Stephen Willian Hawking (1942 - 2018)

Hawking morreu no mesmo dia do nascimento de Albert Einstein 14 de março. Nasceu em 8 de janeiro de 1942, 300 anos depois da morte de Galileu Galilei, pai da ciência moderna. Galileo morreu em 8 de janeiro de 1642, Arcetri, na Itália.

Universidade Católica de Brasília

Universidade Católica de Brasília

Engenharia de Software

Prof. Dr. Milton Pombo da Paz

Universidade Católica de Brasília

Pró-Reitoria Acadêmica

Curso de BES, BCC, ADS, BSI, GTI

Engenharia de Software

Prof. Dr. Milton Pombo da Paz

Plano de Ensino

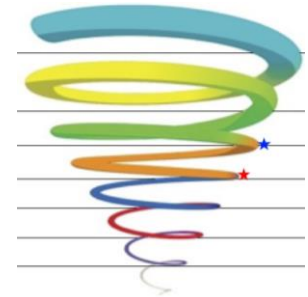
Apresentação da Disciplina e Plano de Ensino.

Definição dos temas de pesquisas e projetos.
Unidade 1 - Introdução
Introdução a Engenharia de Software.
Unidade 1 - Introdução
Importância da Engenharia de Software para o Desenvolvimento de Softwares.

Unidade 2 – Métodos
Tipos de Métodos.
Unidade 2 – Métodos
Tipos de Métodos.
Unidade 3 – Modelos de Processo
Modelos de Ciclo de Vida e de Processos de Software; e
Definição das Fases de um Processo e das Atividades de Apoio do Processo de Software.
Unidade 3 – Modelos de Processo
Processo Unificado.
Unidade 3 – Modelos de Processo
Modelos Ágeis de Desenvolvimento.
Unidade 3 – Modelos de Processo
Modelos Ágeis de Desenvolvimento.
Prova 1.
Unidade 3 – Modelos de Processo
Modelos Ágeis de Desenvolvimento.
Unidade 3 – Modelos de Processo
Modelos Ágeis de Desenvolvimento.
Unidade 3 – Modelos de Processo
Modelos Ágeis de Desenvolvimento.
Unidade 4 - Técnicas de Gestão
Gestão da Configuração de Software.
Unidade 4 - Técnicas de Gestão
Gestão da Configuração de Software.
Apresentação e entrega dos trabalhos.
Apresentação e entrega dos trabalhos.
Prova 2.
Prova Substitutiva.
Aula Síntese e divulgação de resultados.
Revisão da matéria e análise da disciplina no contexto do curso e da sociedade.

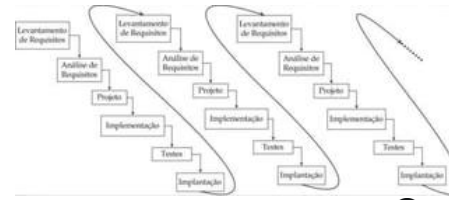
Início

Apresentação da Disciplina e Plano de Ensino.



Iremos trabalhar (ensino evolutivo e incremental):

- Interdisciplinaridade.
- Multidisciplinaridade.
- Transdisciplinaridade.



Início

Definição dos temas de pesquisas e projetos.

- Formação de grupos.
- Cada grupo terá um líder.
- A participação de todos é obrigatória.
- Não participou – sairá do grupo e fará um trabalho individual.
- Entregas no AVA de dois arquivos em PDF: documentação e slides.
- Todos os membros do grupo devem postar.

Instrumentos e Ponderação

- **AT1**: Avaliação individual e sem consulta com questões objetivas (80%) e dissertativas (20%);

As notas de atividades supervisionadas (trabalho final e pesquisas) não serão substituídas.

- **AT2**: Atividades individuais com entrega no AVA ou não. Serão determinadas em sala de aula pelo Professor os trabalhos práticos e/ou de pesquisas individuais. Os temas serão definidos em sala de aula. Os alunos deverão documentar e publicar no AVA. Serão avaliados o conteúdo e apresentação; e

- **Trabalho Final**: em formato de Projeto Final da disciplina será em grupo/equipe com entrega no AVA **como uma das atividades da AT2 da N2** e apresentação oral em sala de aula. Os temas serão definidos entre os estudantes e o Professor, em sala de aula. Serão avaliados conteúdo (documentação) e apresentação oral.

OBS: O Projeto Final é obrigatório e não tem substituição por outra atividade.

Instrumentos e Ponderação

O Art. 13 da Portaria indica que “As notas são indicadas com apenas uma casa decimal e não há arredondamento”. Ou seja, o aproveitamento final dos estudantes nas atividades avaliativas é expresso em escala numérica de 0 (zero) a 10 (dez), com intervalos de 0,1 (um décimo);

O Art. 12 da Portaria define que “Será considerado aprovado o discente que obtiver:

- I - Média (M) igual ou superior a 7,0 (sete); e
- II - Frequência mínima de 75% (setenta e cinco por cento) nas aulas”.

Ou seja, os seguintes casos podem ocorrer:

Média ≥ 7,0 - APROVADO.
Média < 7,0 - AVALIAÇÃO SUBSTITUTIVA (N3).

Instrumentos e Ponderação

AVALIAÇÃO – RECUPERAÇÃO

O Art. 16 da Portaria diz que "No Sistema de Avaliação (SA) presencial do formato convencional o discente que não obtiver a média (MF) igual ou superior a 7,0 (sete) terá direito à realização da Avaliação Substitutiva (N3), que substituirá a menor nota bimestral".

Nesse caso, a média final (MF) da unidade curricular será novamente calculada da seguinte forma:

MF = (**N3**+N2+PPD) Ou MF = (N1+**N3**+PPD)

Em que:

§ 1º A frequência mínima de 75% (setenta e cinco por cento) às aulas ministradas é condicionante para a realização da Avaliação Substitutiva.

§ 2º **N3 = Nota da avaliação substitutiva** vale 4,5 (quatro pontos e cinco décimos) e abrange todo o conteúdo programático e atividades desenvolvidas no semestre. Esta não se aplica ao PPD.

§ 3º A Avaliação Substitutiva (N3) não se aplica ao PPD" (1).

(1) Observação: Considerando o caráter técnico especifico desta disciplina, **a Avaliação Substitutiva (N3), para quem não fez o Trabalho Final, consistirá no desenvolvimento de um Projeto com tema definido pelo Professor** que irá substituir a atividade **da AT2 da N2** correspondente.

Instrumentos e Ponderação

AVALIAÇÃO – RECUPERAÇÃO

O Art. 19 da Portaria orienta que "A realização das avaliações, bem como as devolutivas das notas aos discentes, deve estar prevista no cronograma do Plano de Ensino da unidade curricular".

§ 1º O período de realização das avaliações bimestrais (N1 e N2), avaliação substitutiva (N3) e os respectivos prazos para devolutiva está previsto em Calendário Acadêmico" e no Plano de Trabalho deste Plano de Ensino.

Conforme a atualização da Portaria nº 01 do Sistema de Avaliação (2024/2), ações institucionais ou no âmbito dos cursos (designadas pela Coordenação de Curso) poderão resultar em pontuação extra a média da unidade curricular (MF).

Em 2024/2, será aplicado Exame Unificado do Grupo UBEC e para ele será atribuído até 1,0 (um) ponto extra na média (M) de todas as unidades curriculares em que o estudante estiver matriculado, a exceção do Trabalho de Conclusão de Curso (TCC).

