



# **Processo de Software**

## **Engenharia de Software II**

**Profa. Andréa Sabedra Bordin**

# ENGENHARIA DE SOFTWARE

---

A aplicação de uma abordagem **sistemática**, **disciplinada** e **quantificável** para o desenvolvimento, operação e manutenção de software.

ISO/IEC/IEEE Sistemas e Vocabulário de Engenharia de Software (SEVOCAB)

# O QUE É UM PROCESSO?

---

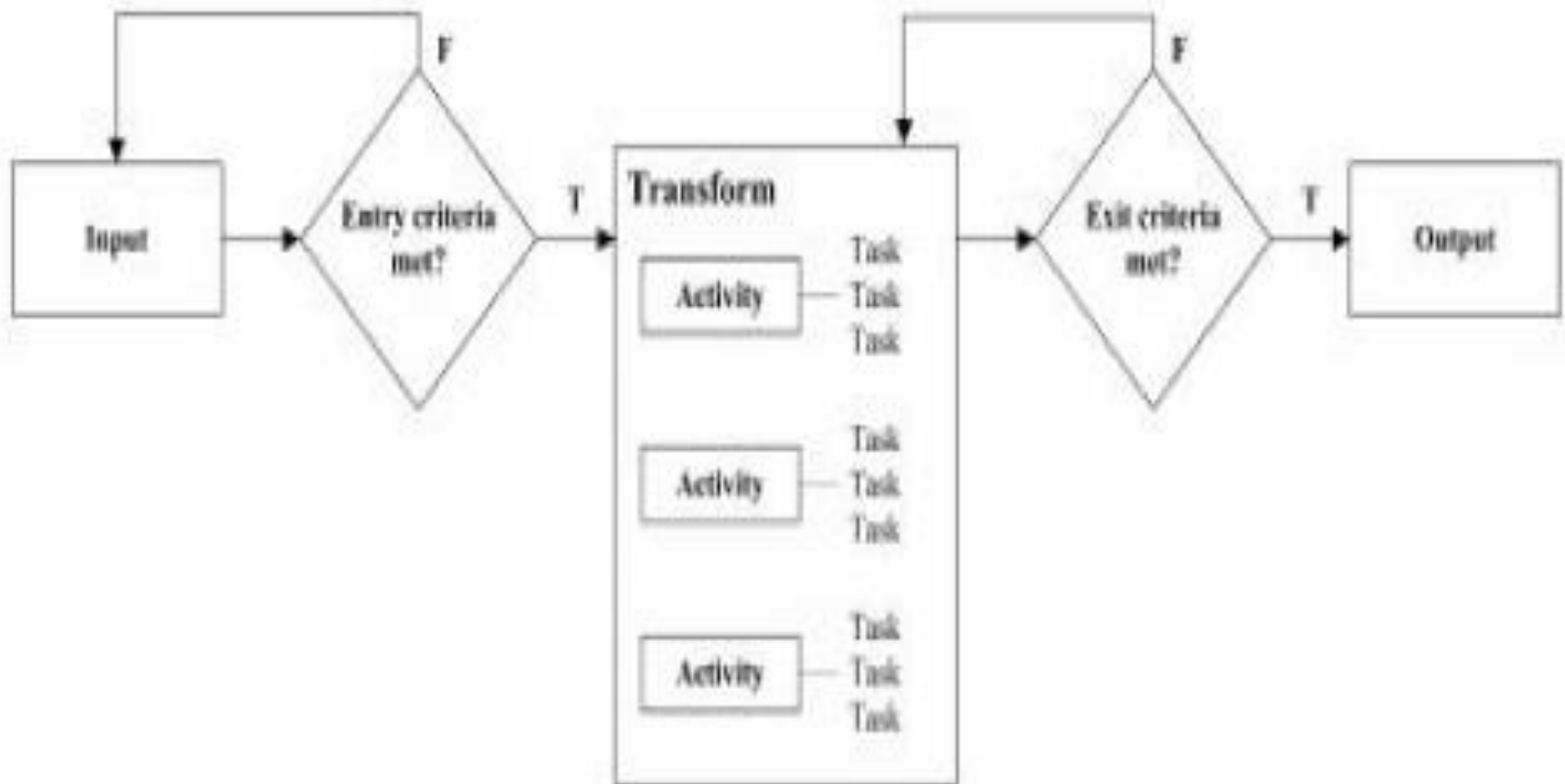
**Processo é um conjunto de atividades**

Interdependentes

Com responsáveis

Com entradas e saídas definidas

# ELEMENTOS DE UM PROCESSO



Para que o desenvolvimento de sistemas deixe de ser **artesanal** e aconteça de forma mais **previsível, organizada** e com mais **qualidade**, é necessário que se estabeleça e se compreenda um **processo de produção**.

