

Engenharia de Software



A Fernanda

Para quem não me conhece, eu sou diretora de produtos e treinamentos da Bluestar Ensino e Tecnologia, e também de relacionamento acadêmico do DFJUG.

Tenho MUITOS anos de experiência profissional em TI, graduada pela UnB e pós graduada pela UCB.



A BLUESTAR

- A Bluestar representa oficialmente diversas empresas de tecnologia nacionais e internacionais, como a Embarcadero, a CA, Corel e a AVG, entre outros grandes nomes do mercado do TI; tem como foco a área de ensino e desenvolvimento de tecnologia.
- Nossos treinamentos são reconhecidos pelo excelente nível técnico e didático dos instrutores, certificados nas provas mais importantes do mercado.
- Com excelente infraestrutura e material didático, o curso profissionalizante da Bluestar é o primeiro de Brasília, atingindo os melhores resultados na colocação profissional dos alunos formados.



Nossa empresa ainda atua em venda de softwares e consultorias especializadas.

Fazemos serviços de seleção especializada, suporte, hospedagem, design gráfico, EAD e desenvolvimento.

Nosso Centro de Treinamento fica localizado na SCRN 716 - Bloco "B" - Loja 36 (Brasília - DF)

Acesse vagas de emprego e notícias de tecnologia em nosso Facebook: Bluestar - Ensino e Tecnologia.



Os Senhores!!

Façam uma apresentação pessoal:

Nome

Cargo e Empresa

Sua experiência com engenharia de software Suas expectativas para o curso

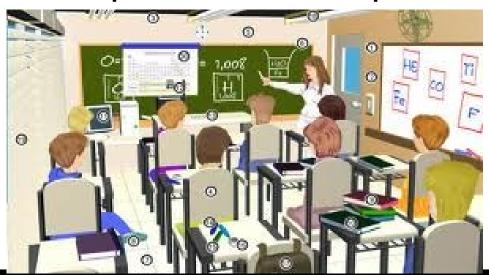


Contrato

Fique à vontade para perguntar, discordar e iniciar debates;

Aparelhos eletrônicos em modo silencioso;

Os slides não precisam ser copiados







Não leve dúvidas para casa. O instrutor fica deprimido quando os alunos não têm dúvidas.

Façam os exercícios – é a melhor maneira de assimilar o conteúdo.





Os telefones celulares são ótimos companheiros de classe – desde que desligados. Se for realmente importante atender, faça-o fora da sala. O instrutor e seu colega ao lado agradecem.



Datas

16 horas
26/05 aula 14 as 18 horas
02/06 aula 14 as 18 horas
09/06 aula 14 as 18 horas
16/06 (estarei fora de Brasília)
23/06 aula 14 as 18 horas
30/06 prova 14 as 18 horas



Conteúdos e Objetivos

- ☐ Este curso é voltado para profissionais da área de TI que desejam adquirir conhecimento nos em Engenharia de software.
- □ Vamos ter uma visão geral de todas as áreas e funcionalidades para melhor compreensão do mundo da TI.
- □ Vamos fazer simulados de concursos com est es assunt os e est udar bastant e j unt os :)
- ☐ Aproveitem e divirtam-se



Temas das Aulas

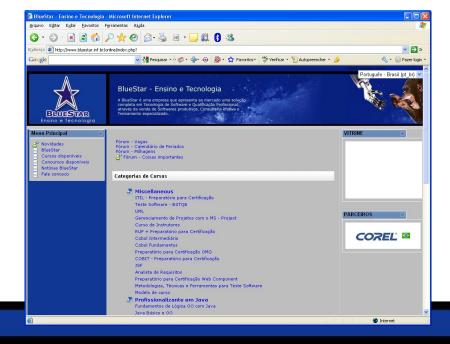
- 1 Definição
- 2 Áreas de conhecimento
- 3 Processo de software
- 3.1 Modelos de processo de software
- 3.2 Modelos de maturidade
- 4 Metodologias e métodos
- 4.1 Modelagem
- 5 Ferramentas, tecnologias e práticas
- 5.1 Ferramentas

- 6 Gerência de projetos
- 6.1 Planejamento
- 6.2 Análise de requisitos
- 6.3 Gestão
- 7 Histórico
- 8 ES no presente e tendências
- 9 Referências
- 10 Bibliografia
- 11 Ligações externas



Material de Apoio

- □ Ambiente EaD
 - http://www.bluestar.inf.br
 - Login: seu CPF
 - Senha: seu primeiro nome





Aula 01

Para que serve engenharia de software?



