

Conceitos da engenharia de software

Apresentação

A engenharia de software tem como objetivo, a aplicação de metodologias no processo de desenvolvimento, visando a promoção da qualidade, aumento da produtividade e redução dos custos. A criação de software foi subestimada e realizada sem nenhuma metodologia, gerando erros em sistemas, como: problemas de cálculos, perdas financeiras e de tempo. Nesse período, podemos dizer que houve a Crise do Software. Com isso, em 1967 a OTAN (Organização do Tratado do Atlântico Norte) designou o termo Engenharia de Software para adequar o processo de desenvolvimento de software com metodologias, já utilizadas em outras engenharias. Uma série de metodologias e técnicas passaram a ser utilizadas antes, durante e depois da criação dos softwares. Dados históricos apontam que houve uma diminuição brutal nos problemas no desenvolvimento de softwares após a adoção dessas metodologias, fazendo com que a indústria de software pudesse entregar sistemas com maior qualidade, em menos tempo e com custos reduzidos de manutenção.

Nesta Unidade de Aprendizagem, você irá adquirir conhecimentos fundamentais para avançar no aprendizado sobre Engenharia de Software. Iremos abordar, inicialmente, conceitos básicos sobre o que é Engenharia de Software, sua história e importância na indústria.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Reconhecer o histórico e conceitos fundamentais da Engenharia de Software.
- Analisar a evolução do desenvolvimento de software.
- Identificar a importância da Engenharia de Software.

Desafio

Paulo é gestor de uma empresa de tecnologia e costuma viajar com frequência para atender clientes. Mediante uma curta fase de ociosidade de sua equipe, o empresário resolveu aproveitar para solicitar o desenvolvimento de um *software* que integrasse a sua agenda e a compra automática de suas passagens aéreas. No segundo mês de uso do *software*, ao chegar no aeroporto e tentar fazer o *check-in*, Paulo percebeu que a passagem havia sido comprada para Fortaleza ao invés de Salvador.

Analise esse cenário e associe o erro do *software* com o conceito de Engenharia de Software.

O erro na compra de passagem é um indício de que a empresa não utilizou os princípios de Engenharia de Software no planejamento do software, pois esse conceito está ligado à confiabilidade e assertividade na produção.

Com a utilização da Engenharia de Software, podemos garantir que o cliente terá um software:

- De qualidade.
- Que atenda às necessidades.
- Resolva os problemas encontrados atualmente.

Também temos maior controle do processo de desenvolvimento, custos e estimativas de tempo. Com os testes podemos garantir o funcionamento do sistema e também a qualidade final, tendo a certeza que o software entregue atende às necessidades levantadas no início do projeto.

Infográfico

A Engenharia de *Software* utiliza os princípios da engenharia para obter *softwares* de maneira econômica e confiável, o que pode garantir inúmeras vantagens para o processo de criação. Acompanhe, no infográfico a seguir, as características e vantagens da engenharia de *software*.

CARACTERÍSTICAS DA ENGENHARIA DE *SOFTWARE*



criação
+
utilização



sólidos princípios
de engenharia



software



econômica



confiável



máquinas
reais

MEIOS

criação
+
utilização



Criar, com base na necessidade real do cliente, fazendo testes de utilização, para possibilitar que o *software* seja entregue com o menor número de erros possíveis.

sólidos
princípios de
engenharia



Utilizar técnicas, modelos e metodologias já conhecidos por outras áreas da engenharia, de forma organizada, em etapas e com prazos, objetivos, custos e outras variáveis bem definidas.

RESULTADO

software



Gerar um *software* de qualidade que atenda à solicitação inicial de forma eficiente.

VANTAGENS

econômica



Tornar o processo de desenvolvimento de *software* mais econômico e sem grandes variações de custos.

confiável



Oferecer um sistema mais confiável e de alta qualidade ao final do projeto.

máquinas
reais



Trabalhar em máquinas reais, ou seja, um sistema computacional acessível ao cliente.