# O QUE É UM PROCESSO DE SOFTWARE?

---

Um processo de engenharia de software é formado por um **conjunto de passos de processo** parcialmente ordenados, relacionados com **artefatos, pessoas, recursos, estruturas organizacionais e restrições**, tendo como objetivo produzir e manter os **produtos de software** finais requeridos.



Especificação  
Projeto  
Implementação  
Validação/Testes  
Manutenção

# VANTAGENS DE SE ADOPTAR UM PROCESSO

---

## *O tempo de treinamento pode ser reduzido*

- Com processos bem definidos e documentados é mais fácil encaixar novos indivíduos na equipe do que quando não se tem processos definidos

## *Produtos podem ser mais uniformizados*

- A existência do processo não garante uma uniformidade na qualidade dos produtos, mas certamente uma equipe com processo bem definido tende a ser mais previsível do que a mesma equipe sem processo algum

## *Possibilidade de capitalizar experiências*

- Um bom processo, bem gerenciado, deve ter mecanismos para melhoria embutidos. Assim, se um desenvolvedor descobrir um meio de fazer as coisas melhor do que descrito no processo, deve haver meios para incorporar estas alterações no processo.

# VANTAGENS DE SE ADOTAR UM PROCESSO

---

- Facilitar a compreensão humana, comunicação e coordenação;
- Auxiliar na gestão de projetos de software;
- Medir e melhorar a qualidade dos produtos de software em um maneira eficiente;
- Apoiar a melhoria do processo.

*Software Body of Knowledge (SWEBOK)*



# ESTRUTURA DE UM PROCESSO

---

## Fases

- Cada fase de um processo deve ter um macro-objetivo bem estabelecido



## Disciplinas

- Um conjunto de atividades correlacionadas, as quais servem a um objetivo específico dentro do processo de desenvolvimento



## Atividades

- Visa criar ou produzir uma mudança de estado visível em um ou mais artefatos durante a execução de um projeto

# ELEMENTOS CORRELATOS ÀS ATIVIDADES

---

## Artefatos

- Entrada
- Saída

## Papeis

- Responsáveis
- Participantes

## Recursos

- Consumíveis
- Não consumíveis

# DETALHAMENTO DE UMA ATIVIDADE

---

## Passos

- Uma descrição, que deve dizer em palavras simples e diretas o que deve ser feito para que a atividade seja realizada
- Como cada um dos artefatos de saída são produzidos a partir dos artefatos de entrada

## Procedimentos

- Realização tecnológica para o passo essencial definido
- Explicação adicional à atividade, o qual indica como realizá-la com as ferramentas e tecnologia disponíveis

## Regras

- Restrições sobre como as atividades podem ser executadas
- Estabelecem limites

# TEMPLATE DA DESCRIÇÃO DE UMA ATIVIDADE

Processo:	<nome do processo>
Fase:	<número e nome da fase>
Atividade:	<número e nome da atividade>
Versão:	<histórico de versões do documento>
Responsável: (obrigatório)	<cargo ou pessoa>
Participantes (opcional)	<cargo ou papel 1> <cargo ou papel 2> ...
Entradas: (opcional)	<artefato 1> <artefato 2> ...
Saídas: (obrigatório)	<artefato 1> <artefato 2> ...
Recursos: (opcional)	<recurso 1> <recurso 2> ...
Passos:	
<passo 1>:	<descrição> <procedimento segundo tecnologia 1> <procedimento segundo tecnologia 2> ... <ul style="list-style-type: none"> <li>• &lt;regra 1&gt;</li> <li>• &lt;regra 2&gt;</li> </ul> ...
<passo 2>:	<descrição> <procedimento segundo tecnologia 1> <procedimento segundo tecnologia 2> ... <ul style="list-style-type: none"> <li>• &lt;regra 1&gt;</li> <li>• &lt;regra 2&gt;</li> </ul> ...
...	...

# EXEMPLO

Processo:	MP – Meu Processo
Fase:	1. Concepção
Atividade:	1.3 Captura de Requisitos a Partir das Entrevistas
Versão:	1.0 inicial
Responsável:	Analista de requisitos
Participantes	-
Entradas:	1. Transcrição de entrevistas com o cliente. 2. Sumário executivo do projeto. 3. Definição de escopo do projeto.
Saídas:	1. Documento de requisitos iniciais.
Recursos:	1. <i>Template</i> de documento de requisitos. 2. Ferramenta CASE (EA v6.0 ou VP v.8.3).

