



Análise e Projeto de Software

Universidade Evangélica de Goiás

Curso de Engenharia de Software

Organização da Disciplina

- **Objetivo:**

Tem como objetivo explorar os métodos, técnicas e ferramentas utilizadas na análise e projeto de software, os princípios fundamentais e práticas do gerenciamento ágil de projeto, alinhando a análise orientada a objetos, a notação UML, os padrões de projeto e a metodologia ágil.

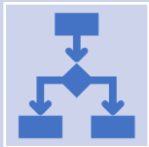
Organização da disciplina - Cronograma



Horário: 19:00 – 22:40



Carga-horária: 80 hrs



Intervalo: 20:40-21:00

Postura em sala de aula

Sugestões e dicas:

- Acompanhe os slides e faça suas anotações, em algum momento são apresentadas algumas questões de prova!!!
- Seja pontual
- Tente relacionar conceitos da aula com a sua prática profissional
- Quanto mais rápido viver a filosofia de “Aprender a aprender” melhor,
- mantra!
- Se conheça! faça um teste de personalidade

Organização da prova/Trabalhos

Prova Teórica (on-line): com valor 0 a 60 pontos

- Prova de caráter avaliativo a ser aplicada através do AVA contendo 22 questões, sendo 2 discursivas e 20 objetivas que abrangem os conceitos abordados em aula durante o Ciclo 1.

Avaliações processuais: com valor 0 a 40 pontos

- **Atividade Prática Supervisionada - APS:** com valor 0 a 9 pontos.
 - Atividade semanal individual composta por 4 questões objetivas relacionadas ao conteúdo abordado na aula.
 - Contemplada nas semanas 01 a 06 (6 semanas) com valor de 0 a 1,5 pontos por atividade.
- **Projeto Integrativo - PI:** com valor 0 a 11 pontos.
 - Atividade, em grupo, composta por um envio de tarefa apresentando o artefato completo referente à parte do Projeto Integrativo.
 - Contemplada na semana 06 com valor de 0 a 11 pontos.
 - O envio da tarefa deverá ser efetivado pelo aluno através do AVA até a semana 07 às 18h59.
- **Atividade Prática - AP:** com valor 0 a 20 pontos.
 - Atividade semanal, em grupo, composta por um envio de tarefa relacionado ao conteúdo abordado na aula.
 - Contemplada nas semanas XX a XX (X semanas) com valor de 0 a X pontos por atividade.
- **Prova OnLine:** com valor de 0 a 10 pontos extras
 - **Prova simulada com questões do ENADE.**



Organização da média final

Média final= $1VA + 2VA + 3VA / 3$ = aprovado
se $\Rightarrow 6,0$



Análise e projetos de Software: Histórico e evolução de engenharia de software

Introdução

Pois bem, quando falamos em engenharia já nos vem a noção de algo que deve ser criado, construído, analisado, desenvolvido e que se deve promover a manutenção.

No nosso dia a dia é, perfeitamente, possível encontrar exemplos da engenharia em edifícios, casas, ruas, estradas, hidrelétricas, veículos e tantos outros.



Histórico e evolução de engenharia de software

O termo Engenharia de Software foi criado pelo professor Friedrich Ludwig Bauer, hoje aposentado, na década de 1960, época em que ele lecionava na Universidade de Tecnologia de Munique, Alemanha, e, utilizado oficialmente em 1968 na Conferência sobre Engenharia de Software da OTAN (*NATO Conference on Software Engineering*).

Histórico e evolução de engenharia de software

Segundo o professor Bauer, “Engenharia de Software é a criação e a utilização de sólidos princípios da engenharia a fim de se obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em computadores reais”.