Conteúdo do Livro

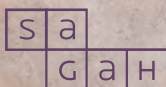
A engenharia de *software* foi criada para tentar solucionar os problemas da "Crise de *Software*". Ela abrange uma série de metodologias que guiam todo o processo de criação de *software*, garantindo a alta qualidade, respeito aos prazos e custos do projeto.

Acompanhe a leitura do capítulo Conceitos da Engenharia de *Software*, da obra *Engenharia de Software* e saiba mais sobre os conceitos básicos de engenharia, bem como sua história e importância na indústria.

Boa leitura!

ENGENHARIA DE *SOFTWARE*

Aline Zanin



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Conceitos da Engenharia de Software

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Reconhecer o histórico e os conceitos fundamentais da Engenharia de Software.
- Analisar a evolução do desenvolvimento de software.
- Identificar a importância da Engenharia de Software.

Introdução

Por muitos anos a Engenharia de Software foi utilizada com o objetivo de criar software de qualidade dentro dos custos e dos prazos estimados pelo cliente, evitando desperdícios de tempo, esforços, direções erradas e atrasos. A criação de software foi subestimada e realizada sem nenhuma metodologia, gerando erros em sistemas, como problemas de cálculos e perdas financeiras e de tempo. Nesse período, podemos dizer que houve a crise do software. Com isso, em 1967, a Organização do Tratado do Atlântico Norte (OTAN) designou o termo Engenharia de Software para adequar o processo de desenvolvimento de software com metodologias já utilizadas em outras Engenharias. Uma série de metodologias e técnicas passaram a ser utilizadas antes, durante e depois da criação de software. Dados históricos apontam que houve uma diminuição brutal nos problemas em software após a adoção dessas metodologias, fazendo com que a indústria de software pudesse entregar sistemas com maior qualidade, em menos tempo e com custos reduzidos de manutenção.

Neste capítulo, você irá adquirir conhecimentos fundamentais para avançar no aprendizado sobre Engenharia de Software. Iremos abordar inicialmente conceitos básicos sobre o que é essa Engenharia, sua história e a importância na indústria.

Histórico e conceitos fundamentais da Engenharia de Software

A Engenharia de Software é uma disciplina da Engenharia, mais especificamente da Ciência da Computação, que estuda todos os processos envolvidos no desenvolvimento de software, uma atividade complexa que envolve a realização conjunta de diversas atividades distintas, as quais exigem habilidades multidisciplinares e, por consequência, trabalho colaborativo de um grande grupo de profissionais (SOMMERVILLE, 2011).

A Engenharia de Software é uma área de grande importância, uma vez que as pessoas e a sociedade como um todo estão a cada dia mais dependentes de software. Por isso, faz-se necessário que seja produzido software mais confiável e de forma mais rápida a cada dia. Além disso, para as empresas desenvolvedoras de sistemas, geralmente é mais barato, a longo prazo, usar métodos e técnicas da Engenharia de Software para o desenvolvimento de sistemas do que desenvolver os sistemas sem documentação e estruturação, uma vez que, desta forma, é desestruturada e dificultada a manutenção do software (SOMMERVILLE, 2011).

Contudo, esta disciplina nem sempre foi alvo de atenção dos profissionais de Tecnologia da Informação. Por muito tempo, o desenvolvimento de sistemas foi realizado sem atenção a processos, metodologias e estruturas organizacionais no que diz respeito a tarefas, atividades e responsabilidades. Essa falta de controle sobre os processos fez com que, muitas vezes, o software fosse entregue aos clientes sem a devida qualidade e com grande número de erros, como problemas de cálculos e perdas financeiras e de tempo. Nesse período, podemos dizer que houve a crise do software. Com isso, em 1967, a OTAN designou o termo Engenharia de Software para adequar o processo de desenvolvimento de software com metodologias já utilizadas em outras engenharias.

A partir desse momento, os profissionais e as empresas de Tecnologia da Informação passaram a preocupar-se mais com os diversos setores que envolvem o desenvolvimento de sistemas, como análise de requisitos, análise de sistemas, desenvolvimento, testes e implantação. Neste contexto, derivaram diversas metodologias, métodos e processos para auxiliar e guiar o trabalho de cada um desses segmentos.

