

Analise e Modelagem de Sistemas

O processo de software

Eduardo Furlan Miranda

2025-07-01

Fonte: WERLICH, C. *Análise e Modelagem de Sistemas*.
Londrina: EDE SA, 2020. ISBN 978-85-522-1683-4.

O Desafio do Analista de Sistemas

- Cotidiano agitado: muitas tarefas e necessidade de planejamento
 - Visitas ao cliente
 - Orientação a programadores
 - Pesquisas e planejamento
- Organização do tempo: essencial para evitar prejuízos
 - Tarefas esquecidas geram problemas no desenvolvimento
- Problema identificado: clientes reportam incômodo com tempo de entrega
 - Missão de investigar e propor melhorias

O Que É um Processo de Software?

- Definição (Sommerville 2011): conjunto de atividades e resultados relacionados
 - Leva à produção de um software desenvolvido
- Definição (Pressman 2016): abordagem adaptável, não rígida
 - Equipe escolhe processos que se encaixam na filosofia da empresa
- Engenharia de software: abordagem sistemática para produção de software
 - É o que chamamos de Processo de Software

Objetivos do Processo de Software

- Foco principal: qualidade do produto final
 - Redução de custos de produção
- Prazos de entrega: cumprir contratos com o cliente
 - Garantir a satisfação do cliente
- Mecanismo essencial: obter software de qualidade
 - Gerenciar o desenvolvimento eficientemente

Benefícios do Processo de Software (I)

- Padronização: cria uma forma padronizada de gerar serviços e produtos
- Reutilização: permite repetir serviços e produtos
 - Reaproveita partes já produzidas e padronizadas
- Retenção de conhecimento: conhecimento é preservado na empresa
 - Novos integrantes podem dar continuidade

Benefícios do Processo de Software (II)

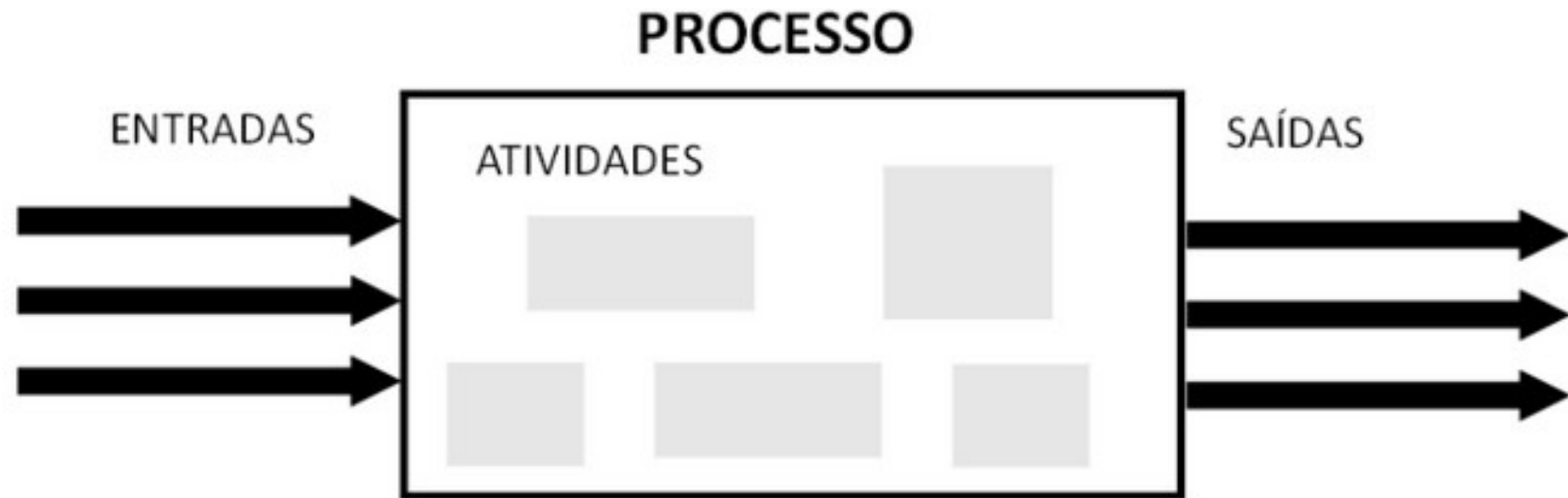
- Definição e guia: serve para definir e guiar atividades de um Projeto de Software
- Especificação completa: permite detalhar todo o processo de desenvolvimento
- Tarefas determinadas: define o que deverá ser executado pela equipe e individualmente
- Redução de riscos: torna os resultados mais previsíveis