Assim, a nota final (média) da unidade curricular será determinada da seguinte forma:

MF = (N1 + N2 + PPD) + PE

Instrumentos e Ponderação

OBSERVAÇÕES

Não serão aceitos trabalhos desenvolvidos a partir de ferramentas de automatização de pesquisas como as de IA;

A nota da pesquisa está vinculada ao cumprimento de todas as atividades, solicitadas em sala de aula, referentes a cada trabalho;

As **datas de entrega das pesquisas e do projeto serão rigorosamente seguidas**. As pesquisas deverão ser realizadas no prazo divulgado em sala de aula, pois poderão ser motivos de discussão com todos os estudantes;

Pesquisas e projeto entregues fora do prazo (no máximo até a próxima aula) **terão seu valor final correspondente a 50% do valor total avaliado**, exceto a apresentação de seminários;

Não haverá avaliação de substituição das pesquisas e dos projetos;

As pesquisas e o projeto deverão ser documentados de acordo com as regras de formato divulgadas em sala de aula;

A participação no Seminário é obrigatória e faz parte da avaliação. A ausência na apresentação dos seminários que venham a ser programados implica em nota zero para a apresentação, comprometendo a nota final;

Não serão aceitos trabalhos já apresentados em semestres anteriores. **Só serão aceitos trabalhos originais**, ou seja, serão desconsideradas as pesquisas que contenham cópias de textos da Internet, sem citação;

Instrumentos e Ponderação

PONTUAÇÃO EXTRA

O estudante que participar do **Exame Unificado do Grupo UBEC poderá ser atribuído até 1,0 (um) ponto** extra na média (M) desta unidade curricular.

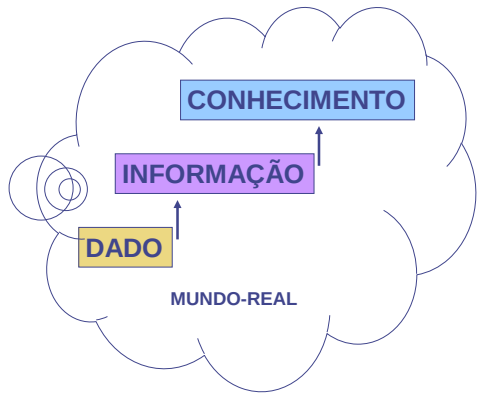
O referido exame será aplicado **presencialmente no dia 24 de outubro**, no turno da disciplina. Este é composto por questões objetivas de formação geral e de formação básica e específica.

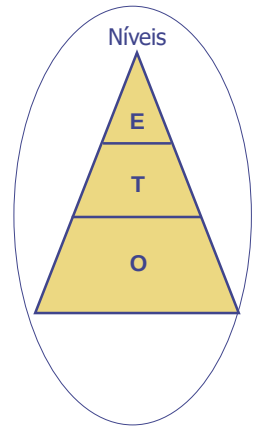
Importante:

- O plano de ensino é flexível e pode sofrer alterações ao longo do semestre, desde que acordadas antecipadamente com os estudantes.
- A descrição das atividades e metodologias vai descrita no ANEXO I – PLANO DE TRABALHO SEMESTRAL.
- Caso haja necessidade de reposição de aulas, as mesmas serão ministradas em datas e horários não previstos neste cronograma, sendo estas, antecipadamente, combinadas entre o professor e os estudantes desta disciplina.
- Os materiais para o acompanhamento das aulas, bem como toda a informação necessária será disponibilizada pelo sistema on-line (AVA). Portanto, é importante que o aluno se conecte semanalmente ao sistema para obter o material e se atualizar.
- Para melhor aproveitamento das aulas, **recomenda-se fortemente que o uso do celular** seja limitado a realização de atividades pedagógicas, quando solicitado e autorizado pelo professor.

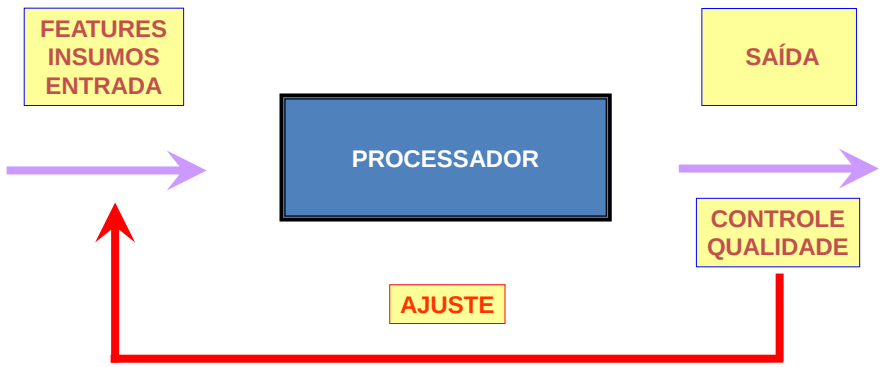
Fenomenologia - Johannes Hessen

Dado ➡ Informação ➡ Conhecimento ↻





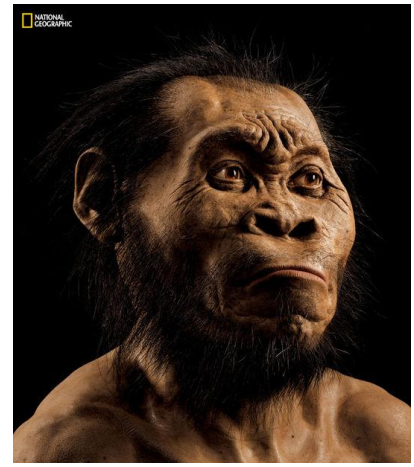
Processamento – Ciclo Universal



Necessidades

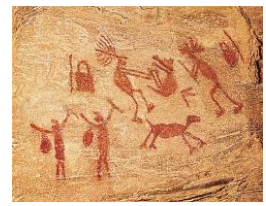
- **Necessidade humana de armazenar as experiências**

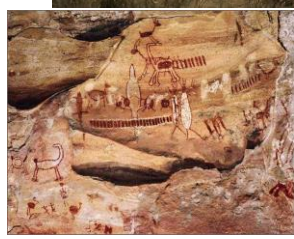
- **Pedra**
- **Madeira**
- **Papel**
- **Máquinas**



Necessidades

- **Necessidade humana de armazenar as experiências**



Necessidades

- **Necessidade humana de armazenar as experiências**
- Johannes Gutenberg: inventor, gravador e gráfico do Sacro Império Romano-Germânico.
- Revolução da imprensa.
- Segundo no mundo a usar a impressão por tipos móveis, por volta de 1439, após o chinês Bi Sheng no ano de 1040, e o inventor global da prensa móvel.



Unidade 1 - Introdução

Introdução à Engenharia de Software.

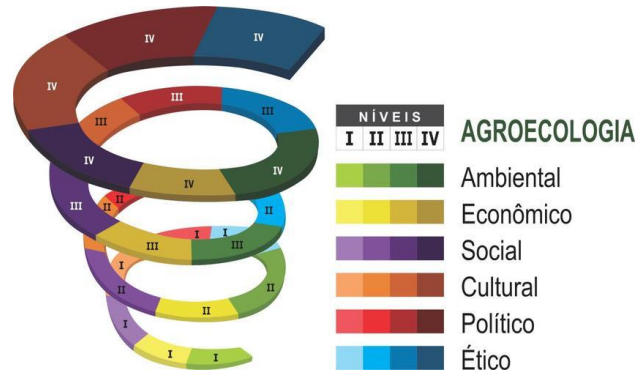
- É necessário organizar as organizações antes de se realizar qualquer automatização.

- Isso se dá por meio de um Planejamento Estratégico.

Unidade 1 - Introdução

Planejamento Estratégico Organizacional.