Subprocesso

Passos:	
1.	<p>Listar requisitos funcionais candidatos.</p> <p>EA v6.0: Criar diagrama de requisitos e criar uma caixa para cada requisito candidato preenchendo o texto do requisito no campo "description"</p> <p>VP v8.3: Criar um diagrama de requisitos e uma classe estereotipada como &lt;&lt;requirement&gt;&gt; para cada requisito, preenchendo o texto do requisito no atributo "text", e preenchendo o atributo "kind" com "functional".</p> <ul style="list-style-type: none"> <li>Numerar os requisitos funcionais como RF01, RF02, ...</li> <li>Iniciar sempre com verbo no infinitivo.</li> </ul>
2.	<p>Listar requisitos suplementares e não funcionais.</p> <p>EA v6.0: Criar requisitos suplementares em um pacote separado dos funcionais. Indicar os requisitos não funcionais após o texto do requisito funcional associado indicado pela marca "RESTRIÇÕES:".</p> <p>VP v8.3: Criar requisitos suplementares em um pacote separado. Criar requisitos não funcionais como classes estereotipadas do diagrama com atributo "kind" preenchido com o tipo do requisito (interface, segurança, ...)</p> <ul style="list-style-type: none"> <li>Associar requisitos não funcionais a algum requisito funcional.</li> <li>Classificar requisitos suplementares pelo seu tipo: interface, segurança, tolerância a falhas, performance, etc.</li> <li>Não criar requisitos desnecessários.</li> </ul>
3.	<p>Agrupar requisitos funcionais em pacotes.</p> <ul style="list-style-type: none"> <li>Não permitir que mais de 20 requisitos estejam em cada pacote, a não ser em casos que se trate efetivamente de requisitos altamente coesos.</li> <li>Agrupar os requisitos em pacotes por afinidade, ou seja, requisitos mais próximos são aqueles que tratam dos mesmos objetos.</li> <li>Requisitos do tipo inserir, alterar, remover e consultar, sobre um objeto devem ser agrupados em um único requisito "manter" estereotipado como &lt;&lt;crud&gt;&gt;.</li> </ul>
4.	<p>Gerar o documento de requisitos.</p> <p>EA v6.0: Usar o gerador de documentação acessível a partir do menu superior.</p> <p>VP v8.3: Usar a opção "generate report" disponível no menu superior.</p> <ul style="list-style-type: none"> <li>Deve ser gerada uma versão pdf para impressão e uma versão html que ficará online na intranet do projeto.</li> </ul>

# A DESCRIÇÃO ESTÁ CLARA O SUFICIENTE?

---

- O descritivo de uma atividade não deve ser detalhado a ponto de ser cansativo para um analista que tenha alguma noção do que está fazendo.
- Porém, também não pode ser tão genérico a ponto de que dois analistas produzam resultados totalmente diferentes a partir dele.
- É necessário que cada passo esteja claramente definido, e quem vai determinar se está claro o suficiente são as pessoas que vão usar esta descrição de atividade.
- O documento de processo não é estático, ele pode evoluir.