UniEVANGÉLICA
CENTRO UNIVERSITÁRIO



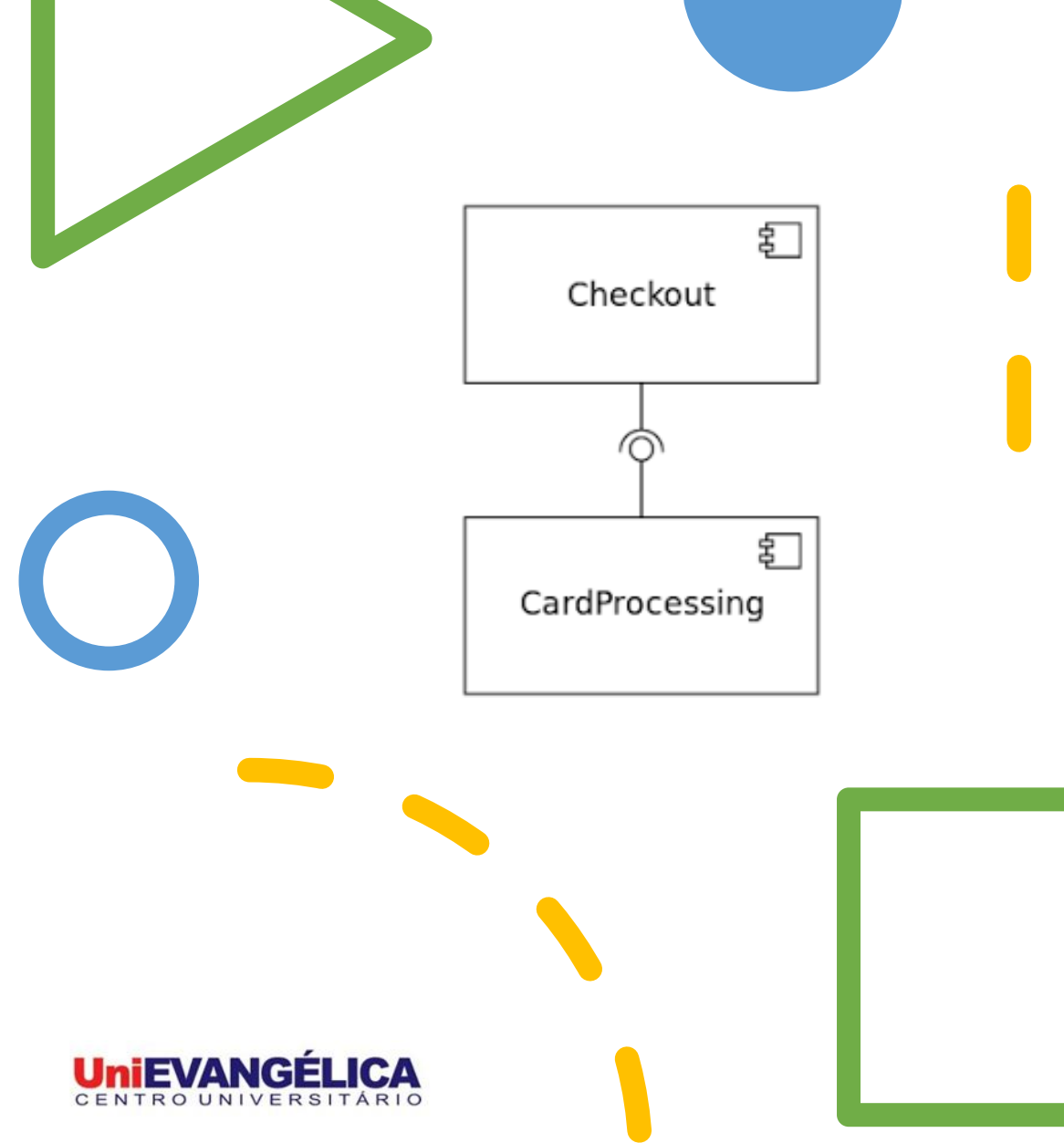
Histórico e evolução de engenharia de software

Desde então, este termo é amplamente utilizado para identificar a área do conhecimento da informática cujo objetivo é o de especificar (descrever), desenvolver e promover a manutenção em sistemas de software (programas de computador), aplicando tecnologias e boas práticas (práticas consideradas as mais adequadas) provenientes das disciplinas de ciência da computação e gerência de projetos, entre outras, levando, assim, a produtividade e qualidade.