- Objetivos Estratégicos.
  - Macro-metas
    - Metas.
      - Ações.
        - Projetos
    - Macro-ações
      - Ações.
        - Projetos



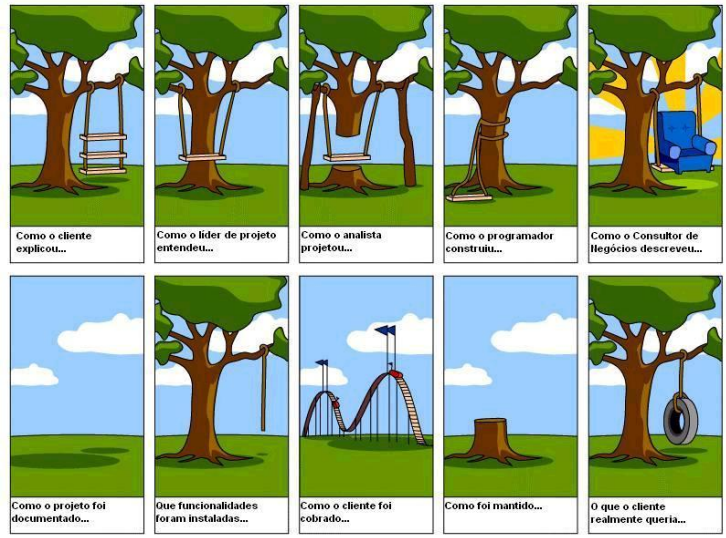
Unidade 1 - Introdução

Planejamento Estratégico Organizacional evita que:

- Hajam diversas linguagens na organização.
- Os interesses sejam conflitantes.
- Haja desperdício de recursos.
- Gere muito calor para realizar um trabalho.
- Haja confusão nas ações, metas, objetivos, táticas e estratégias.
- Vazio e interferência de poder.
- Haja ausência de definição de objetivos.
- Hajam diversos protocolos na organização.
- Os setores tenham indefinição de suas obrigações.
- O cliente fique sem atenção devida.
- Haja ausência de método, processo, ciclo de vida, técnicas e ferramentas.

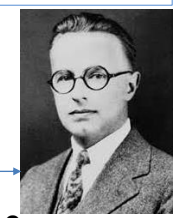
Unidade 1 - Introdução

PEO evita que:

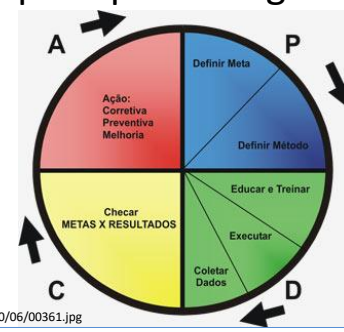


Unidade 1 - Introdução

Planejamento Estratégico Organizacional:



- Razão de tudo – Ciclo PDCA (Plan, Do, Check, Act).
- Criado por **Walter A. Shewhart** - 1930.
- Usado por **Willian E. Deming** no Japão pós 2º guerra mundial.
- **Revolucionou** a:
  - Administração.
  - Governança e Gestão.
  - Engenharia (sistemas).
  - Economia.



http://logisticaatual.files.wordpress.com/2010/06/00361.jpg

Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Informação.
- Estratégia.
- Tática.
- Operação.
- Requisitos de alto nível.
- Gestão da Informação.

Unidade 1 - Introdução

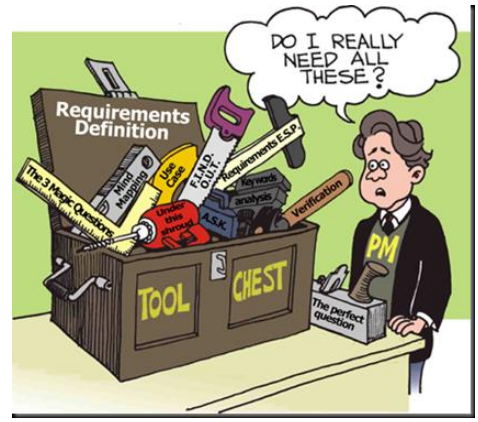
Planejamento Estratégico Organizacional:

- **Estratégia**: escolhas.
- **Táticas**: como desenvolver e implantar as escolhas estratégicas.
- **Operações**: realização das táticas.

Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Requisitos de alto nível.
  - Ética
  - Agilidade
  - Integração
  - Interação
  - Parceria
  - GI
  - Sustentabilidade



http://lh4.ggpht.com/-NQnYh0AfCdg/TfD_raiOH1I/AAAAAAAAAoE/ovXtHBsn1Bk/image%25255B4%25255D.png?imgmax=800

Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- **Gestão da Informação**: integração de todas as informações da organização para fins de otimizar as escolhas estratégicas, táticas e operações.

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- AGREGAR VALOR AO NEGÓCIO.
- OFERECER VANTAGEM COMPETITIVA.

Unidade 1 - Introdução

Planejamento Estratégico Organizacional:

- Missão.
- Visão de Futuro.
- Valores.
- Código de ética.
- Desenvolvimento:
  - Dimensões de análise.
  - Objetivos Estratégicos.
  - Macro-metas
  - Metas.
  - Macro-ações.
  - Ações.
  - Projetos
- Mapa de Rotas Estratégicas.
- Mapa e Rotas Tecnológicas.

http://www.sbcoaching.com.br/blog/wp-content/uploads/2013/11/mulher-binoculos-visao.jpg

Unidade 1 - Introdução

Planejamento Estratégico da TI - PETI.

O alinhamento estratégico do projeto com as diretrizes organizacionais.

- Alinhado às estratégias de negócio.

Alinhado ao PE.

Unidade 1 - Introdução

Tipos de Sistemas: operacionais, táticos e estratégicos.

- Sistemas de Informação Executiva.
- Sistemas de Informação Gerencial.
- Sistemas de Informação Operacional.

Unidade 1 - Introdução

Ambiente de Engenharia de Software.

- Empresa/organização/instituição – é um “negócio”.
- Planejamento Estratégico.
- Adoção de métodos, processos, técnicas e tecnologias.
- Priorização de desenvolvimento de SI.
- Criação de projetos de sistemas.
- Desenvolvimento de projetos.

Unidade 1 – Introdução

Hardware: duro, tangível, máquina.

- ✓ Impressora.
- ✓ Monitor.
- ✓ Teclado.
- ✓ Mouse.
- ✓ CPU

Unidade 1 - Introdução

O que é Software?

- Definição
  - *1º* - instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados;
  - *2º* - estruturas de dados que permitem a manipulação das informações;
  - *3º* - documentos que descrevem a operação e uso dos programas.
- MICHAELIS: Software (*inglês – informática*)
  - "Qualquer programa ou grupo de programas que instrui o hardware sobre a maneira como ele deve executar uma tarefa".

Unidade 1 – Introdução

Software: mole, intangível.

Software Básico:

- ✓ Sistema operacional: Windows, Linux, Mac OS, Solaris, Android, IoS.
- ✓ Drives de impressora, mouse, teclado, monitor.

Software Aplicativo:

- ✓ Sistema de Controle de Estoque.
- ✓ Sistema de Folha de Pagamento.
- ✓ Word, Excel, Powerpoint, AutoCAD, Firefox, Chrome.