Aula 01

Para que serve engenharia de software?



Engenharia de software

Engenharia de software é uma área da computação voltada à especificação, desenvolvimento e manutenção de sistemas de software, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade.



- linguagens de programação
 - banco de dados
 - ferramentas
 - plataformas
 - bibliotecas
 - padrões
 - processos
 - Qualidade de Software.



Uso de modelos abstratos e precisos Analista:

- especificar
 - projetar
- implementar
- manter sistemas de software
- * avaliando e garantindo suas qualidades.
- oferecer mecanismos para se planejar e gerenciar o processo de desenvolvimento de um sistema.



Definição

Friedrich Ludwig Bauer foi o primeiro dizendo: "Engenharia de Software é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe em máquinas reais". O próprio significado de engenharia já traz os conceitos de criação, construção, análise, desenvolvimento e manutenção.



O termo foi criado na década de 1960 e utilizado oficialmente em 1968 na NATO Science Committee. Sua criação surgiu numa tentativa de contornar a crise do software e dar um tratamento de engenharia (mais sistemático e controlado) ao desenvolvimento de sistemas de software complexos.



E o que são sistemas complexos?



Um sistema de software complexo se caracteriza por um conjunto de componentes abstratos de software (estruturas de dados e algoritmos) encapsulados na forma de procedimentos, funções, módulos, objetos ou agentes e interconectados entre si, compondo a arquitetura do software, que deverão ser executados em sistemas computacionais.



Empresas desenvolvedoras de software passaram a empregar esses conceitos sobretudo para orientar suas áreas de desenvolvimento, muitas delas organizadas sob a forma de Fábrica de Software.



O que é uma fábrica de software?



O que a gente gostaria que fosse





O que é





E qual o motivo para existir Engenharia de Software?

- atividade de crescente
- utilização de computadores nas mais diversas áreas do conhecimento humano
 - crescente demanda por soluções computadorizadas
- aumento da complexidade da informação.













Outro motivo





Escopo – custo - prazo





Ou seja...

- Qualidade
- Produtividade
- Processos
- Métodos
- Técnicas
- Ferramentas
- ambientes de suporte ao desenvolvimento de software.



Dividir para conquistar





Assim como em outras áreas, em uma abordagem de engenharia de software, inicialmente o problema a ser tratado deve ser analisado e decomposto em partes menores, em uma abordagem "dividir para conquistar". Para cada uma dessas partes, uma solução deve ser elaborada.



Solucionados os sub-problemas isoladamente, é necessário integrar as soluções. Para tal, uma arquitetura deve ser estabelecida. Para apoiar a resolução de problemas, procedimentos (métodos, técnicas, roteiros etc) devem ser utilizados, bem como ferramentas para parcialmente automatizar o trabalho



Neste cenário, muitas vezes não é possível conduzir o desenvolvimento de software de maneira individual. Pessoas têm de trabalhar em equipes, o esforço tem de ser planejado, coordenado e acompanhado, bem como a qualidade do que se está produzindo tem de ser sistematicamente avaliada.



Aula 2

Áreas de conhecimento



SWEBOK (Corpo de Conhecimento da Engenharia de Software), versão 2004, as áreas de conhecimento da Engenharia de Software

O Guide to the Software Engineering Body of Knowledge, conhecido pela sigla SWEBOK, é um documento criado sob o patrocínio da IEEE com a finalidade de servir de referência em assuntos considerados, de forma generalizada pela comunidade, como pertinentes a área de Engenharia de Software.