Benefícios do Processo de Software (III)

7

- Visões comuns: proporciona alinhamento para a equipe de desenvolvimento
 - Facilita a comunicação interna
- Agilidade: pode ser usado como um "template" para outros projetos
 - Aumenta a agilidade em novos desenvolvimentos
- Resolução de problemas: ajuda a superar problemas como a troca de pessoal

Representação de um Processo de Software



Parâmetros na Definição de Processos (Engholm Jr. 2010)

- Evento de início: o que determina o começo do processo
- Matriz de responsabilidades: define quem fará o quê
- Atividades e sequencialidade: as atividades a serem executadas e sua ordem
- Entradas e saídas: de cada atividade do processo

Mais Parâmetros na Definição de Processos

10

- Regras e políticas: critérios a serem aplicados nas atividades
- Infraestrutura necessária: recursos e ambiente para o desenvolvimento
- Resultado gerado: o que é produzido na execução de cada processo
- Prazos de entrega: definidos em cada fase do processo

Estrutura de um Processo Genérico de Software

11

- Conjunto mínimo: atividades essenciais para obter um software finalizado
- Quatro atividades fundamentais (Sommerville 2011):
 - Especificação de software
 - Projeto e Implementação de software
 - Validação de software
 - Evolução de software

Atividade Fundamental 1: Especificação de Software 12

- Definição do software: o que será desenvolvido, suas restrições e funcionalidades
- Atividades comuns:
 - Estudo de Viabilidade
 - Elicitação e Especificação de Requisitos
 - Validação dos Requisitos
- Comunicação essencial: entendimento dos requisitos com os envolvidos

Atividade Fundamental 2: Projeto e Implementação de Software

13

- Projeto: definição das estruturas do software
 - Estruturas modulares, interfaces gráficas
 - Estruturas de dados (banco de dados)
- Modelagem: criação de diagramas para melhor entendimento
 - Base para codificação e validação
- Implementação: codificação do software e criação do banco de dados

Atividade Fundamental 3: Validação de Software

14

- Verificação: se o que foi construído atende às solicitações do cliente
 - Assegura a qualidade do software
- Testes: realizados para validar códigos e verificar requisitos
- Feedback do cliente: cliente testa e fornece retorno