Unidade 1 – Introdução

Tipos de Softwares:

- **SOFTWARE DE SISTEMA**: constituído pelos sistemas operacionais (SO). Auxiliam o usuário, para passar os comandos para o computador. Ele interpreta nossas ações e transforma os dados em códigos binários, que podem ser processados.
- **SOFTWARE APLICATIVO**: os programas utilizados para aplicações dentro do SO, que não estejam ligados com o funcionamento do mesmo. Exemplos: Word, Excel, Paint, Bloco de notas, calculadora.
- **SOFTWARE DE PROGRAMAÇÃO**: são softwares usados para criar outros programas, a parir de uma linguagem de programação, como Java, PHP, Pascal, C+, C++, entre outras.
- **SOFTWARE DE TUTORIAL**: São programas que auxiliam o usuário de outro programa, ou ensine a fazer algo sobre determinado assunto.
- **SOFTWARE DE JOGOS**: São Softwares usados para o lazer, com vários tipos de recursos.
- **SOFTWARE ABERTO**: É qualquer dos softwares acima, que tenha o código fonte disponível para qualquer pessoa.
- **SOFTWARE DE EXERCITAÇÃO:** Similar ao software tutorial, mas aqui o usuário conta com maior interatividade através de resposta diante de questões que serão apresentadas.
- **SOFTWARE DE INVESTIGAÇÃO:** Nesta categoria se enquadram todos os softwares que permitem a localização de diversas informações a respeito de diversos assuntos. As enciclopédias são normalmente classificadas nesta categoria.
- **SOFTWARE DE SIMULAÇÃO:** Geralmente utilizados para simulações de situações da vida real. Dentre os mais conhecidos estão os simuladores de vôo e os gerenciadores de cidades, muito conhecidos pelo mundo jovem nos jogos, mas, quando pensamos em simuladores podemos errar a ligá-los somente a jogos, hoje eles são bastante usados em situações de treinamentos de pessoas para enfrentar casos no seu dia-a-dia.

Unidade 1 - Introdução

Aplicações de Software

Pressman, página 20

- Software Básico
- Software de Tempo Real
- Software Comercial
- Software Científico ou de Engenharia
- Software Embutido
- Software de Computador Pessoal
- Software de Inteligência Artificial

Unidade 1 - Introdução

Características do Software - 1

O Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico:

- Custos são concentrados no trabalho de engenharia.
- Projetos não podem ser geridos como projetos de manufatura.
- "Fábrica de Software!"

Unidade 1 - Introdução

Características do Software - 2

- Software não desgasta!
  - Software não é sensível aos problemas ambientais que fazem com que o hardware se desgaste.
  - Ver curvas de falha, páginas 14 e 15 do Pressman.
  - Toda falha indica erro de projeto ou implementação: manutenção do SW é mais complicada que a do HW.

Unidade 1 - Introdução

Características do Software - 3

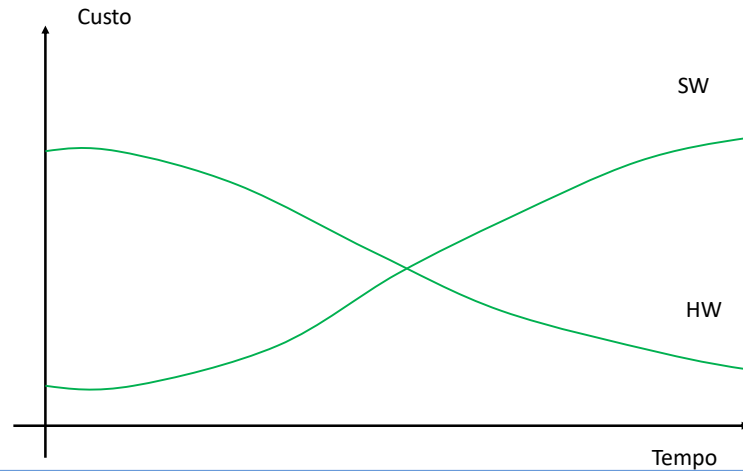
- A maioria dos softwares é feita sob medida e não montada a partir de componentes existentes.
- != Hardware.
- Situação esta mudando:
  - Orientação a objeto.
  - Reusabilidade: diminui custos e melhora projetos.

Unidade 1 - Introdução

"O Software ultrapassou o Hardware como chave para o sucesso de muitos sistemas baseados em computador" (Pressman, pg. 3, 1992).

Unidade 1 - Introdução

Historicamente ...



Unidade 1 - Introdução

A importância do Software

- Durante as 3 primeiras décadas da era do computador, o principal desafio era desenvolver um **HARDWARE** de baixo custo e alto desempenho.
- Hoje o desafio é melhorar a qualidade (e reduzir os custos) das soluções baseadas em **SOFTWARE**!

Unidade 1 - Introdução

O Software é o que faz a diferença!!!

- *Completeza* da informação.
- *User-friendlyness.*
- *Web-enhanced.*
- Inteligência.
- Funcionalidade.
- Compatibilidade.
- Suporte.

Unidade 1 - Introdução

A evolução do Software

Computação → - Nova Revolução Industrial (*Toffler*)

- 3a. Onda

Unidade 1 - Introdução

Revolução Industrial
Primeira Onda

- Ferro (Darby, 1709)
- Máquina a vapor:
  - Inventada (Newcomen, 1712)
  - Aperfeiçoada (WATT, 1766 - ’69 -’82)
- Mecanização da indústria têxtil:
  - Tear Mecânico (Kay, 1722)
  - Maquina de fiar (Hargreaves, 1764)
- Aspectos sociais, políticos e econômicos
  - Têxteis, Carvão e Ferro

Unidade 1 - Introdução

Revolução Industrial
Segunda Onda

- Aço (Bessemel, 1856 e 1885 - Liga)
- Locomotiva a Vapor (Rede de Transporte - 1830)
- Máquina de Costura (SINGER,1851)
- Motor a combustão interna:
  - Primeiro eficiente (OTTO, 1876)
  - Produção automobilística em massa (Daimler e Benz, 1896)
- Desemprego e fim da escravidão

Unidade 1 - Introdução

Revolução Industrial

Terceira Onda

- Energia Nuclear (Fermi, 1942)
- Uso Industrial/Comercial da Eletricidade
- Computadores Eletrônicos (ENIAC 1946)
- Transistor (Shockley, et al., 1948)

Sociedade Industrial → (transformação) → Sociedade da Informação

Unidade 1 - Introdução

Filosofando...

- A mudança de uma sociedade industrial para uma baseada na informação é uma *Radical Mudança Econômica*:
  - Material tem menos valor e Informação tem mais valor

Antes: quanto menos pessoas possuísse algo, maior o valor.	**Hoje**: quanto mais pessoas possuem algo, maior o valor.

Unidade 1 - Introdução

A evolução do software

- Software é dividido em 4 Eras:
  - Primeiros anos: 1950 - 1965
  - Segunda Era: 1965 - 1975
  - Terceira Era: 1975 - 1988
  - Quarta Era: 1988 - ...

Unidade 1 – Introdução

Engenharia

MICHAELIS

- Engenharia = Engenho + aria
  - Engenho (*lat ingeniu*) :
    - 1. capacidade para discorrer ou inventar com prontidão;
    - 2. talento, aptidão natural; gênio.
  - Sufixo -aria (*lat -ariu + ia*):
    - 1. designa oficina, loja, ação, condição,...
  - "Arte de aplicar os conhecimentos científicos à invenção".

Unidade 1 - Introdução

Necessidade da Engenharia de Software

- **Automatização das Atividades da rotina social**.
  - Comércio.
  - Indústria.
  - Educação.
  - Engenharia.
  - Administração.
  - Economia.
  - Aviação.
  - Energia.
  - Pesquisa.
  - Astronomia.
  - Inteligência Artificial.
  - Trânsito.
  - Etc.