Evolução do desenvolvimento de software

O desenvolvimento de software, bem como outras Ciências, empregou diversas mudanças e adaptações para melhorar, facilitar e adaptar-se ao cotidiano dos profissionais que realizam esse trabalho. As principais evoluções no desenvolvimento de software podem ser classificadas em dois grandes grupos: mudanças processuais e mudanças tecnológicas.

Mudanças tecnológicas

Embora atualmente, quando se fala em software, sejamos remetidos a lembrar de computadores modernos, *smartphones*, *tablets*, etc., o desenvolvimento de software começou muito antes desses dispositivos serem criados, sendo programado por volta de 1725, em cartões perfurados. Posteriormente, surgiram as primeiras linguagens de programação, tais quais as que existem hoje, sendo elas FORTRAN (1955), *List Processor* (LISP) e *Common Business Oriented Language* (COBOL). Posteriormente, surgiram linguagens de programação de alto nível, isto é, que se aproximam mais da linguagem humana, são exemplos: *Java*, *JavaScript*, *Visual Basic*, *Object Pascal* e PHP (PACIEVITCH, 2017).

Junto com as linguagens de programação, foram sendo criados paradigmas para o desenvolvimento de sistemas. Um paradigma nada mais é do que a forma como um sistema é construído e seu desenvolvimento é organizado. Os paradigmas mais conhecidos são o paradigma estruturado e o paradigma orientado a objetos, sendo o paradigma orientado a objetos o mais utilizado atualmente. A programação orientada a objetos é um paradigma em que o software é construído considerando que tudo o que é inserido no programa é um objeto e que esse objeto pertence a uma classe e tem características (atributos) específicas sobre as quais podem ser feitas ações (métodos). Por outro lado, um princípio básico da programação estruturada é que um programa pode ser dividido em três partes que se interligam, sendo elas sequência, seleção e iteração (ABÍLIO, 2017). Na sequência, o código do programa é criado para ser executado de forma sequencial, seguindo estritamente a ordem na qual foi programado. Na seleção, o programa encontra locais onde pode seguir um ou mais caminhos distintos. Na interação, é permitido ao programa executar diversas vezes o mesmo trecho de código.

