

## AULA 2 – ATIVIDADES DO PROCESSO DE SOFTWARE

### OBJETIVO DA AULA

Compreender quais são as atividades utilizadas no processo de software, apresentando quais são os objetivos de cada uma delas, bem como estabelecer quais são os artefatos que devem ser elaborados em cada processo.

### APRESENTAÇÃO

A construção de software deve considerar o estabelecimento de um processo de elaboração organizado, que deve ser avaliado conforme o objetivo e as características de cada projeto de software.

Esta aula apresenta as atividades essenciais para que o software seja desenvolvido sem que nenhuma fase, seja “esquecida”, com o objetivo de aplicar as etapas fundamentais que devem ser aplicadas em qualquer processo.

As atividades básicas do processo de software são: especificação, desenvolvimento, validação e evolução. É importante ressaltar que elas são conduzidas e organizadas de maneira diferente de acordo com o processo de desenvolvimento escolhido. A correta aplicação das mesmas certamente facilitará o processo de desenvolvimento, tanto no quesito de completude, quanto o de busca pela qualidade.

### 1. PROCESSO DE SOFTWARE

Em Engenharia de Software devemos escolher o processo de software que deverá ser implementado, ou seja, o método que utilizaremos no desenvolvimento do produto como, por exemplo, o processo em cascata ou incremental.

Antes disso, precisamos conhecer o que chamamos de atividades ou disciplinas do processo de software. Essas disciplinas são normalmente organizadas em 5 atividades básicas: especificação, desenvolvimento, validação e evolução.

Embora a literatura apresente variações quanto a essa organização de atividades, é importante destacar que elas serão facilmente definidas como etapas da maioria dos processos de software que estudaremos. O que muda é a forma como essas disciplinas são organizadas e implementadas.

Os Processos de Software são as fases indispensáveis para produzir e manter um software. Cada atividade possui uma organização de suas atividades técnicas e gerências, e envolvem métodos, ferramentas, artefatos e regras que possibilitam sistematizar e organizar o desenvolvimento e manutenção de produtos de software.

Essas atividades visam definir:

- Os artefatos (qualquer produto gerado no desenvolvimento de um software, como Códigos-fonte, diagramas, documentos que serão desenvolvidos);
- Quando e como eles serão desenvolvidos;
- O perfil de quem irá desenvolvê-los;
- A razão do desenvolvimento de cada artefato;
- Onde os artefatos serão desenvolvidos.

Além de ajudar na organização e sistematização do desenvolvimento do software, ao final de cada atividade os artefatos também podem ser utilizados como documentação final do software.

É certo que a correta aplicação das atividades do processo de software agregarão incontáveis benefícios ao projeto como um todo. É notório que os principais benefícios percebidos dizem respeito à clareza da equipe desenvolvedora sobre o software porque todos os membros da equipe são capazes de compreender o que deve ser feito ao longo do desenvolvimento do produto, além de também saberem o papel de cada desenvolvedor no projeto.

Vale também ressaltar o crescimento da equipe naquele processo de desenvolvimento, pois cada vez que o processo é repetido, mais assertivo a equipe ficará em termos de custo, organização do cronograma e definição dos prazos de entrega.

O uso das atividades de um processo de software torna-se fundamental sempre que for necessária a construção de um software, buscando a sua qualidade e a sua continuidade para que seja prolongada a vida útil do produto. Sendo assim, vamos conhecer as tarefas e ferramentas destinadas a cada atividade.

## 2. ESPECIFICAÇÃO

É a atividade aplicada à identificação das necessidades do software. Conhecer o que o sistema faz e o que como o usuário terá o seu trabalho afetado por ele.

Também conhecida como engenharia de requisitos, a atividade de especificação visa compreender e identificar os requisitos do sistema. Um requisito é uma característica que o sistema deve contemplar, uma funcionalidade que o sistema deve possuir para satisfazer os contratos e especificações feitas pelo cliente ou usuário.