Requisitos (Requirements) de Software Projeto (Design) de Software Construção (Construction) de Software Teste (Testing) de Software Manutenção (Maintenance) de software Gerência de Configuração de Software Gerência de Engenharia de Software Processos de Engenharia de Software Ferramentas e Métodos de Engenharia de Software Qualidade (Quality) de Software



Conforme Pressman, a
Engenharia de Software (ES) é
uma tecnologia em camadas. E a
base de todas essas camadas é
o foco na qualidade do software
desenvolvido.





Camadas:

- processos
 - métodos
- Ferramentas



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
 - restrições



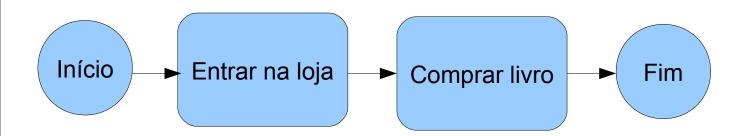


Objetivo: produzir e manter os produtos de software finais requeridos.





Os processos são divididos em atividades ou tarefas. Uma atividade é um passo de processo que produz mudanças de estado visíveis externamente no produto de software.





Atividades incorporam e implementam procedimentos, regras e políticas, e têm como objetivo gerar ou modificar um dado conjunto de artefatos.

Escolher

Comprar livro

Levar ao Caixa

Pagar



Outro conceito importante: ARTEFATO





Um artefato é um produto criado ou modificado durante um processo. Tal produto é resultado de uma atividade e pode ser utilizado posteriormente como matéria prima para a mesma ou para outra atividade a fim de gerar novos produtos.



Uma atividade aloca recursos (por exemplo, computadores, impressoras e material de expediente), é escalonada, monitorada e atribuída a desenvolvedores (agentes), que podem utilizar ferramentas para executá-la.





Toda atividade possui uma descrição, a qual pode especificar os artefatos necessários, as relações de dependência com outras atividades, as datas de início e fim planejadas, os recursos a serem alocados e os agentes responsáveis pela mesma.



O agente pode ser uma pessoa ou uma ferramenta automatizada (quando a atividade é automática) e relaciona-se com as atividades de um processo. Os agentes podem estar organizados em cargos, aos quais podem ser definidas diferentes responsabilidades.



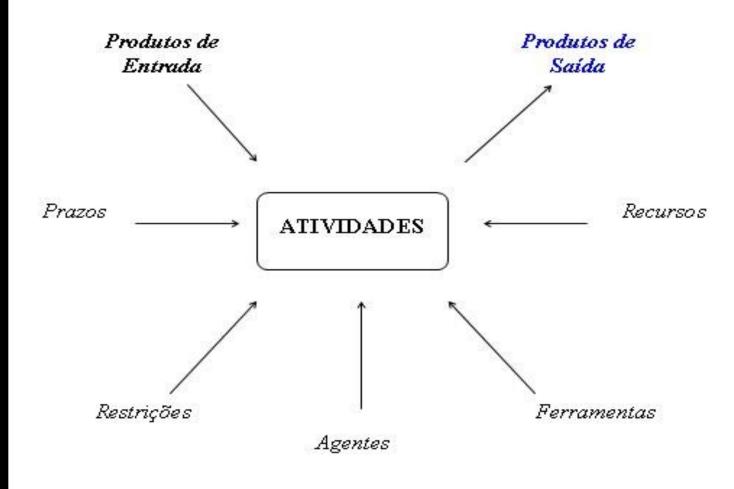




A realização do processo é afetada pelas restrições, que podem atingir atividades, agentes, recursos, artefatos, papéis e seus relacionamentos. Uma restrição é uma condição definida que um passo de processo deve satisfazer antes ou depois de ser executado.



Atividade de Processos de Engenharia de Software e seus inter-relacionamentos





Ou seja, Processo de software, ou processo de engenharia de software, é uma sequência coerente de práticas que objetiva o desenvolvimento ou evolução de sistemas de software.



Estas práticas englobam as atividades de:

- especificação
- projeto (design)
- implementação
 - testes

e caracterizam-se pela interação de ferramentas, pessoas e métodos.