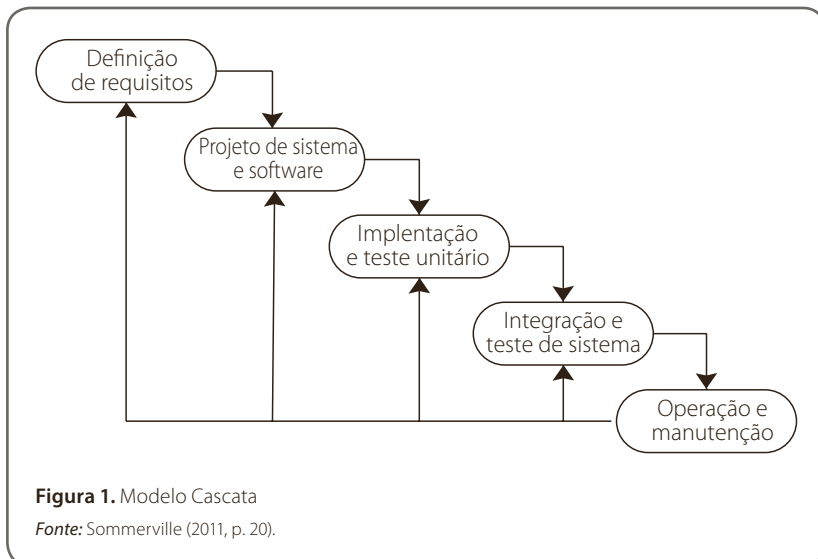


Fique atento

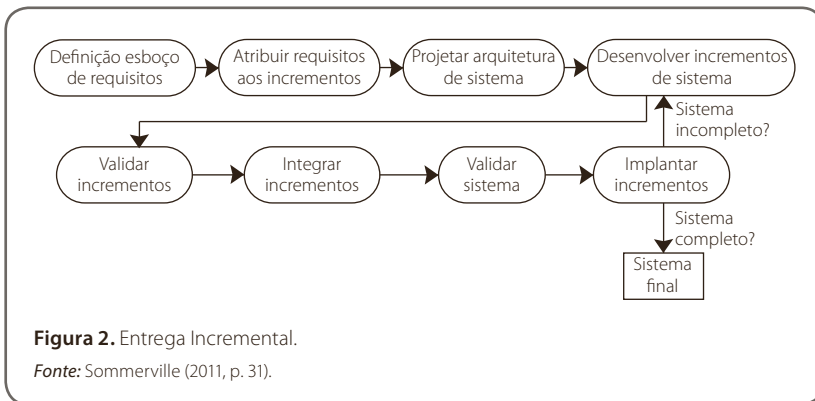
- Ao programar uma calculadora, quando cria-se um programa e o único fluxo que este pode seguir é receber dois números, somar esses números e mostrar o resultado, faz-se um programa utilizando apenas sequência.
- Quando se insere neste programa a opção de selecionar se deve somar ou subtrair os números, se está programando uma seleção.
- Quando, ao final do cálculo, pergunta-se para o usuário se deseja fazer novamente, se está programando uma interação.

Mudanças de processo

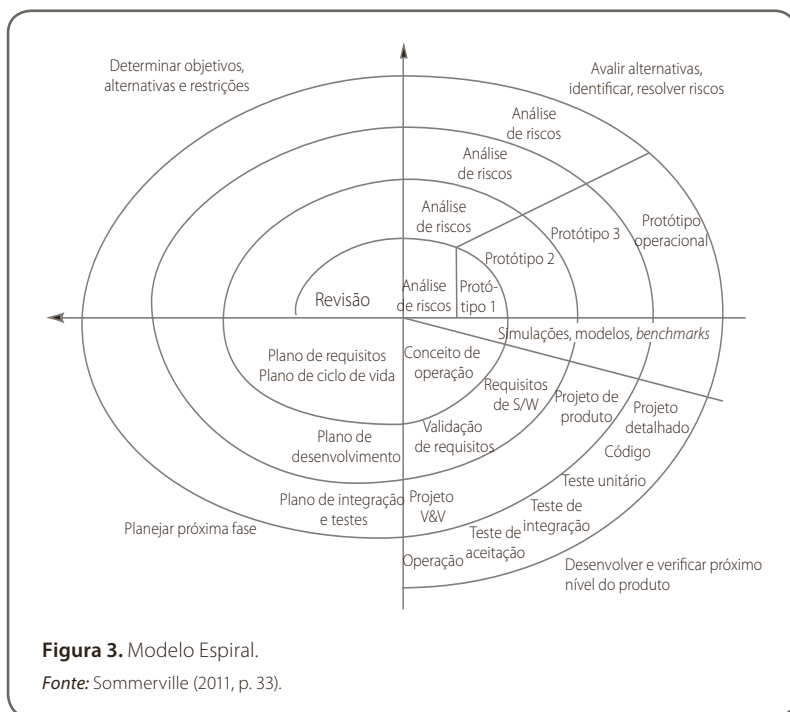
No desenvolvimento de sistemas, além das mudanças tecnológicas, foram ocorrendo mudanças na forma como as empresas se organizam e estruturam o trabalho. A forma tradicional de desenvolvimento de sistemas foi a primeira a ser criada, empregando o ciclo de vida em estrutura de **cascata** (1970), na qual as etapas são executadas de forma sequencial, sem que seja possível retornar de uma etapa posterior para uma etapa anterior.



Posteriormente, falou-se em desenvolvimento **iterativo e incremental**. Nesse modelo, implementa-se pequenas partes entregáveis do software para que o cliente tenha um *feedback* mais rápido sobre o produto que está sendo desenvolvido.



O modelo em **espiral** se assemelha muito ao modelo iterativo e incremental, uma vez que ele também considera pequenas entregas de software e a execução de todas as etapas espiralmente (várias vezes). Contudo, o ciclo de vida espiral considera a presença explícita da análise de riscos como uma das etapas de cada iteração. Nesse processo em espiral, o ciclo de vida do software é representado como uma espiral em que a volta na espiral representa uma fase do processo de software, sendo que a volta mais interna pode preocupar-se com a viabilidade do sistema, o ciclo seguinte, com definição de requisitos, o seguinte, com o projeto do sistema, e assim por diante.