Diferenças:

- **Informatização**: computador no escritório.
- **Automatização**: processos de negócio.
- **Automação**: processos de negócio com outras máquinas.

Unidade 1 - Introdução

Fundamentos da Engenharia de Software

- Baseada nos princípios da Engenharia de Produção.
- Visa o **desenvolvimento** de softwares **como engenhos** com qualidade e precisão.

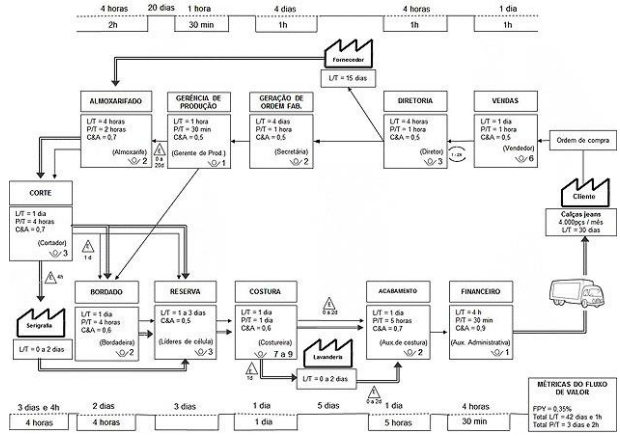
Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Necessidades da Engenharia de Software

Entender, analisar, projetar, implementar, implantar e manter.

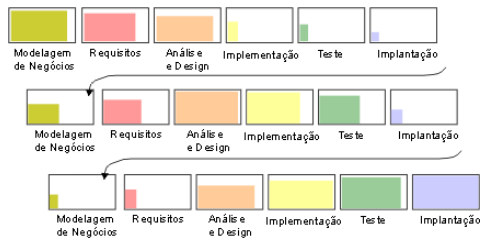


http://www.revistaespacios.com/a13v34n10/13341004.html

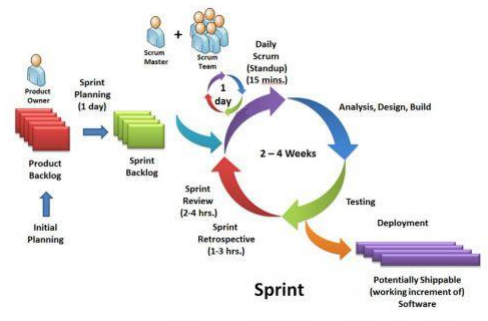
Unidade 1 - Introdução

Necessidades da Engenharia de Software

Entender, analisar, projetar, implementar e implantar.



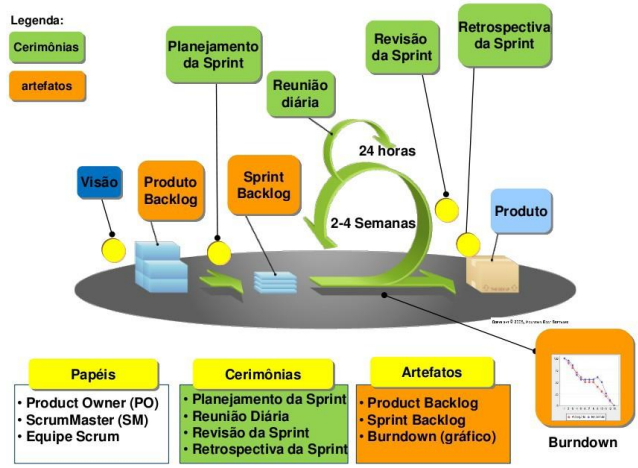
http://www.funpar.ufpr.br:8080/rup/process/workflow/manageme/co_phase.htm



http://www.nedimtaktakkimdir.com/?pnum=179&pt=Scrum%20Nedir?

Unidade 1 - Introdução

Necessidades da Engenharia de Software



https://arimuya.blogspot.com.br/2014/08/engenharia-de-software-metodologia-de.html

Unidade 1 - Introdução

Problemas da Engenharia de Software

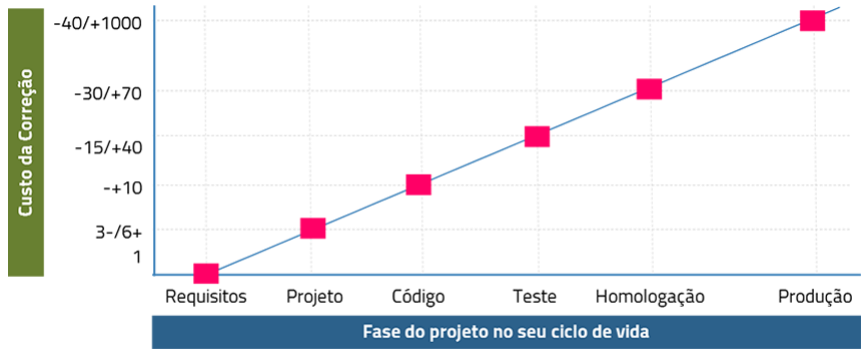
Afeta:

- Marca da instituição.
- Negócio.
- Reputação da:
  - Instituição.
  - Computação.
  - Profissionais.
- Custos operacionais.
- Qualidade.

Unidade 1 - Introdução

Problemas da Engenharia de Software

Custo de manutenção



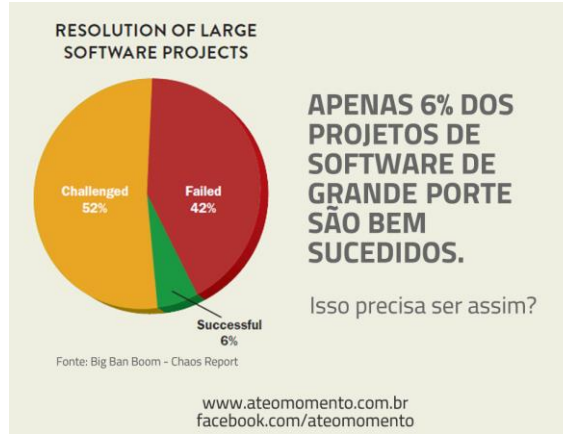
facebook.com/ateomomento
www.ateomomento.com.br

http://www.ateomomento.com.br/porque-os-projetos-projetos-dao-errado/

Unidade 1 - Introdução

Problemas da Engenharia de Software

Projetos bem sucedidos.



http://www.ateomomento.com.br/porque-os-projetos-projetos-dao-errado/

Unidade 1 - Introdução

Problemas da Engenharia de Software

Alguns dos mais famosos erros de softwares da história.

Erros de software custam, só para a economia dos EUA, cerca de 60 milhões dólares anualmente em retrabalho, perda de produtividade e danos reais.

Falhas de software podem ser caras, embaraçosas, destrutivas e mortais.

A seguir alguns “desastres” que aconteceram no mundo do software nos últimos anos, em ordem cronológica:

https://www.profissionaistI.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

1. Problemas no Mariner (1962)

Custo: 18,5 milhões dólares.

Desastre: Mariner, um foguete com uma sonda espacial para Vênus, foi desviado de seu percurso de voo logo após o lançamento. O controle da missão destruiu o foguete 293 segundos após a decolagem.

Causa: Um programador, ao passar para o computador uma fórmula que haviam lhe entregue escrita manualmente, se esqueceu de uma barra. Sem ela, o software tratava variações normais de velocidade como se fossem sérios problemas, causando falhas por tentativas de correções que acabaram por enviar o foguete fora do curso.

https://www.profissionaistI.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

2. Hartford Coliseu Desmorona (1978)

Custo: 70 milhões de dólares, além de outros danos de 20 milhões para a economia local.