Histórico e evolução de engenharia de software

Sendo assim, pode-se dizer que a engenharia de software preocupa-se com os aspectos da construção de um sistema de software, desde o início de sua especificação até a manutenção, após ter sido posto em operação. De acordo com Sommerville (2003), há duas frases que descrevem esse conceito:

- Disciplina de Engenharia
- Todos os aspectos da produção de software

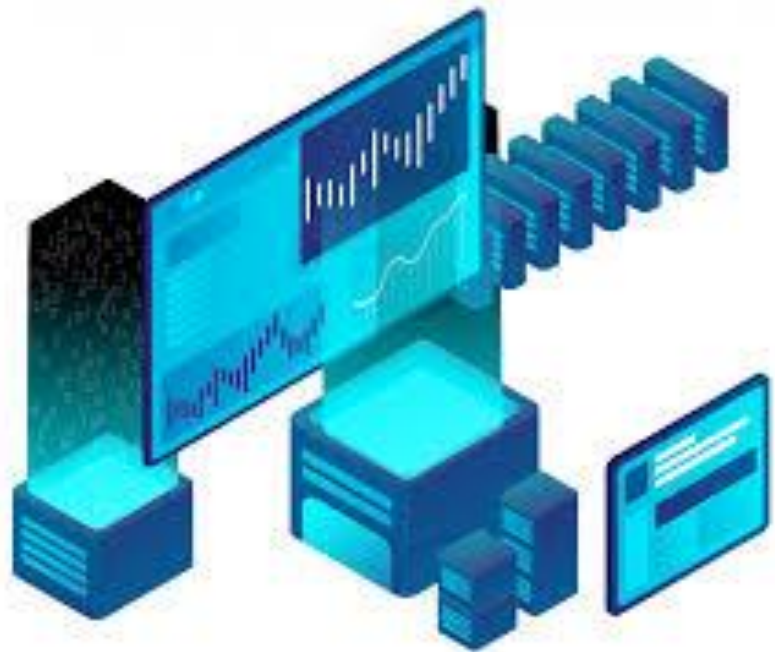


Histórico e evolução de engenharia de software

Disciplina: os engenheiros fazem uso de teorias, métodos e ferramentas para tornar os produtos úteis, procurando soluções para problemas, mesmo que não existam teorias aplicáveis e métodos de auxílio;

Todos os aspectos da produção de software: não somente os aspectos técnicos são importantes, as atividades de gerenciamento de projetos e desenvolvimento de ferramentas, os métodos e as teorias de apoio à produção de software também são questões importantes dentro da engenharia de software.





Histórico e evolução de engenharia de software

- Nos **anos 40** os primeiros usuários de computadores escreviam código de máquina (instruções executadas pelos computadores) à mão.
- As primeiras ferramentas de software, tais como, macro *assemblers* (montadores de código) e interpretadores, foram criados nos **anos 50** com o objetivo de melhorar a qualidade do código escrito e a produtividade dos programadores, surgindo, assim, a primeira geração de compiladores que aperfeiçoavam (otimizavam) o código de máquina.
- Nos **anos 60** essas ferramentas evoluíram ainda mais permitindo uma maior qualidade nos programas, bem como uma produtividade mais alta por parte dos programadores.



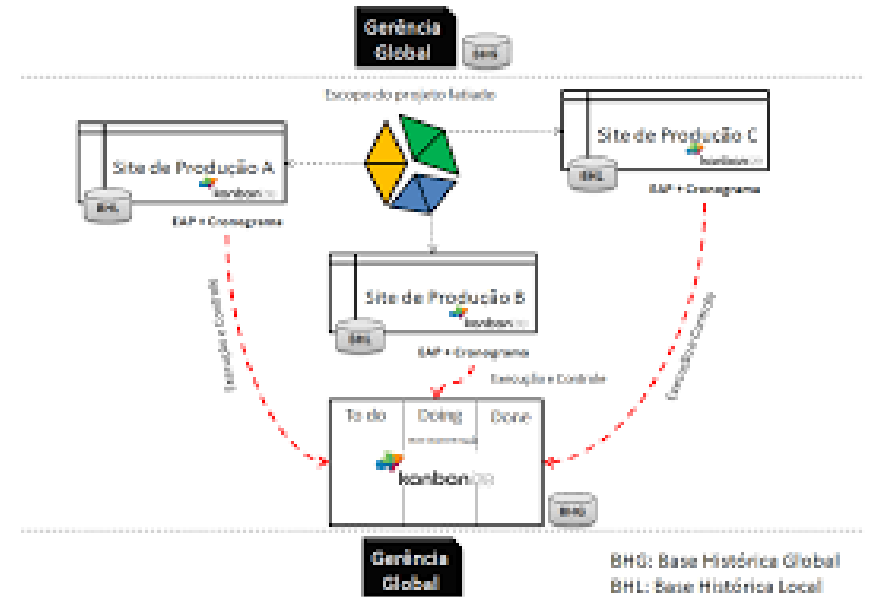
Histórico e evolução de engenharia de software