No ano de 2001, um grupo de profissionais de tecnologia da informação lançou um documento chamado Manifesto Ágil para o Desenvolvimento de Sistemas. Deste então popularizaram-se os métodos ágeis de desenvolvimento de sistemas, sendo os mais conhecidos o método Scrum e o método XP. Todos eles têm em comum a aplicação dos valores propostos pelo Manifesto Ágil para o Desenvolvimento de Sistemas, sendo eles: indivíduos e interações são mais que processos e ferramentas, software em funcionamento é mais que documentação abrangente, colaboração com o cliente é mais que negociação de contratos e respostas a mudanças são mais que seguir um plano (BECK et al., 2001).

Todos esses ciclos de vida, somados aos Métodos Ágeis de Desenvolvimento de Sistemas, apresentaram estrutura e organização maiores para o processo de desenvolvimento de sistemas, propiciando melhoria da comunicação entre os envolvidos no processo, seja entre os próprios profissionais de Tecnologia da Informação ou destes com o cliente. Com a adoção de processos e a atenção às evoluções tecnológicas, buscando sempre acompanhar aquilo que o mercado

tem de melhor para oferecer, pode-se atingir maior excelência nos produtos entregues e atender melhor às necessidades do cliente.

Importância da Engenharia de Software

Conforme supracitado, a Engenharia de Software é uma disciplina de Engenharia cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até a sua manutenção, quando o sistema já está sendo usado (SOMMERVILLE, 2011). Neste contexto, é clara a importância fundamental da Engenharia de Software, uma vez que, se o processo de desenvolvimento de sistemas envolve diversas atividades distintas e a Engenharia de Software é a disciplina que preocupa-se em estudar e monitorar o bom andamento de todas essas atividades e a integração entre elas, é trivial notar que é baseado na Engenharia de Software o sucesso de um projeto no que tange a sua organização.

Engenharia de Software envolve **planejamento**. Planejamento diz respeito também a pessoas e cronograma de trabalho. Pessoas porque divide as responsabilidades, de forma individual ou coletiva. Cronograma porque conforme o planejamento é que os gestores têm a possibilidade de mensurar o tempo necessário para o desenvolvimento de cada projeto. Engenharia de Software envolve também a preocupação com a **qualidade** do produto. Qualidade, neste contexto, não se refere apenas à entrega de produtos em funcionamento, mas também ao atendimento das necessidades do cliente, e, por isso, a área da Engenharia de Software, que trata de engenharia de **requisitos** ou análise de requisitos, tem importância fundamental, na medida em que é por meio dela que as equipes de desenvolvimento recebem a expectativa do cliente sobre o produto que está sendo desenvolvido para buscar atendê-la.

Além disso, na fase de **desenvolvimento** da programação em si, a Engenharia de Software se faz presente, uma vez que a escolha do processo ideal irá influenciar diretamente no trabalho cotidiano de todos os envolvidos, incluindo a programação. Esse fator ganha relevância ainda maior em times que trabalham em horários distintos ou locais geograficamente distribuídos.

Uma vez entregue o software para o cliente, a Engenharia de Software tem sua importância revelada no momento de realizar a manutenção nesse software. Isto porque, se o sistema tiver sido corretamente planejado, o código do sistema tende a estar mais limpo e com menos defeitos. Isso irá causar menos manutenção e facilitar as manutenções que precisam ser realizadas.



Referências

ABÍLIO, I. *Programação orientada a objetos versus programação estruturada*. Disponível em: <<http://www.devmedia.com.br/programacao-orientada-a-objetos-versus-programacao-estruturada/32813>>. Acesso em: 17 ago. 2017.

BECK, K. et al. *Manifesto for agile software development*. c2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 17 ago. 2017.

PACIEVITCH, Y. *História da programação*. Disponível em: <<http://www.infoescola.com/informatica/historia-da-programacao>>. Acesso em: 17 ago. 2017.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011.