# NOTAÇÕES PARA DEFINIÇÃO DE PROCESSOS

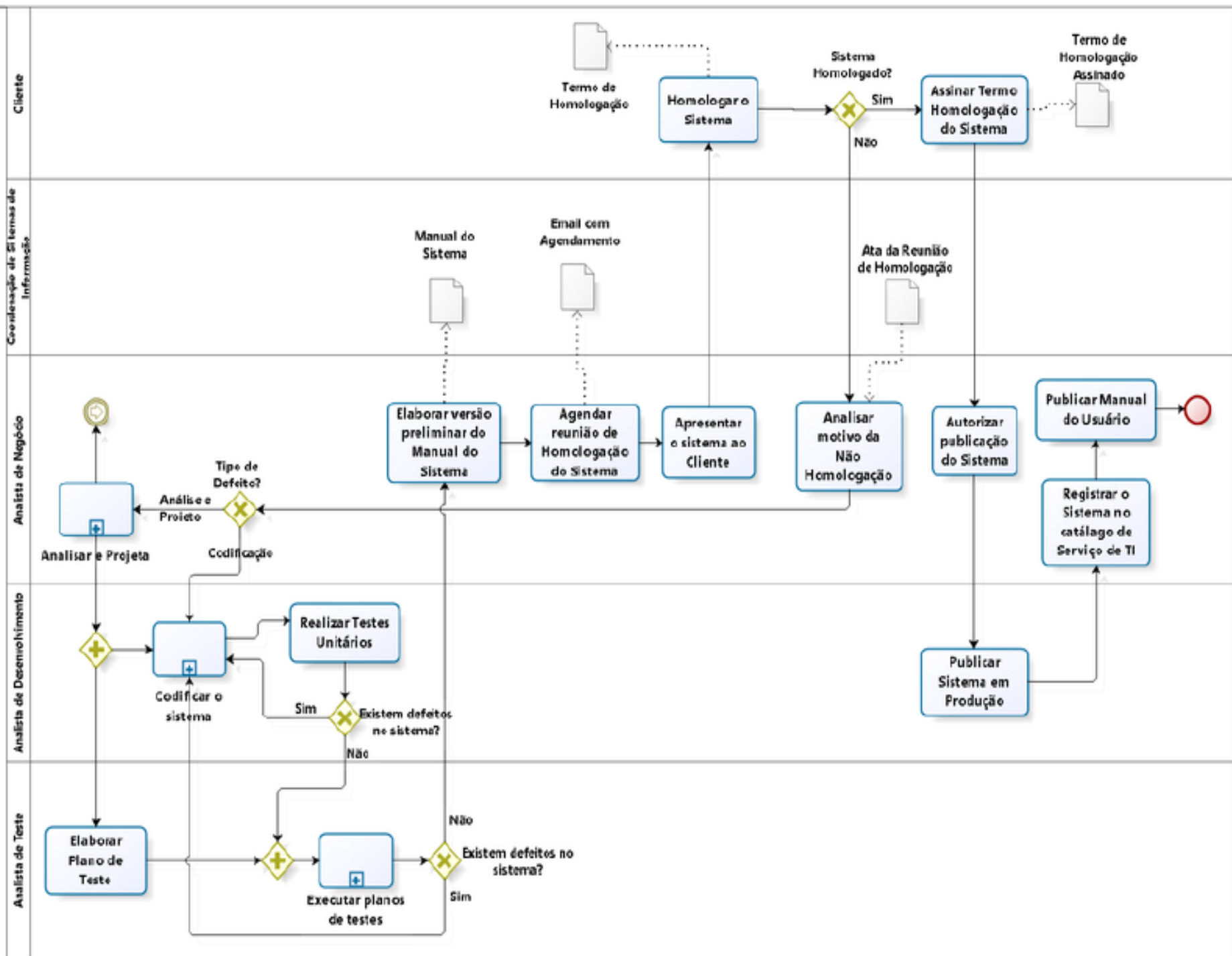
---

As notações para definir processos de software incluem:

- listas textuais de atividades e tarefas descritas em linguagem natural;
- diagramas de fluxo de dados;
- gráficos de estado;
- BPMN (*Business Process Modeling Notation*);
- Diagramas de atividades da UML.



## Processo de Desenvolvimento de Software do NTI/UFAC



# SUBPROCESSOS

---

- Um processo de software pode incluir **subprocessos**.
- Por exemplo, a **validação de requisitos de software** é um processo usado para determinar se os requisitos fornecerão uma base adequada para desenvolvimento de software.
- É um **subprocesso** do **processo de Engenharia de Requisitos**.
- As **entradas** para validação de requisitos são normalmente a especificação de requisitos de software e os recursos necessários para realizar validação (pessoal, ferramentas de validação, tempo suficiente).

# SUBPROCESSOS

---

- As **tarefas** da atividade de **validação de requisitos** podem incluir:
  - análises de requisitos;
  - prototipagem;
  - validação do modelo.
- Essas tarefas envolvem **atribuições de trabalho** para indivíduos e equipes.
- A **saída** da validação de requisitos é normalmente uma especificação validada de requisitos de software que fornece entradas para os processos de **design do software** e **teste de software**.

# CATEGORIAS DE PROCESSOS DE SOFTWARE

---

- **Processos primários**

- processos de software para desenvolvimento, operação e manutenção de software.

- **Processos de apoio**

- são aplicados intermitentemente ou continuamente ao longo de um ciclo de vida do produto de software para oferecer suporte a processos primários;
- incluem processos de software, como:
  - gerenciamento de configuração, garantia de qualidade e verificação e validação.