Um modelo de processo de desenvolvimento de software, ou simplesmente modelo de processo, pode ser visto como uma representação, ou abstração dos objetos e atividades envolvidas no processo de software. Além disso, oferece uma forma mais abrangente e fácil de representar o gerenciamento de processo de software e consequentemente o progresso do projeto.



Modelos ciclo de vida

- 1.Sequencial ou Cascata (do inglês waterfall) com fases distintas de especificação, projeto e desenvolvimento.
- 2.Desenvolvimento iterativo e incremental desenvolvimento é iniciado com um subconjunto
 simples de Requisitos de Software e
 iterativamente alcança evoluções subsequentes
 das versões até o sistema todo estar
 implementado
 - 3. Evolucional ou Prototipação especificação, projeto e desenvolvimento de protótipos.



- 4.V-Model Parecido com o modelo cascata, mas com uma organização melhor, que permite que se compare com outros modelos mais modernos.
 - 5.Espiral evolução através de vários ciclos completos de especificação, projeto e desenvolvimento.
- 6.Componentizado reuso através de montagem de componentes já existentes.
 - 7.Formal implementação a partir de modelo matemático formal.

8.Ágil

9.RAD

10.Quarta geração



O Desenvolvimento de um Sistema de Informação, independentemente do modelo de ciclo de vida utilizado, abrange basicamente 4 estágios:

 Problema inicial, que pode ser um erro em um sistema/software existente ou a necessidade da criação de um software para automatizar um processo;





2) Definição e Análise do problema que deverá ser resolvido;





3) Desenvolvimento Técnico ou Codificação que resolverá o problema através da aplicação de alguma tecnologia.





4) Implantação da solução, ou seja, o sistema é entregue ao usuário final.



