

NÍVEL	Graduação
DISCIPLINA	Engenharia de <i>Software</i>
AULA	01
PROFESSOR	Maristela Regina Weinfurter Teixeira

Rota Conversa Inicial

Olá aluno! Seja bem-vindo ao primeiro encontro da disciplina **Engenharia de Software**.

Em nossa primeira aula, iremos abordar conceitos importantes para a compreensão do que é a chamada engenharia de *software*. Estabeleceremos uma linha de raciocínio sobre processos, metodologias, técnicas e ferramentas utilizadas nas várias etapas do desenvolvimento de *software*.

A professora que irá nos acompanhar durante os estudos chama-se Maristela Regina Weinfurter Teixeira, atua na área de tecnologia de informação há mais de vinte e cinco anos, tanto na área empresarial quanto acadêmica.

Acompanhe o primeiro vídeo da nossa aula no material *online*, a professora Maristela nos apresentará os temas que estudaremos a seguir!

Contextualizando

O *software* foi incorporado em praticamente todos os contextos de nossa vida. A engenharia de *software* surge em meados dos anos 70, com o intuito de responder algumas questões como as disponíveis no botão a seguir. Clique para conferir!

- Por que levamos tanto tempo para concluir um *software*?
- Por que nossos custos sempre são tão altos?
- Por que não conseguimos encontrar todos os erros antes da entrega?
- Por que gastamos tempo e esforço para manutenção de *software*?

- Por que continuamos com dificuldades na medição do progresso do desenvolvimento de *software*?

Você deve estar se perguntando se ainda ocorrerão perguntas e problemas que envolvam a engenharia de *software*, não?

Com certeza, ainda temos muito o que aprender até perceber que o processo de desenvolvimento de *software* deve ser considerado como o processo de fabricação de qualquer outro tipo de produto. Claro, observando-se que produzimos algo intelectual e intangível.

A engenharia de *software*

A engenharia de *software* surgiu em meados dos anos 70 como uma forma de contornar a chamada crise do *software*. Tal crise ocorria em decorrência da complexidade no desenvolvimento, além de problemas relacionados a prazos, custos e qualidade. Os fundamentos científicos da engenharia de *software* envolvem o uso de métodos, modelos, técnicas, processos e ferramentas que possam ser utilizados durante todas as fases de desenvolvimento de *software*. Além de oferecer mecanismos para planejamento e gerenciamento de projeto de desenvolvimento de *software* e garantias de aplicações de aspectos de qualidade aos processos.

O foco da engenharia de *software* compreende todos os aspectos da produção de *software*, desde seu estágio inicial de especificações até a fase de manutenção do *software*.

Software é um conjunto de programas de computador, que executados, fornecem funções e desempenhos desejados. O *software* é desenvolvido, não é fabricado no sentido clássico.

Confira alguns tópicos nos quais a engenharia de *software* é aplicada!

- Sistemas de informação
- Aplicações científicas
- Aplicações para engenharias
- Aplicações embutidas
- Aplicações Web
- Aplicações para inteligência artificial

Estabelece o emprego de princípios das engenharias para obtenção de um *software* confiável e econômico. Segundo a IEEE, a engenharia de *software* é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de *software*.

Sua base é obtida através da camada de processos. Todas as camadas utilizam ferramentas e métodos com foco na melhoria contínua dos processos e qualidade, conforme esquema a seguir:



O processo de engenharia de *software* conecta as camadas de tecnologia e possibilita o desenvolvimento de um *software*. Métodos fornecem técnicas para o desenvolvimento de *software*. Um método é composto por várias tarefas que compreendem modelagem, análise, comunicação, implementação, teste e suporte. As ferramentas são utilizadas para automatização da construção de *software* através de processos e métodos escolhidos.

Agora, acompanhe no material *online* as explicações da professora Maristela sobre o tema que acabamos de estudar! Preste bastante atenção!

Histórico

Para compreendermos melhor o drama caracterizado pelo que chamamos de crise do *software*, há uma figura clássica que circula há anos na Internet. Mudam as tecnologias, mudam as gerações de *stakeholders* e desenvolvedores e continuamos com problemas similares. Compreender o

que nossos *stakeholders* precisam é uma tarefa árdua e não provém do conhecimento de tecnologias, mas sim da área de humanas.

O termo engenharia de *software* torna-se forte após 1968, quando sistemas se tornam cada vez mais complexos. Dá-se início a utilização de métodos, técnicas e ferramentas. A crise do *software* é relatada através de alguns sintomas:

- Atraso na entrega.
- Orçamento estourado.

- Falhas de *software*.
- Custos elevados.
- Requisitos mal elaborados.
- Baixa qualidade do *software*.
- Gerenciamento sem indicadores.