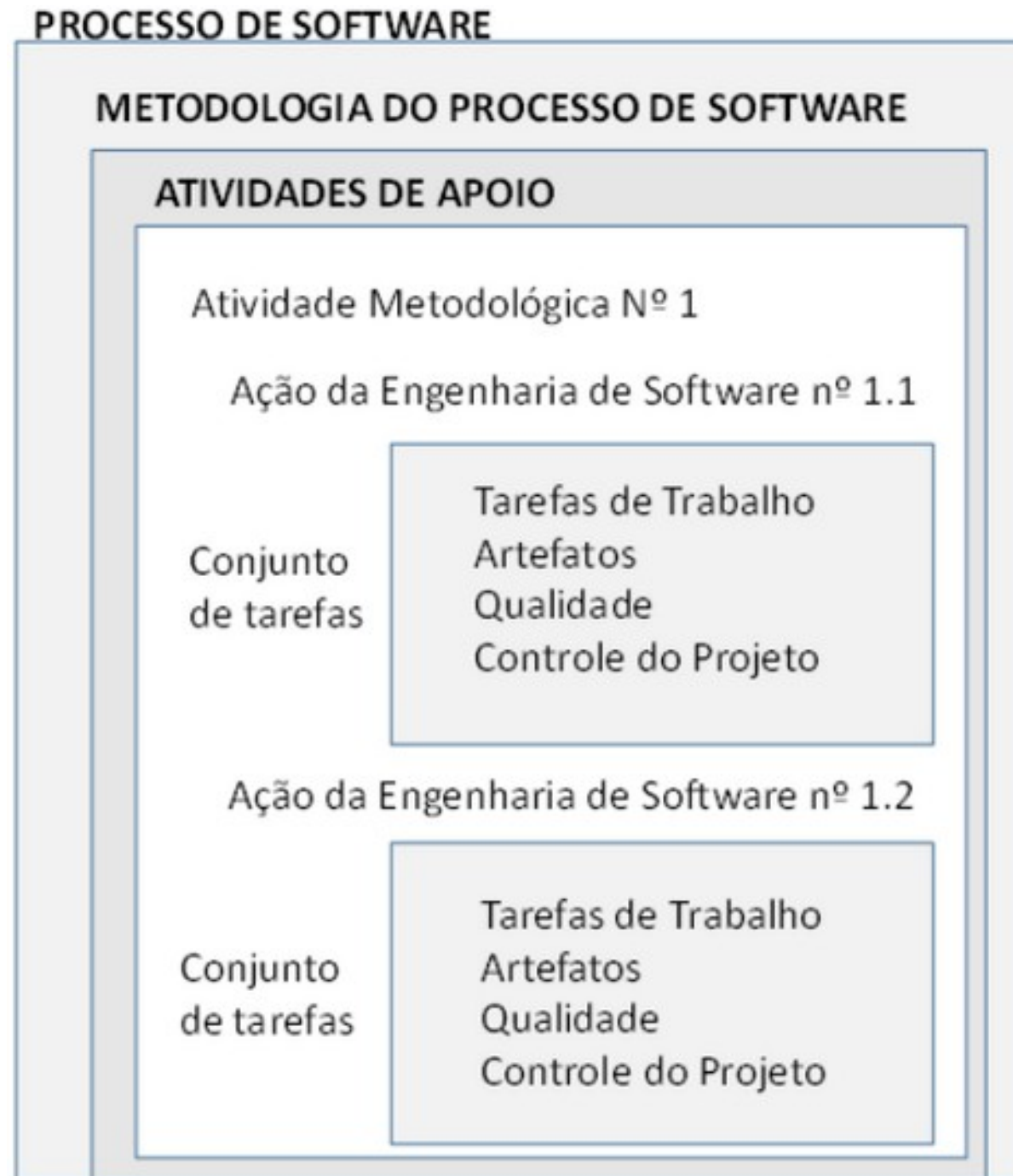
Atividade Fundamental 4: Evolução de Software

15

- Processo contínuo: melhoramento do software após a entrega
 - Inclusão de novos recursos
- Adaptações: realizadas por um período acordado
 - Acompanha alterações solicitadas pelo cliente
- Diferenciação: cada software é único, exigindo adaptação contínua

Exemplo de Metodologia de Processo de Software

16



As Cinco Atividades Metodológicas (Pressman 2016)

17

- Engenharia de software: uma metodologia genérica é composta por:
 - Comunicação
 - Planejamento
 - Modelagem
 - Construção
 - Entrega

Atividade Metodológica 1: Comunicação

- Primeira ação: entender os objetivos do projeto
- Comunicação primordial: entre todos os envolvidos
- Resultados: levantamento claro dos requisitos e funcionalidades

Atividade Metodológica 2: Planejamento

- Criação de um plano: um "mapa" do projeto do software
 - Documentação do Levantamento de Requisitos
 - Documentação da Especificação de Requisitos
- Detalhamento: descreve tarefas técnicas, riscos e recursos
 - Estimativas de prazos e alocação de recursos
 - Cronograma de trabalho

Atividade Metodológica 3: Modelagem

- Objetivo: melhor entendimento das necessidades do software
- Criação: modelos e diagramas são desenvolvidos
 - Utilizados para a codificação do software
 - Para validação das partes envolvidas no projeto

Atividade Metodológica 4: Construção

- Fase de codificação: realização da programação baseada nos modelos
 - Implementação das decisões de Especificação e Projeto
- Testes inclusos: realização de testes para validar os códigos gerados
 - Verificação da conformidade com os requisitos

