) do processo e nas colunas os participantes, que são as pessoas atuantes ou interessadas no processo (**stakeholder**: autor do processo, diretores, gerentes, engenheiros, técnicos, mediadores, secretários, consultores especialistas, operadores e demais envolvidos no processo). Neste tipo de matriz pode-se também (em vez de colocar os participantes nas colunas) relacionar as atividades (ou tarefas) com os setores da empresa atuantes no processo. Veja a matriz a seguir.

A distribuição de responsabilidades não deve ser atribuída ao acaso. A identificação das responsabilidades é de vital importância para o controle do recurso humano no processo organizacional.

Algumas questões devem ser respondidas para determinar os critérios de identificação das responsabilidades:

- Quem faz a maior parte do trabalho?
- Quem é mais afetado?
- Quem influencia mais o processo?
- Quem é mais cobrado pelos resultados do processo, pelas operações e pelas atividades?

Principais atributos do processo de montagem da matriz:

- **R** (= Responsável) – é o autor do processo, da atividade ou tarefa. **Para cada atividade (ou tarefa) deve existir obrigatoriamente um e apenas um “R”;**
- **A** (= quem Aprova) – o responsável pela aprovação normalmente é o cliente (interno ou externo), diretor ou gerência do processo. **O atributo “A” deve ser usado um apenas para a atividade (ou tarefa) porém, não existe obrigatoriedade para aplicar este atributo em todas as tarefas.** Nem todas as atividades (ou tarefas) necessitem de aprovação;
- **S** (= quem da Suporte) – são os auxiliares diretos que trabalham na atividade junto ao responsável (R);
- **P** (Participante) – são pessoas, terceiros, consultores que possuem conhecimentos específicos sobre uma determinada atividade.

➔ **Nota:** Apesar de permitir colocar dois atributos simultâneos em uma determinada célula para algum membro da equipe, não é aconselhável que se faça isto, deve haver um bom senso quanto ao tamanho, complexidade ou qualidade exigida para a tarefa.

Figura: Mapeamento do processo de Gerenciamento de Requisitos para o elaboração do escopo do projeto. Distribuição de papéis e responsabilidades com base na matriz de responsabilidades do PMBOK.

item	PESSOAL	Cliente	Analista do Processo	Gerente do Projeto	Gerente do Sistema	Analista de Sistemas	Codificação e Testes
	ATIVIDADE						
a	Objetivo do Negócio	R	S	S	S	-	-
b	Requisitos do processo	A	R	S	P	-	-
c	Modelagem do processo de negócio	A	S	R	P	-	-
d	Caso de Uso	A	S	S	R	P	-
e	Diagrama de Componentes	-	-	A	S	R	S
f	Diagrama de Classe	-	-	P	A	S	R

Atributos do Processo: R = Responsável; A = quem Aprova; S = quem da Suporte; P = Participante;

Fonte: PMBOK, 2010 e PMBOK, 2000.

BIBLIOGRAFIA

- ABNT – Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 12207 – Tecnologia de informação - Processos de ciclo de vida de software.** Rio de Janeiro: ABNT, 1998. Manual.
- CMMI Product Development Team. **CMMI for Systems Engineering/Software Engineering, Version 1.02. Staged Representation.** Pittsburgh, U.S.A: Carnegie Mellon, SEI, 2000.
- FIORINI, Soeli T.; STAA, Arndt von; BAPTISTA, Renan Martins. **Engenharia de Software com CMM.** Rio de Janeiro: BRASPORT, 1999.
- PAULK, Mark C.; CURTIS, Bill; CHRISSIS, Mary Beth; WEBER, Charles V. **The Capability Maturity Model for Software v1.1.** Pittsburgh – USA: SEI - Software Engineering Institute Customer Relations, 1993. Artigo.
- PMBOK, Project Management Body of Knowledge. **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK). – 4ª ed.** Atlanta, USA: Project Management Institute, Inc. 2010.
- PMBOK, Project Management Institute. **Project Management Body of Knowledge.** Pennsylvania, USA: PMBOK Guide. 2000.
- PRESSMAN, Ph.D. Roger S. **Engenharia de Software. – 5ª ed.** Rio de Janeiro: McGraw-Hill, 2002.
- PRESSMAN, Ph.D. Roger S. **Engenharia de Software. – 6ª ed.** Rio de Janeiro: McGraw-Hill, 2007.
- PFLIEGER, Shari Lawrence. **Engenharia de Software. – 2ª ed.** São Paulo: Prentice Hall, 2004.
- SOMMERVILLE, Ian. **Engenharia de software.** São Paulo: Pearson Addison Wesley, 2003.