Um detalhe não menos importante é que antes tínhamos um programador solitário resolvendo todos os problemas de seus programas espalhados pela organização e sem nenhum tipo de integração. Hoje, o cenário é constituído por um time de desenvolvedores e um time de *stakeholders*

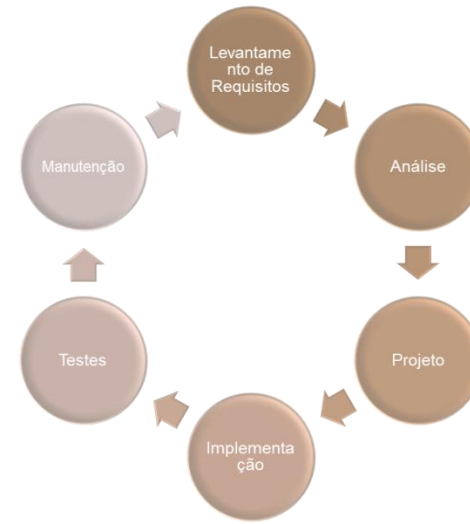
trabalhando em sistemas complexos e dentro de um ambiente de diferentes plataformas altamente conectados.

Assista à videoaula com as explicações trazidas pela professora Maristela que está disponível no material *online!*

O ciclo de vida de sistemas

O desenvolvimento de um sistema segue um ciclo de vida, sabemos que um sistema é algo

dinâmico e que muda constantemente. Esta criação ou mudanças ocorrem segundo um ciclo de vida que é observado.



As fases importantes neste ciclo de vida são levantamento de requisitos, análise, projeto, implementação, testes e manutenção. Com a manutenção surgem novos requisitos e o ciclo ocorre infinitamente enquanto o *software* existir.

No levantamento e análise de requisitos identificamos e avaliamos as necessidades reais para construção ou mudanças no *software*. Na fase de projeto, partimos para o detalhamento das características importantes para a construção do *software*. Neste momento, são definidos aspectos relacionados a banco de dados, sistemas, hardware, processamento e redes. A implementação abre início ao processo de codificação em uma linguagem de programação sobre uma determinada plataforma para trazer do abstrato para o concreto nosso produto. Testes promovem a verificação e validação de sistemas para garantir qualidade do processo e do *software*. Na manutenção encontraremos erros que ainda não foram eliminados, bem como melhorias em relação ao *software* em questão. Este ciclo de vida segue modelos de processos que trabalharemos posteriormente. Entre eles encontram-se o modelo clássico, espiral, prototipagem e ágil.

Assista ao vídeo com as explicações da professora Maristela no material *online* e aproveite para revisar o conteúdo visto!

Tema 4 Metodologias, técnicas e ferramentas

Antes de falarmos sobre metodologias, técnicas e ferramentas, é importante refletirmos sobre algumas questões importantes:

- Por que os custos de desenvolvimento são tão altos?
- Por que não achamos os erros antes da entrega?
- Por que gastamos tanto tempo e esforço mantendo programas existentes?
- Por que temos dificuldades em avaliar o progresso de um projeto?

São as questões que deram origem à engenharia de *software* e que permanecem em vários casos sem resposta até nossos dias atuais.

Um princípio básico da engenharia de *software* é utilizarmos métodos, técnicas e ferramentas como auxílio e automação do processo de desenvolvimento de *software*. O método é um caminho para a realização de alguma tarefa. A técnica é um conjunto de procedimentos utilizados para realização de uma ou mais atividades. Ferramentas são instrumentos manuais ou digitais que realizam determinadas tarefas.

Este é o contexto da engenharia de *software* que iremos estudar até nosso último encontro. Como desenvolvermos *software* com orçamentos adequados, prazos bem estabelecidos, qualidade e melhoria de processo contínuo, gerenciando um time de profissionais multidisciplinar e entregando um produto final que supere as expectativas de nossos *stakeholders*.

Agora, não deixe de conferir a explicação da professora Maristela, disponível no material *online*!

Trocando ideias

Você já parou para refletir sobre questões como a falta de cumprimento de prazos, custos e *software* de qualidade ruim? Se estamos desde o final dos anos 60 falando sobre engenharia de *software*, por que ainda não estamos bons no desenvolvimento de *software*?

Aproveite para discutir com seus colegas de curso, através do fórum da disciplina, sobre quanto tempo precisamos para amadurecimento de ideias na sociedade e nas empresas como um todo. Isto aplicando-se para qualquer área.

Na Prática

Com base em tudo o que você estudou até agora, faça o seguinte exercício: busque em alguns livros e revistas que abordem o tema engenharia de *software* e observe questões cruciais como: crise do *software*, ciclo de vida, métodos, técnicas e ferramentas no contexto de desenvolvimento de *software*.

Síntese

Nessa aula, vimos o surgimento da área de engenharia de *software*, suas preocupações, seu contexto e o que estudaremos até o final da disciplina. Sempre é importante lembrar:

- Há um ciclo de vida para os sistemas;
- Faz-se necessário o uso de métodos, técnicas e ferramentas para o desenvolvimento de *software*;
- *Software* é algo que muda constantemente.

Até a próxima aula!