Desastre: Poucas horas depois de milhares de fãs deixarem o Coliseu Hartford, o teto de treliça de aço desabou sob o peso da neve molhada.

Causa: O programador do software CAD, utilizado para projetar o coliseu, incorretamente assumiu que o suporte do telhado de aço enfrentaria apenas compressão natural. Mas quando um dos suportes inesperadamente recebeu um bloco de neve, este desencadeou uma reação em cadeia que derrubou o telhado de outras seções como dominós.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

3. CIA distribui gás aos soviéticos (1982)

Custo: Milhões de dólares e danos significativos a economia soviética.

Desastre: O software de controle se descontrolou e produziu uma intensa pressão no gasoduto Trans-Siberian, resultando na maior explosão não-nuclear da história.

Causa: Agentes da CIA supostamente plantaram um erro em um sistema de computação canadense comprado pelos soviéticos para controlar seus gasodutos. A compra fazia parte de um plano estratégico soviético para roubar ou secretamente obter tecnologia sensível dos EUA. Quando a CIA descobriu a compra, eles sabotaram o software para que passasse inspeção soviética, mas falhava em operação.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

4. 3ª Guerra Mundial (1983)

Custo: Quase toda a humanidade.

Desastre: O sistema de alerta precoce soviético falsamente indicou que os Estados Unidos tinham lançado cinco mísseis balísticos. Felizmente, o oficial de serviço soviético tinha uma "sensação esquisita no estômago" e, se os EUA estavam realmente atacando, eles lançariam mais de cinco mísseis, por isso ele relatou o aparente ataque como um alarme falso.

Causa: Um bug no software soviético falhou ao detectar reflexos solares como falsos mísseis.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

5. Máquina medicinal mata (1985)

Custo: Três mortos e três seriamente feridos.

Desastre: A máquina de radiação canadense Therac-25 irradiou doses letais em pacientes.

Causa: Por causa de um bug sutil chamado de "*condição de corrida*", um técnico acidentalmente configurou o Therac-25 de modo que o feixe de elétrons seria como um fogo de alta potência.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

6. Crash na Wall Street (1987)

Custo: $500 bilhões em um dia.

Desastre: Em 19 de outubro de 1987, o índice Dow Jones caiu 508 pontos, perdendo 22,6% de seu valor total. Esta foi a maior perda que Wall Street já sofreu em um único dia.

Causa: Um mercado em grande alta foi interrompido por uma série de investigações conduzidas pela SEC e por outras forças do mercado. Como os investidores fugiram de ações investigadas, um número muito grande de ordens de venda foram gerados pelos computadores, quebrando sistemas e deixando os investidores efetivamente cegos.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

7. Linhas da AT&T "morrem" (1990)

Custo: 75 milhões de ligações perdidas e 200 reservas aéreas perdidas.

Desastre: Um switch dos 114 centros de swiches da AT&T sofreu um problema mecânico que fez com que todo o seu centro fosse desligado. Quando o seu centro voltou a ativa, enviou uma mensagem aos outros, o que causou o desligamento dos outros centros e deixou a empresa parada por 9 horas.

Causa: Uma única linha de código em uma atualização de software implementada para acelerar chamadas causou um efeito cascata que desligou a rede.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

8. Patriot Acaba com Soldados (1991)

Custo: 28 soldados mortos e 100 feridos.

Desastre: Durante a primeira Guerra do Golfo, um sistema (Patriot) americano de mísseis na Arábia Saudita falhou ao interceptar um míssil vindo do Iraque. O míssil destruiu acampamentos americanos.

Causa: Um erro de arredondamento no software calculou incorretamente o tempo, fazendo com que o sistema Patriot ignorasse os mísseis Scud de entrada.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

9. Pentium Falha em uma Divisão Longa (1993)

Custo: $475 milhões e a credibilidade de uma empresa.

Desastre: O Pentium da Intel ocasionalmente cometeu erros ao dividir números de ponto flutuante em um intervalo específico. Por exemplo, dividindo 4195835.0/3145727.0 obteve 1,33374 ao invés de 1,33382, um erro de 0,006%. Embora o bug afetasse apenas alguns usuários, se tornou um pesadelo nas relações públicas. Com uma estimativa de 5 milhões de chips defeituosos em circulação, a Intel se ofereceu para substituir os chips Pentium apenas para os consumidores que poderiam provar que eles precisavam de alta precisão. Contudo a Intel acabou substituindo os chips de qualquer um que reclamou.

Causa: O divisor na unidade de ponto flutuante do Pentium tinha uma tabela de divisão falha, faltando cerca de cinco mil entradas, resultando nestes erros de arredondamento.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

10. Ariane Rocket Goes Boom (1996)

Custo: $500 milhões.

Desastre: Ariane 5, o mais novo foguete da Europa não-tripulado, foi intencionalmente destruído segundos após seu lançamento em seu voo inaugural. Também foram destruídos quatro satélites científicos para estudar como o campo magnético da Terra interage com os ventos solares.

Causa: O desligamento ocorreu quando o computador de orientação tentou converter a velocidade do foguete de 64-bits para um formato de 16 bits. O número era muito grande, o que resultou em erro de estouro. Quando o sistema de orientação desligou, o controle passou para uma unidade idêntica redundante, que também falhou porque nele estava correndo o mesmo algoritmo.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

11. Estudo Desastroso (1999)

Custo: Credibilidade da ciência.

Desastre: Neste caso, o software utilizado para analisar os desastres tinha um desastre próprio. O jornal *The New England Journal of Medicin*e relatou aumento das taxas de suicídio depois de graves desastres naturais. Infelizmente, estes resultados mostraram-se incorretos.

Causa: Um erro no programa mostrava a taxa de suicídios por ano como o dobro do seu valor real, o que foi suficiente para inutilizar toda a pesquisa.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

12. Passaportes britânicos para lugar nenhum (1999)

Custo: £12.6 milhões.

Desastre: A agência de passaportes do Reino Unido implementou um sistema da Siemens que falhou ao emitir documentos para meio milhão de cidadãos britânicos. A agência teve que pagar milhões ao governo para compensar a raiva da população.

Causa: A Agência lançou seu novo sistema sem testá-lo de forma adequada ou treinar seus funcionários. Ao mesmo tempo, uma mudança na lei exigia que todos os menores de 16 anos viajando ao exterior deveriam obter um passaporte, resultando em um aumento enorme na procura de passaportes, o que sobrecarregou o sistema .

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

13. Bug do Milênio (1999)

Custo: $500 bilhões.

Desastre: O desastre de um homem é a fortuna de outro, como demonstra o Bug do Milênio. Empresas gastaram bilhões com programadores para corrigir uma falha no software legado. Embora nenhum falha significativa ocorreu, a preparação para o Bug do Milênio teve um custo significativo e impacto no tempo em todas as indústrias que usam a tecnologia computacional.

Causa: Para economizar espaço de armazenamento de computador, softwares legados muitas vezes armazenavam anos para datas com números de dois dígitos, como 99 para 1999. Esses softwares também interpretavam 00 para significar 1900, em vez de 2000, por isso, quando o ano de 2000 veio, bugs apareceriam.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

14. Tratamento de Câncer Mortal (2000)

Custo: 8 pessoas mortas e 20 seriamente feridas.

Desastre: O software de radiação da empresa Multidate calculou mal a dosagem de radiação que deveria ser enviada, expondo pacientes a níveis fatais de radiação. Os físicos que foram indicados para checar as máquinas foram condenados a morte.