Leituras recomendadas

ENDEAVOR BRASIL. *PDCA: a prática levando sua gestão à perfeição*. 2015. Disponível em: <<https://endeavor.org.br/pdca/>>. Acesso em: 4 nov. 2016.

PEQUENO, C. N.; CARVALHO, M. G. F.; FONTES, V. M. *Redução do consumo de produto químico utilizado em uma linha de produção de uma indústria de pneus*. 2015. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção)–Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2015.

REY, B. *Como fazer um brainstorming eficiente*. 2013. Disponível em: <<http://exame.abril.com.br/carreira/como-fazer-um-brainstorming-eficiente/>>. Acesso em: 4 nov. 2016.

RODRIGUES, S. *Crise: perigo, oportunidade e... papo furado*. 2013. Disponível em: <<http://veja.abril.com.br/blog/sobre-palavras/lendo-a-lenda/crise-perigo-oportunidade-e-papo-furado/>>. Acesso em: 4 nov. 2016.

SILVA, M. D. L. et al. Gestão da produção: estudo sobre a gestão da manutenção na geração de energia e vapor utilizando caldeiras de uma indústria. In: ENCONTRO PARAENSE DE ENGENHARIA DE PRODUÇÃO, 7., 2016, Belém. *Anais...* Belém: [s.n.], 2016.

SLACK, N. et al. *Gerenciamento de operações e de processos: princípios e práticas de impacto estratégico*. 2. ed. Porto Alegre: Bookman, 2013.

SOUZA, T. de J. F. et al. Proposta de melhoria do processo de uma fábrica de polpas por meio da metodologia de análise e solução de problemas. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 35., 2015, Fortaleza. *Anais...* Fortaleza: ABEPRO, 2015. Disponível em: <http://www.abepro.org.br/biblioteca/TN_STP_207_228_27341.pdf>. Acesso em: 4 nov. 2016.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Dica do Professor

A Engenharia *Software* surgiu em 1967 para atender uma necessidade de desenvolvimento de *softwares* de qualidade, a partir de técnicas de engenharia. Veja, na dica do professor a seguir, o que foi a crise de *software*, e acompanhe alguns conceitos e a importância da engenharia de *software* no desenvolvimento de sistemas.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Exercícios

1) O que foi a Crise de *Software*?

- A) A Crise de *Software* permitiu o desenvolvimento de *software* de alta qualidade já que houve um aumento da concorrência.
- B)** A Crise de *Software* foi um termo que surgiu nos anos 70. O termo expressava as dificuldades do desenvolvimento de *software* frente ao rápido crescimento da demanda por *software*.
- C) A Crise de *Software* foi acompanhada pela Crise de *Hardware*, que acabou gerando inúmeros desempregos na década de 70.
- D) A Crise de *Software* foi um termo criado para expressar momentos em que um sistema apresenta processamento lento.
- E) A Crise de *Software* ocorreu após a Segunda Guerra Mundial quando nenhum *software* era vendido.

2) Qual foi o motivo da criação da Engenharia de *Software*?

- A) A Engenharia de *Software* foi criada porque nenhum *software* disponível antes da Engenharia de *Software* conseguia realizar cálculos complexos.
- B)** A Engenharia de *Software* foi criada para permitir o uso de elementos da engenharia de forma controlada e sistemática no desenvolvimento de *software*. Também para evitar a Crise de *Software*.
- C) A Engenharia de *Software* foi criada para acelerar o desenvolvimento de *software* no Brasil.
- D) A Engenharia de *Software* foi criada para facilitar o uso de *software*.
- E) A Engenharia de *Software* foi criada para permitir que a produção de novos sistemas tivesse mais elementos gráficos e amigáveis ao usuário.

3) Com a introdução da Engenharia de *Software*, o que mudou no processo de desenvolvimento de *software*?