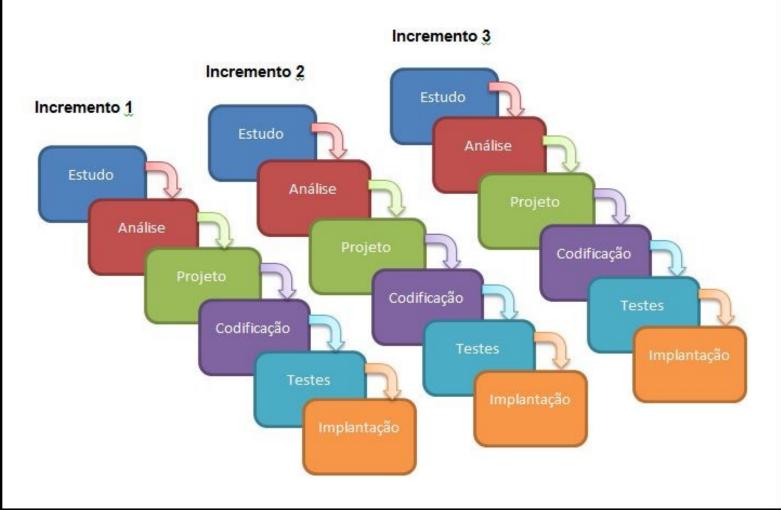


Modelo em Cascata



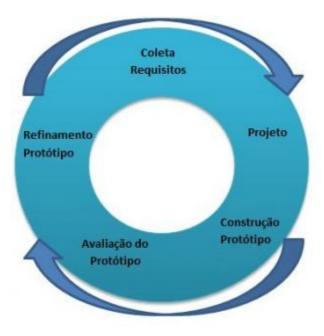


Modelo Incremental Iterativo



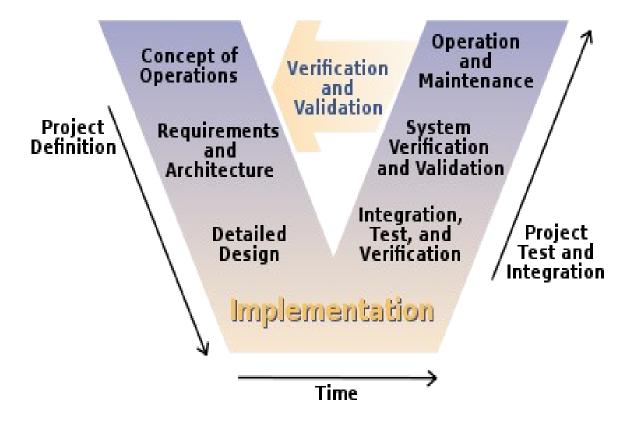


Prototipação



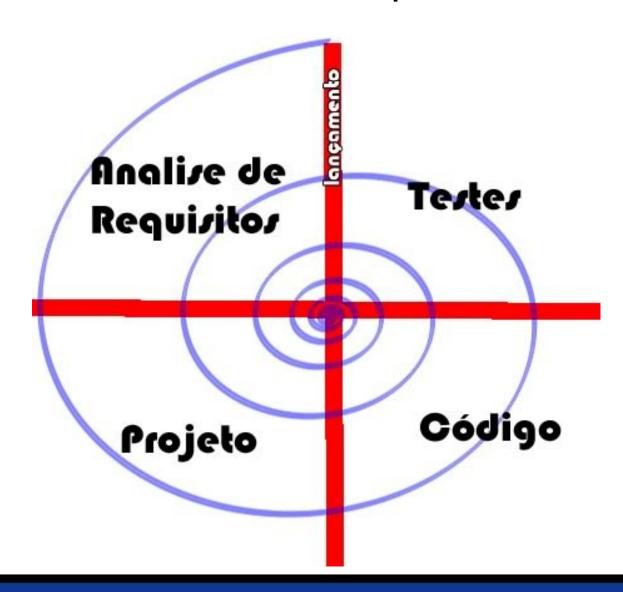


Modelo em V



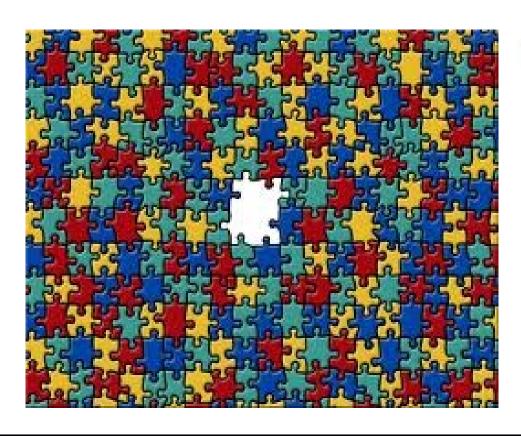


Modelo em Espiral





Modelo Componentizado







Modelo Formal





Modelo	Vantagens	Desvantagens
Cascata	Minimiza o tempo de planejamento. Funciona bem para equipes tecnicamente mais fracas.	Inflexível. Apenas a fase final produz uma entrega que não é um documento. Torna-se difícil voltar atrás para corrigir erros.
Espiral	As interações iniciais do projeto são as mais baratas, permitindo que as tarefas de maior risco sejam levadas com o mínimo de custos. Cada iteração da espiral pode ser customizada para as necessidades específicas de cada projeto.	É complexo e requer atenção e conhecimento especiais para o levar até o final.
Prototipagem Evolucionária	Os clientes conseguem ver os progressos. É útil quando os requisitos mudam rapidamente e o cliente está relutante em aceitar um conjunto de requisitos.	É impossível determinar com exatidão o tempo que o projecto vai demorar. Não há forma de sabero número de iterações que serão necessárias.
Codificação e Correção	Não há tempo gasto em planejamento, documentação, gestão de qualidade e cumprimento de standards. Requer pouca experiência.	Perigoso. Não há forma de assegurar qualidade e identificar riscos. Falhas fundamentais não percebidas imediatamente resultando em trabalho jogado fora.



Modelo Ágil





Os métodos ágeis iniciais—criado a priore em 2000— incluíam Scrum (1986), Crystal Clear, Programação extrema (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (1995).



Modelo RAD