Atividade Metodológica 5: Entrega

- Liberação do software: parcial ou total para o cliente
- Feedback do cliente: cliente realiza testes e fornece retorno
- Fase pós-entrega: adaptações e correções por um período acordado

Fluxos de Processo (Pressman 2016)

- Definição: descreve como as atividades metodológicas são organizadas
- Variabilidade: a sequencialidade das atividades pode mudar
- Tipos de fluxos:
 - Linear
 - Interativo
 - Evolucionário
 - Paralelo

Tipo de Fluxo 1: Fluxo de Processo Linear

- Característica: atividades realizadas em sequência
- Ordem: Comunicação -> Planejamento -> Modelagem -> Construção -> Entrega
- Ponto de início e fim: inicia na Comunicação e termina na Entrega

Tipo de Fluxo 2: Fluxo de Processo Interativo

25

- Característica: repetição de uma ou mais atividades
- Propósito: repetição ocorre antes de avançar para a próxima atividade

Tipo de Fluxo 3: Fluxo de Processo Evolucionário

26

- Característica: atividades executadas de modo circular
- Natureza incremental: cada ciclo envolve as cinco atividades
 - Gera uma versão mais completa do software

Tipo de Fluxo 4: Fluxo de Processo Paralelo 27

- Característica: atividades realizadas simultaneamente
 - Duas ou mais atividades podem ser executadas ao mesmo tempo
- Exemplo: atividade de Comunicação pode ocorrer em paralelo com a Análise
- Benefício: agiliza o desenvolvimento, como a Especificação e o Projeto

Qualidade no Processo de Software

- Não garante sozinho: o processo não garante qualidade ou prazo por si só
 - Nem o atendimento total às funcionalidades solicitadas
- Influência direta: padrões de qualidade impostos durante os processos
- Necessidade: estabelecer procedimentos e padrões para garantir qualidade
- Vantagem da modelagem: permite examinar e aprimorar antes da finalização

Avaliação e Aperfeiçoamento dos Processos

29

- Importância: processo deve atender a critérios básicos de qualidade
 - Garante integração e validação entre atividades
- Abordagens de avaliação e aperfeiçoamento (Pressman 2016):
 - SCAMPI
 - CBA IPI
 - SPICE
 - ISO 9001:2000 para software

Abordagens de Melhoria: SCAMPI e CMMI

- SCAMPI (Standard CMMI Appraisal Method for Process Improvement):
 - Modelo de avaliação do Processo em cinco etapas
 - Ajuda a coletar evidências e identificar falhas
- CMMI (Capability Maturity Model Integration):
 - Conjunto de práticas para aumentar o amadurecimento organizacional
- Modelos CMMI:
 - CMMI for Development (melhores práticas para desenvolver produtos)
 - CMMI for Acquisition (melhores práticas para adquirir produtos)
 - CMMI for Services (melhores práticas para entregar serviços)

Abordagens de Melhoria: CBA IPI, SPICE e ISO 9001:2000

31

- CBA IPI (CMM Based Appraisal for Internal Process Improvement):
 - Técnica de diagnóstico para avaliar maturidade da organização
 - Identifica pontos fortes/fracos e prioriza melhorias
- SPICE (ISO/IEC15504):
 - Padrão com requisitos para avaliação da eficácia do Processo de Software
 - Fornece estrutura para avaliação
- ISO 9001:2000 para Software:
 - Padrão genérico de qualidade aplicável a produtos e serviços
 - Outros modelos: ISO/IEC 9126, ISO9000, ISO/IEC12207

Conclusões e Próximos Passos (Propostas de Melhoria)

32

- Controle de erros: adotar abordagens metodológicas
 - Testar e utilizar novas metodologias apropriadas
- Implantar métodos de melhoria:
 - Considerar SCAMPI, CBA IPI, SPICE, ISO 9001:2000
- Pesquisa contínua: fator importante na carreira do Analista de Sistemas
 - Pesquisar normas ISO 9001:2000 e metodologias ágeis