por um conjunto de componentes abstratos de software (estruturas de dados e algoritmos) descritos na forma de procedimentos, funções, módulos, objetos ou agentes interconectados entre si, compondo, assim, a arquitetura de software, e que devem ser executados em sistemas computacionais.

Histórico e evolução de engenharia de software

A partir de então, ferramentas colaborativas e repositórios de código apareceram na década de **1970**. Nos **anos 80** os computadores pessoais e as estações de trabalho (computadores com maior poder de processamento) tornaram-se comuns.

Os profissionais de desenvolvimento de software também viram surgir nesta época a primeira linguagem de programação orientada a objetos comercial, denominada **SMALLTALK**.



Histórico e evolução de engenharia de software

A programação orientada a objetos e os processos ágeis, como XP (*Extreme Programming* - Programação Extrema), na qual equipes pequenas ou médias desenvolvem software cujos requisitos não são claros ou que mudam constantemente, obtiveram uma maior aceitação nos **anos 90**.

Ainda nos **anos 90** os computadores de maior processamento tiveram uma queda acentuada de preços. A complexidade das aplicações de software cresceu mais com o avanço da tecnologia, a Internet e os dispositivos móveis permitiram que o software se tornasse cada vez mais disponível.





Histórico e evolução de engenharia de software

A partir de 2000 surgiu o código gerenciado e plataformas interpretadas, tais como, Java, NET, Ruby, Python e PHP, proporcionaram facilidades na criação de aplicações de software.



A engenharia de software B2B (*Business to Business*) e do B2C (*Business to Commerce*).

O conceito de Web 2.0, uma evolução no desenvolvimento de aplicação para Web, permitindo uma experiência de uso mais próxima as aplicações de software instaladas e executadas diretamente no computador do usuário e o conceito de Serviços Web (*Web Services*), no qual uma aplicação utiliza um serviço, um tipo de funcionalidade computacional, que foi criado por outra pessoa e que reside em algum computador conectado a Internet, são os mais recentes desafios da engenharia de software.

De acordo com Pressman, 2006, a engenharia de software é uma tecnologia em camadas. Todas as camadas, processos, métodos e ferramentas, têm como base o foco na qualidade do software desenvolvido.



Papel do software

Quanto mais as aplicações de software são utilizadas, mais elas facilitam a vida das pessoas. Terminais de autoatendimento (caixas eletrônicos), e-mail e software de mensagens instantâneas (p. ex. Microsoft Messenger – MSN e Yahoo Messenger), aparelhos eletroeletrônicos (televisores e fornos de microondas) podem receber as mais diversas programações realizadas pelos usuários através de um controle remoto ou em um painel digital, carros que realizam várias verificações no sistema elétrico e de injeção de combustível e emitem sinais alertando sobre o não uso do cinto de segurança, aviões capazes de realizar muitas de suas operações, praticamente, sem a intervenção humana, aplicações de software que interpretam e leem textos auxiliando portadores de deficiência visual são apenas alguns exemplos de como o software está inserido na vida cotidiana.



Papel do software

Quanto mais a tecnologia evolui, quanto mais a humanidade se globaliza, quanto mais atividades as pessoas procuram realizar, maior é o impacto do software na sociedade e na cultura. O software exerce papel importante na vida do ser humano ajudando, auxiliando, conduzindo e executando as mais diversas tarefas com maior eficiência (cada vez mais rápido) e eficácia (cada vez melhor).