Nesta disciplina detalhamos cada requisito para conhecer a totalidade de funcionalidades e restrições que um software deve possuir. Esta é uma fase crítica, e necessita de que o analista de sistemas se aproprie de cada detalhe, cada processo que deverá ser implementado para que a elaboração do software seja bem-sucedida. É importante destacar que um erro nessa fase é crítico, pois impactam em toda a cadeia de construção do software.

Artefatos são produtos gerados a partir de um trabalho realizado em uma fase da atividade de processo de software. Nesta primeira atividade, um artefato importante é a geração de um documento chamado de documento de requisitos, que reúne todos os requisitos identificados para satisfazer a solicitação do cliente.

Existem quatro atividades no processo de engenharia de requisitos:

- Estudo da viabilidade;
- Elicitação e análise de requisitos;
- Especificação de requisitos;
- Validação de requisitos.

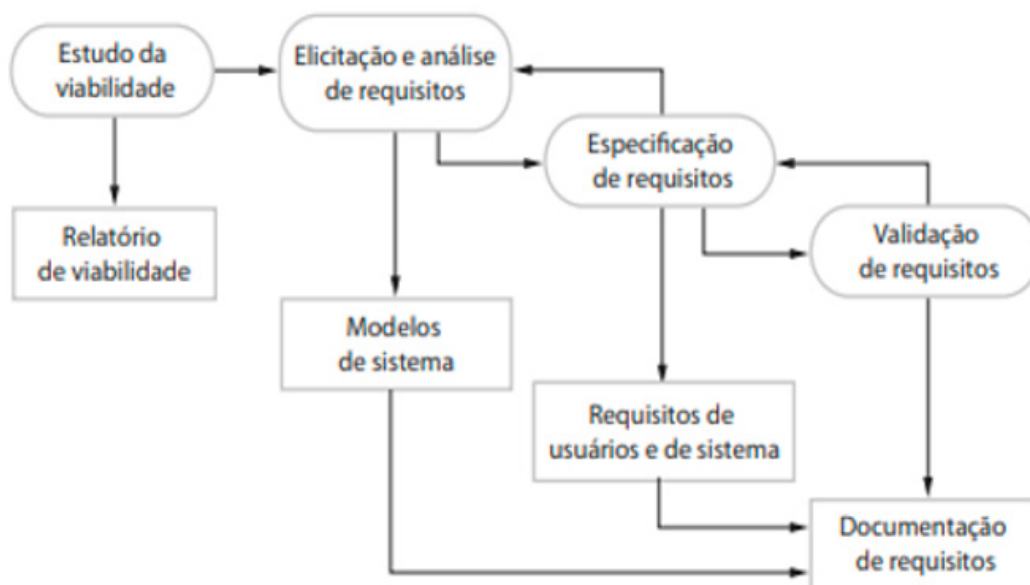
No estudo da viabilidade é realizado um levantamento sobre questões como restrições orçamentais e entendimentos sobre se a construção do sistema deve prosseguir, ou não, considerando o interesse da parte do cliente e da parte da empresa ou equipe de desenvolvimento.

Na elicitação e análise, todos os requisitos são identificados, resultando no entendimento do que o sistema deverá ser capaz de executar. É possível usar protótipos do sistema para ajudar no entendimento da totalidade dos requisitos identificados.

Na especificação de requisitos são detalhados os requisitos identificados, produzindo uma lista com a especificação de cada requisito, tanto no contexto de sua funcionalidade quanto nas restrições de hardware e software pertinentes e necessários para a sua execução.

Por fim, na validação, os requisitos são verificados quanto a sua consistência e completude, nesse momento deve-se atualizar o documento de requisitos para que os erros encontrados sejam resolvidos e integrados na versão final do documento.

**FIGURA 1 | Os Requisitos da Engenharia de Processos**



Fonte: SOMMERVILLE, 2019.

Ainda sobre os artefatos gerados nessa fase, podem ser criados Diagramas Entidade Relacionamento (DER), para especificação do banco de dados que irá compor o sistema, e também o Diagrama de Caso de Uso da UML (linguagem unificada de modelagem), que possibilitará exibir o relacionamento do usuário com cada funcionalidade do sistema. Outra ferramenta da UML que pode ser útil é o diagrama de atividades que permite o detalhamento do passo a passo na execução de cada requisito.