- A)** Iniciou-se o uso de técnicas e metodologias sistemáticas e controladas já presentes na engenharia e amplamente utilizadas em outras áreas.
- B) A Engenharia de *Software* melhorou o entendimento do desenvolvedor na leitura dos requisitos de *Software*.
- C) Aumentaram as vendas de sistemas de *software* na década de 80.
- D) Permitiu que mais pessoas pudessem ter acesso a sistemas de *software*.
- E) Removeu da criação de *software* as técnicas e metodologias sistemáticas e controladas já presentes na engenharia e amplamente utilizadas em outras áreas.
- 4) João, dono de uma empresa de *software*, tem que criar um sistema para um cliente. Até o momento, o cliente fez apenas uma ligação informando o tipo de *software* que ele quer. Qual a primeira coisa que João deve fazer?**
- A) Ir para a sua empresa e começar a programar imediatamente.
- B) Modelar algumas telas do sistema e perguntar ao cliente a sua opinião.
- C) Contratar uma grande equipe de desenvolvedores para criar o *software* o mais rápido possível.
- D)** Entender o negócio do cliente e realizar reuniões para mensurar o que ele precisa.
- E) Informar para o cliente que em um mês o sistema estará em pleno funcionamento, além de informar qual será o custo do sistema.
- 5) Qual é a base dos elementos da Engenharia de *Software*?**
- A) Métodos.
- B) Ferramentas.
- C)** Foco na qualidade.
- D) Processo.
- E) Conceitual.

Na prática

Você consegue identificar a necessidade do uso de metodologias da Engenharia de *Software*? Para demonstrar essa importância, iremos analisar e comparar duas situações no desenvolvimento de sistemas, uma utilizando o método chamado "*Go Horse*" e a outra a Engenharia de *Software*.

Conteúdo interativo disponível na plataforma de ensino!

Podemos ver que Pedro não utilizou nenhum método para garantir a qualidade do sistema, não planejou o desenvolvimento, não testou o produto final e entregou um *software* de má qualidade. Além disso, a correção dos problemas levou quatro vezes mais tempo que o planejado inicialmente e custou mais, pois precisou alocar um desenvolvedor durante todo o período. Esses problemas poderiam ter sido evitados se Pedro tivesse utilizado métodos amplamente abordados na Engenharia de *Software*.

Conteúdo interativo disponível na plataforma de ensino!

João seguiu etapas bastante utilizadas na Engenharia de *Software*, essas etapas puderam garantir a entrega de um *software* de qualidade, desempenhando as funções de acordo com o que o cliente precisava. O sistema foi entregue dentro do prazo e custo estimados, obtendo lucro no final.

Comparando as duas situações...

No exemplo da situação 1, Pedro utilizou o método "*Go Horse*" onde se passa uma tarefa pouco planejada para o desenvolvedor iniciar imediatamente, construindo algo que não vai ao encontro das necessidades do cliente.

Na situação 2, João utilizou etapas da Engenharia de *Software* para estruturar o desenvolvimento, tentando garantir um produto de qualidade, com custo e tempo adequados. Na Engenharia de *Software* encontramos diversos modelos, técnicas e análises, além das fases demonstradas neste exemplo.

Saiba mais

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Engenharia de Software - Parte A

No vídeo a seguir, você poderá saber mais sobre a importância da Engenharia de Software, bem como conhecer alguns modelos de processos e técnicas, o desenvolvimento ágil e a gestão de projetos.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Uma breve História da Engenharia de Software

O termo Engenharia de software tornou-se conhecido após uma conferência em 1968, com a discussão das dificuldades da projeção de sistemas complexos. Veja, a seguir, uma breve história da Engenharia de Software.



Aponte a câmera para o código e acesse o link do conteúdo ou clique no código para acessar.

Engenharia de Software - Os Paradigmas Clássico & Orientado a Objetos

Confira uma introdução aos fundamentos teóricos da engenharia de software e os aspectos mais práticos do ciclo de vida do software, na obra Engenharia de Software: Os Paradigmas Clássico & Orientado a Objetos.

Conteúdo interativo disponível na plataforma de ensino!

Engenharia de Software - Uma Abordagem Profissional

No livro Engenharia de Software: Uma abordagem Profissional, você poderá saber mais sobre a segurança de software e os desafios específicos ao desenvolvimento para aplicativos móveis.

Conteúdo interativo disponível na plataforma de ensino!