# CATEGORIAS DE PROCESSOS DE SOFTWARE

---

- **Processos organizacionais**
  - fornecem suporte para a engenharia de software;
  - incluem treinamento, análise de medição de processo, gerenciamento de infraestrutura, portfólio e reutilização, melhoria de processos organizacionais e gerenciamento de modelos de ciclo de vida de software.
- **Processos de projetos cruzados**
  - envolvem mais do que um único projeto de software em uma organização;
  - Incluem: reutilização, linha de produtos de software e engenharia de domínio.

# CATEGORIAS DE PROCESSOS DE SOFTWARE

---

- **Processos de gerenciamento de projetos**
  - incluem processos de planejamento e estimativa, gestão de recursos, medição e controle, liderança, gestão de riscos, gerenciamento de stakeholders e coordenação de processos primários, de apoio, organizacionais e de projetos cruzados.

# COMO DEFINIR UM PROCESSO DE SOFTWARE?

## NORMA NBR ISO/IEC 12207:2008

---

- A norma estabelece **processos, atividades e tarefas** que devem ser aplicados durante a **aquisição, fornecimento, desenvolvimento, operação, manutenção e descarte de software**.
- A ISO/IEC 12207 e suas adaptações apresentam definições e conceitos que são **independentes do ciclo de vida** escolhido e, portanto, podem ser aplicadas a variados contextos.

# COMO DEFINIR UM PROCESSO DE SOFTWARE?

---

- É importante destacar que não há um processo de software melhor.
- Processos de software devem ser selecionados, adaptados e aplicados conforme apropriado para cada projeto e cada contexto organizacional.



# EQUIPE DE PROCESSO

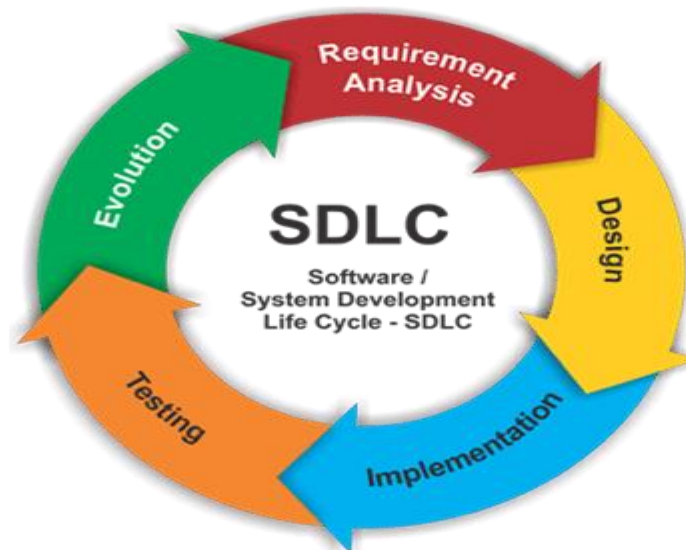
---

- Um ou mais engenheiros de software que serão responsáveis pela manutenção, avaliação e otimização do processo.
- O tamanho do grupo deveria variar entre 1 a 3% do número de profissionais da empresa ligados ao desenvolvimento de software.
- Ele centraliza e capitaliza o esforço colaborativo dos mais diferentes agentes no sentido da **melhoria contínua do processo** adotado na empresa.

# CICLO DE VIDA DE DESENVOLVIMENTO DE SOFTWARE

---

Um **ciclo de vida de desenvolvimento de software** (*software development life cycle - SDLC*) **inclui** os **processos de software** usados para especificar e transformar os requisitos de software em um produto de software.



# PROJETO – PROCESSO – MODELO DE PROCESSO

---

## Projeto

- Consiste na execução concreta de um conjunto de atividades que visam à criação de um produto específico
- Ocorre em um tempo determinado

## Processo

- Um conjunto de regras que definem como um projeto deve ser executado

## Modelo de processo

- Conjunto de regras mais abstratas que especificam a forma geral de processos
- Apresenta uma filosofia, uma forma geral de comportamento, baseada na qual processos específicos podem ser definidos

# MODELO DE PROCESSO DE SOFTWARE

---

Processos de software são construídos de acordo com **modelos** (estilos)

Um **modelo de processo de software** é uma representação abstrata de um processo.

- Apresenta uma descrição de um processo de alguma perspectiva particular

Famílias de Modelos de processo:

- **Prescritivos**
- **Ágeis**

WASLAWICK (2013)

Existem outras  
classificações na  
literatura!

# BIBLIOGRAFIA