### 3. DESENVOLVIMENTO

A atividade de desenvolvimento envolve o projeto e a implementação do software. Esta fase requer a tradução de tudo o que foi feito na fase anterior em sistemas executáveis.

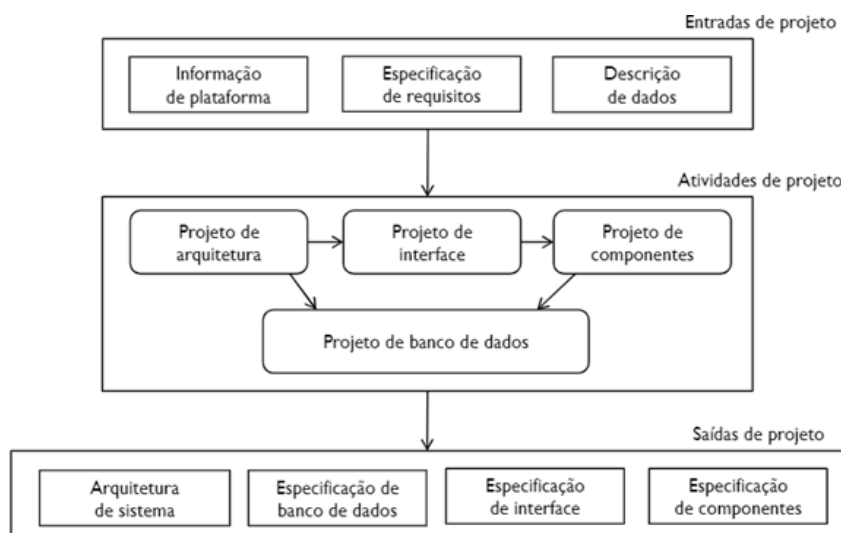
Em termos de projeto, deve-se descrever a estrutura geral do software, incluindo arquitetura (linguagem de programação a ser utilizada, por exemplo), interfaces, algoritmos e banco de dados necessários para implementar o sistema.

Nesse ponto é importante observar os padrões de implementação que são normalmente definidos pela empresa. Alguns exemplos dessas definições são modelos de identificação, padronização na nomenclatura de variáveis e objetos de banco de dados, definição de comentários no código, organização na criação das funções, procedimentos, *triggers* e tudo mais.

Na implementação, independente da linguagem adotada, é importante considerar que todo sistema precisa ter continuidade e para isso é preciso que sejam considerados aspectos de manutenibilidade e compreensão.

Nesta fase, os artefatos gerados normalmente são o Diagrama de Estrutura Modular (DEM) para visualização dos módulos, submódulos e empacotamento, apresentando visualmente a definição dos programas a serem desenvolvidos. No caso do banco de dados, a modelagem dos dados é convertida para instruções de banco de dados, normalmente em linguagem SQL. Pode-se também refinar os diagramas entidade relacionamento (DER) e/ou os diagramas de classes e de máquina de estados da UML.

FIGURA 2 | **Um Modelo Geral do Processo do Projeto**



A Figura 2 mostra os artefatos produzidos nesta atividade, além de especificar as saídas após a finalização desta fase de construção do software.

É importante conhecer quais são as tarefas de cada parte do projeto:

- No projeto de arquitetura definimos a arquitetura geral do sistema, como os seus módulos ou subsistemas;
- No projeto de interface estabelecemos aspectos de como os componentes do sistema se comunicarão, ou seja, um componente se comunica com o outro sem precisar saber como ele foi implementado;
- No projeto de componentes deve-se projetar o funcionamento de cada parte do sistema;
- No projeto de banco de dados é especificada a camada de dados do sistema e as suas estruturas.

Ao fim desta atividade temos um produto que ainda carece de validação e testes para que seja colocado em produção.

## 4. VALIDAÇÃO

Também chamada de atividade de verificação e validação, esta fase tem o objetivo de apurar se o sistema construído atende às especificações do cliente, avaliando os resultados da execução de cada operação previamente definida.