Causa: O software calculava a dosagem de radiação baseando-se na ordem de entrada dos dados, e algumas vezes enviava o dobro da dose do que deveria.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

15. EDS Drops Child Support (2004)

Custo: £539 milhões (e a conta ainda cresce)

Desastre: A grande empresa de serviços EDA desenvolveu um sistema para o Centro de Suporte à Crianças do Reino Unido (CSA) que acidentalmente pagou a mais 1.9 milhões de pessoas, recebeu em menos de 700.000 casos, registrando uma lista incansável de erros.

Causa: A EDS apresentou um **sistema de TI** complexo e grande demais para o CSA, ao mesmo tempo em que tentava reestruturar a agência.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

16. Desastre no FBI (2005)

Custo: $105 milhões.

Desastre: O FBI desistiu da revisão de um sistema após quatro anos de esforço. O projeto Arquivo Virtual foi um maciço sistema de software integrado para agentes compartilharem arquivos de casos e outras informações.

Causa: Má gestão e uma tentativa de construir um projeto de longo prazo sobre tecnologia ultrapassada, resultou em um sistema complexo e inutilizável.

https://www.profissionaisti.com.br/2012/01/alguns-dos-mais-famosos-erros-de-softwares-da-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

http://app.crowdtest.me/10-falhas-software-marcaram-historia/

10 falhas de software que marcaram a história

Um canal do Youtube, o **Alltime10s**, preparou um vídeo com 10 falhas de software que marcaram a história. Achamos bastante interessante a lista.

As falhas listadas aqui causaram grandes transtornos para muita gente.

Unidade 1 - Introdução

Problemas da Engenharia de Software

10. Apagão de U$ 6 bilhões

Em 2003, os EUA enfrentaram um super apagão no nordeste do país. Conhecido como ” The Great Northeast Blackout”, o incidente foi causado por uma falha no sistema de alarme. Incrível, mas foi o suficiente para deixar 50 milhões de pessoas sem energia e provocar 11 mortes. O prejuízo chegou a US$6 bilhões para o governo americano.

9. Recall da Honda

Por causa de um defeito de programação no sistema dos carros, a Honda teve que realizar um recall de mais de 2 milhões de automóveis. O fato, ocorrido em 2011, custou alguns milhões de dólares à empresa japonesa. O airbag era ativado com muita força e pelo componente errado.

http://app.crowdtest.me/10-falhas-software-marcaram-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

8. Google como malware

Em 2009, uma barra invertida adicionada por um programador às URLs que eram direcionadas para o buscador do Google provocou a identificação do site como sendo um malware no mundo inteiro por cerca de 1 hora. Prejuízo de quase US$ 3 milhões.

7. Trânsito ruim

A justiça da Califórnia convocou 1,2 mil pessoas para trabalharem como júri no mesmo horário e no mesmo dia. O incidente ocorrido em 2012 foi fruto de uma falha no sistema da justiça da Califórnia. O trânsito nas estradas que davam acesso à região do júri ficou engarrafado e provocou a ira de muitos motoristas.

http://app.crowdtest.me/10-falhas-software-marcaram-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

6. Promoção da Pepsi

A promoção da Pepsi em 1992 era a seguinte : quem tirasse a tampinha com o número 349 impresso, ganharia uma premiação em dinheiro. Um problema no sistema das máquinas de impressão resultou na distribuição de 800 mil tampinhas com a numeração premiada nas Filipinas. Na época, a empresa não entregou os prêmios, o que provocou bastante revolta.

5. Epidemia no World of Warcraft

Há quase 10 anos , os desenvolvedores do game World of Warcraft espalharam um vírus dentro do jogo, chamado de “Corrupted Blodd”. A doença “de brincadeira” se espalhou no jogo de maneira incontrolada e imprevista, o que provocou a morte de vários personagens no mundo inteiro. Os jogadores ficaram muito irritados com o jogo.

http://app.crowdtest.me/10-falhas-software-marcaram-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

4. Mortos vivos

Um hospital dos EUA, chamado St. Mary’s Mercy, apresentou erros no sistema e declarou a morte de 8,5 mil pacientes em 2002. O sistema disparou o envio de contas com o atestado de óbito para os parentes e notificações para o governo e empresas.

3. Radiação excessiva

Entre os anos de 1985 e 1987, os hospitais dos EUA utilizavam um aparelho chamado Therac-25 para o tratamento com radiação contra o câncer. Havia um erro de programação no software que aplicava uma radiação 100 vezes maior do que a recomendada nos pacientes. Foram registradas 6 mortes nesse período.

http://app.crowdtest.me/10-falhas-software-marcaram-historia/

Copyright
Todos os Direitos Reservados

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

2. Quase faliu

A KCG (Knight Capital Group) trabalha há anos com investimentos. Em 2012, quase foi à falência devido a uma falha em um novo software que a empresa comprou. O Software gerou milhares de negociações que não poderiam ser feitas. Em meia hora, a KCG perdeu US$440 milhões e ficou em situação crítica.

1. Quase uma terceira guerra mundial

No ano de 1979, uma terceira guerra mundial quase aconteceu. Isso porque os sistemas de defesa dos EUA identificaram que a União Soviética estava preparada para um ataque com mísseis contra o país, de forma que uma retaliação estava sendo programada. Felizmente, antes que o pior acontecesse, foi identificado que tratava-se do programa de simulação que foi iniciado acidentalmente.

http://app.crowdtest.me/10-falhas-software-marcaram-historia/

Unidade 1 - Introdução

Problemas da Engenharia de Software

O Caso do DABHS (Denver Airport Baggage Handling System)

Custo do projeto: US$ 4.9 bilhões

- 100 mil passageiros por dia.
- 1,200 vôos.
- 53 milhas quadradas.
- 94 portões de embarque e desembarque.
- 6 pistas de pouso / decolagem.

Características:

– 4000 Telecars.

– 21 milhas de trilhos.

– 5000 Olhos Eletrônicos, 400 Receptores de Rádio, 56 Scanners, 100 computadores em rede.

Estava planejado que o Aeroporto de Denver abriria em 31 de Outubro de 1993.

- Problemas no desenvolvimento do DABHS fizeram que o aeroporto abrisse em 28 de Fevereiro de 1995.

- Prejuízos devido ao atraso:

Capacidade do sistema foi drasticamente diminuída.

Aproximadamente $500.000.000 ($1.1M por dia).

Erros no sistema automático de transporte de bagagens (*misloaded*, *misrouted*, *jammed*):

- Atraso na abertura do aeroporto com custo total estimado em US$360 Milhões.

86 milhões para consertar o sistema.

Unidade 1 - Introdução

Problemas da Engenharia de Software

Therac-25

Equipamento de Radioterapia, controlado por computador criada pela empresa AECL (Atomic Energy of Canada Limited).

Entre 1985 e 1987 se envolveu em 6 acidentes, causando mortes por overdoses de radiação.

Causas:

Software foi adaptado de uma antecessora, Therac-6:

- falhas por falta de testes integrados. O código do software não havia sido revisado/testado independentemente.

- falta de documentação. O projeto do software não havia sido documentado com detalhes suficientes para permitir o entendimento dos erros.

– A documentação do sistema fornecida aos usuários não explicava o significado dos códigos de erro que a máquina retornava.

- página 382 do Pfleeger.