Papel do software

Quanto mais a tecnologia evolui, quanto mais a humanidade se globaliza, quanto mais atividades as pessoas procuram realizar, maior é o impacto do software na sociedade e na cultura. O software exerce papel importante na vida do ser humano ajudando, auxiliando, conduzindo e executando as mais diversas tarefas com maior eficiência (cada vez mais rápido) e eficácia (cada vez melhor).

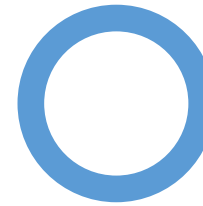


Características do software

Antes de entender quais as principais características de um software é preciso compreender o aspecto do processo de criação do ser humano.

Quando um produto qualquer é construído várias etapas se seguem:

- Análise
- Projeto
- Desenvolvimento
- teste.



Características do software

Uma vez que estas etapas tenham sido seguidas um produto passa a existir na forma física.

Como exemplo, tem-se a construção de uma casa, uma vez que o engenheiro civil tenha conhecimento do local no qual a casa será erguida e os desejos do futuro morador, ele realiza uma análise para saber quais os materiais mais adequados para a construção, então constrói um esboço da casa na forma de projeto.



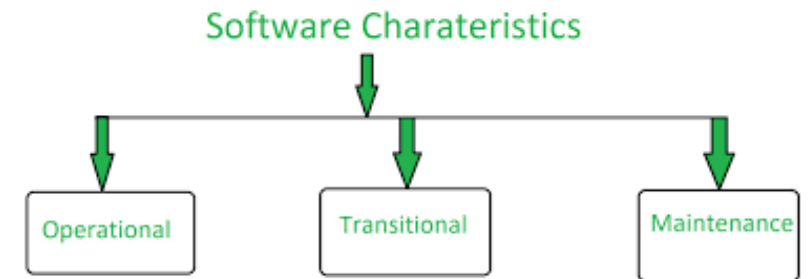


Características do software

Uma vez que o projeto esteja pronto, o mestre de obras é encarregado de orientar os pedreiros na construção da casa, ao mesmo tempo em que realiza verificações e testes para confirmar que a construção atende aos desejos do morador e as especificações do projeto. Assim que a construção é terminada, tem-se um produto físico, algo palpável.

Características do software

Quando o assunto é software, a idéia de construção é semelhante a qualquer outro produto, porém com diferentes abordagens. O software faz parte de um sistema lógico e não de um sistema físico, conforme atesta Pressman (2006).



Características do software

Com isto, as principais características de um software são:

1. O software é desenvolvido e não manufaturado;
2. Software não se desgasta;
3. Apesar da possibilidade de se desenvolver um software a partir de componentes pré-existentes, a maior parte das aplicações de software ainda é desenvolvida sob demanda, de acordo com as necessidades do usuário.



Processo de software, intraestrutura do processo

O processo de software é um esboço, uma estrutura inicial para as tarefas necessárias à construção de um software de alta qualidade. Sendo assim, cabe a seguinte indagação:

Processo de software é sinônimo de engenharia de software?

Para esta questão há **duas respostas**, dependendo de como a ela é percebida.



Processo de software, infraestrutura do processo

1. Tanto o processo de software quanto a engenharia de software descrevem como a construção de um software deve ser realizada.
2. Porém, a engenharia de software vai além, ela acrescenta tecnologias (métodos técnicos e ferramentas automatizadas) que formam o processo.



Processo de software, infraestrutura do processo

Os métodos abrangem um conjunto de tarefas necessárias à construção de um software. Estas tarefas são análise de requisitos, projeto, construção de programas, testes e validações e manutenção.

As ferramentas, que podem ser automatizadas ou semi-automatizadas, apoiam o processo e os métodos. Se a informação criada por uma ferramenta puder ser usada por outra se diz que a engenharia de software é auxiliada pelo computador ou, do inglês, CASE (*Computer Aided Software Engineering*).



Processo de software, infraestrutura do processo

Uma ferramenta CASE pode ser apenas um software que inclua editores de projeto e dicionários de dados, por exemplo, ou uma combinação de software, hardware e um repositório de informações a respeito de análise, projeto, construção e testes. Ferramentas CASE são apresentadas mais adiante.