Nesta atividade são aplicados testes de software, que apresentam cenários capazes de simular a operação do sistema, avaliando se o mesmo atinge os resultados esperados.

Aqui são verificados e validados os procedimentos desenvolvidos para garantir a qualidade do software. É preciso verificar a presença de erros e se os requisitos estão sendo atendidos de acordo com a necessidade dos usuários.

Além dos testes, existem outros tipos de verificação do software, como inspeções e revisões, que podem ocorrer não apenas nesta atividade, mas também ao longo de todas as atividades anteriores.

Na validação passamos por testes de componente, de sistema e de aceitação. Estes testes visam garantir o comportamento do sistema em três estágios, para que o sistema não seja testado como uma unidade única, mas como um conjunto integrado.

Aqui o desenvolvimento é finalizado e disponibilizado para uso no usuário. Os procedimentos são aprovados pelos *stakeholders*, e na maioria das vezes um treinamento é realizado com os usuários do negócio, pois o usuário precisa saber manusear o sistema computacional e conhecer como o computador irá atender suas necessidades.

Ao final desta atividade, o sistema está pronto para o chamado teste beta, que significa que o sistema está pronto para ser utilizado comercialmente. Neste teste alguns usuários fazem uso do produto no dia a dia, revelando aos desenvolvedores possíveis erros ou problemas encontrados. Após os devidos ajustes, o sistema enfim está liberado para ser comercializado.

## 5. EVOLUÇÃO

Esta última atividade tem o objetivo de garantir que o sistema possa ser flexível ao ponto de se ajustar às mudanças de requisitos, legislação e demais alterações que determinado domínio pode exigir.

A evolução também é conhecida como manutenção de software, e ela exige grande esforço da equipe desenvolvedora, mas esse esforço pode ser atenuado se determinados padrões de qualidade, como uma boa documentação, por exemplo, tenham sido seguidas ao longo da gestão do produto.

Outro ponto crítico aqui é a aferição da qualidade, pois serão verificadas se as características necessárias para satisfazer a necessidade dos usuários do negócio forma atendidas. Essas características estão relacionadas a todos os aspectos do sistema como complexidade, recursos envolvidos, metodologias e técnicas aplicadas, além de procedimentos associados a detecção e remoção de erros e, a relação custo-benefício. Essas medições podem ser quantitativas e qualitativas.

## CONSIDERAÇÕES FINAIS

Nesta unidade abordamos algumas das atividades essenciais no processo de software. Vimos que processos de software são as fases indispensáveis para produzir e manter um software, onde cada atividade possui uma organização, técnicas e resultam em métodos, ferramentas, artefatos e regras que possibilitam sistematizar e organizar o desenvolvimento e manutenção de produtos de software.

As atividades que listamos nesta unidade foram especificação, desenvolvimento, validação e evolução, e cada uma delas exerce um papel importantíssimo na construção do software.

Na especificação cuidamos do entendimento do problema, bem como de toda a gestão de seus requisitos. Após esse levantamento inicial, damos sequência ao desenvolvimento, que é quando estabelecemos o projeto e iniciamos o processo de desenvolvimento do software.

Ao final deste desenvolvimento tratamos dos testes e validação do produto, para que enfim ele seja colocado em produção. Na evolução cuidamos da existência do software, cuidando de sua manutenção, no sentido de reagir às mudanças de requisitos, legislação ou no ajuste de suas funções.

## MATERIAIS COMPLEMENTARES

Artigo: *Processos de Software: o que você precisa saber*. 2018. Disponível em: <https://medium.com/@danielemsilva/processos-de-software-o-que-voc%C3%AA-precisa-saber-9b-89c359d3e7>. Acesso em: 01 nov. 2022.

## REFERÊNCIAS

KENNETH C. LAUDON; JANE P. LAUDON. *Sistemas de informação gerenciais*. 11ª edição. Editora Pearson, 2014.

PRESSMAN, R. G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison. Wesley, 2019.