Fonte: http://pt.wikipedia.org/wiki/Therac-25

Unidade 1 - Introdução

Problemas da Engenharia de Software

Ariane 5

REFERÊNCIAS

http://magnum.ime.uerj.br/~demoura/Especializ/Ariane/

SOCIEDADE BRASILEIRA DE MATEMÁTICA APLICADA E COMPUTACIONAL
http://www.sbmac.org.br/

ESA - EUROPEAN SPACE AGENCY
http://www.esrin.esa.it/esa/ariane/ariane5.html

Confiabilidade e Qualidade de Software
http://www2.isec.pt/~fmoreira/www/Cadeiras/CQS/Bibliografia.htm

Relatório oficial da Queda do Voo Inaugural do Ariane5
http://www2.isec.pt/~fmoreira/www/Cadeiras/CQS/ariane5rep.html

www.fotonica.ufpe.br/ensino/engcomputadores

O Boletim SBMAC divulga uma versão condensada do relatório sobre a falha no primeiro lançamento do Ariane 5, em parte baseada no texto do SIAM News, V29 N.8, out/96.
http://www.sbmac.org.br/com-fig/public/bol/bol-2/artigos/ariane5.html

UCB – Engenharia de Software - Prof. Milton

Unidade 1 - Introdução

Problemas da Engenharia de Software

Ariane 5

Projeto da Agência Espacial Europeia que custou:

10 anos. US$ 8 Bilhões. Carga avaliada em US$ 500 Milhões.

Capacidade 6 toneladas. Garante supremacia europeia no espaço.

Em 4 de junho de 1996, explosão 40 segundos após o lançamento - foguete se autodestruiu junto com a carga.

Causa:

– O correu um *runtime error* (erro de execução - out of range, overflow) e ambos computadores se desligaram.

Os computadores principal e back-up deram shut-down ao mesmo tempo.

Um programa que convertia um valor em ponto flutuante para um inteiro de 16 bits recebeu como entrada um valor que estava fora da faixa permitida.

– Inclusive o resultado da conversão não era mais necessário após a decolagem.

O veículo detonou suas cargas explosivas de autodestruição e explodiu no ar.

Porque ele estava se quebrando devido às forças aerodinâmicas.

O foguete tinha perdido o controle de direção (atitude).

Unidade 1 - Introdução

Problemas da Engenharia de Software

Plataforma Alpha – Mar do Norte

https://www.rsem.com.br/aprendendo-com-a-experiencia-relembrando-o-acidente-de-piper-alpha/

Unidade 1 - Introdução

Problemas da Engenharia de Software

Navio Mercante

- Falha de software no sistema do leme.

Unidade 1 - Introdução

Quais são as causas dos problemas?

- ✓ A sofisticação do software ultrapassou nossa capacidade de construção.
- ✓ Nossa capacidade de construir programas não acompanha a demanda por novos programas.
- ✓ Nossa capacidade de manter programas é ameaçada por projetos ruins.

Unidade 1 - Introdução

Causas óbvias.

- ✓ Não dedicamos tempo para coletar dados sobre o desenvolvimento do software - resulta em estimativas “a olho”.
- ✓ Comunicação entre o cliente e o desenvolvedor é muito fraca.
- ✓ Falta de testes sistemáticos e completos.

Unidade 1 - Introdução

Causas menos óbvias.

- ✓ O Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico (característica 1).
- ✓ Gerentes sem *background* em desenvolvimento de SW.
- ✓ Profissionais recebem pouco treinamento formal.
- ✓ Falta investimento (em ES).
- ✓ Falta métodos e automação.

Unidade 1 - Introdução

Uma Crise no horizonte

- A indústria de Software tem tido uma “crise” que a acompanha há quase 30 anos:
  - Aflição Crônica != Crise
- Problemas não se limitam ao software que não funciona adequadamente, mas abrange:
  - desenvolvimento, testes, manutenção, suprimento, etc.

Unidade 1 - Introdução

Surgimento da Engenharia de Software

A crise do software:

Esses problemas citados são conhecidos mundialmente como “The Software Crisis (CRISE DESOFTWARE)”.

A crise foi identificada nos anos 60.

Problemas persistem até hoje:

– Previsão pobre (tempo, custo, recurso).
– Programas de baixa qualidade (Pressa).
– Programas de baixa qualidade (Pressa).
– Alto custo para manutenção (Corretivas e Evoluções).
– Duplicação de esforço.

Unidade 1 - Introdução

Surgimento da Engenharia de Software

A crise do software:

1968 na NATO Conference on Software Engineering (Conferência sobre Engenharia de Software da OTAN).

Percebeu-se que é mais barato planejar do que corrigir.

Desenvolvimento de software passou a ser baseado em princípios da engenharia (enfoque estruturado e metodológico).

A partir de então começou-se a usar o termo engenharia de software.

Unidade 1 - Introdução

Perguntas que a Engenharia de Software quer responder.

- ✓ Porque demora tanto para concluir um projeto (não cumprimos prazos)?
- ✓ Porque custa tanto (uma ordem de magnitude a mais)?
- ✓ Porque não descobrimos os erros antes de entregar o software ao cliente?
- ✓ Porque temos dificuldade de medir o progresso enquanto o software está sendo desenvolvido?

Unidade 1 - Introdução

Surgimento da Engenharia de Software

- ✓ Histórico de sistemas desde a década de 50.
- ✓ Desenvolvimento artesanal – natural.
- ✓ Foco era o hardware – pouca importância no software.
- ✓ Problemas de qualidade.
- ✓ Falta de documentação.
- ✓ Ausência de método e processo.
- ✓ Falta de controle de qualidade.
- ✓ Falta de ciclo de vida.

Unidade 1 - Introdução

Surgimento da Engenharia de Software

- ✓ Prazos não eram cumpridos.
- ✓ Recursos extrapolavam a estimativa.
- ✓ A qualidade do software era questionável.
- ✓ O produto final na maioria das vezes não refletia o que foi pedido.
- ✓ Difícil e constante manutenção.
- ✓ Não era possível conhecer os efeitos de uma mudança.

Unidade 1 - Introdução

Mitos do software - administrativo.

- ✓ Um manual oferece tudo que se precisa saber.
- ✓ Computadores de última geração solucionam problemas de desenvolvimento.
- ✓ Se estamos atrasados, basta adicionar programadores e tirar o atraso.

Unidade 1 - Introdução

Mitos do software – do cliente.

- ✓ Uma declaração geral é suficiente para começar a escrever programas.
- ✓ Mudanças podem ser facilmente acomodadas em um projeto.

Unidade 1 - Introdução

Mitos do software – do profissional.

- ✓ Um programa está terminado ao funcionar.
- ✓ Quanto mais cedo escrever o código, mais rápido terminarei o programa.
- ✓ Só posso avaliar a qualidade de um programa em funcionamento.
- ✓ A única coisa a ser entregue em um projeto é o programa funcionando.

Bibliografia Básica

PRESSMAN, Roger S. Engenharia de Software. 6. ed. São Paulo: McGraw-Hill, 2010.

SCHACH, Stephen R. Engenharia de Software. 7. ed. Porto Alegre: ArtMed, 2010.

SOMMERVILLE, Ian. **Engenharia de Software.** 8. ed. São Paulo: Pearson Addison Wesley, 2007.

Bibliografia Complementar

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 12207: **Tecnologia de informação - processos de ciclo de vida de software.** Rio de Janeiro: ABNT, 1998.

LARMAN, Craig. **Utilizando UML e padrões:** uma introdução à análise e ao projeto orientados a objeto e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

PADUA FILHO, Wilson de Paula. **Engenharia de software.** 3. ed. Rio de Janeiro: LTC, 2008.

PFLEEGER, Shari Lawrence. **Engenharia de software:** teoria e prática. 2. ed. São Paulo: Pearson Education do Brasil, 2007.

PRESSMAN, Roger S; LOWE, David Brian. **Engenharia web.** Rio de Janeiro: LTC, 2009.

Bibliografia Complementar

(1) Engenharia de Software, *Roger Pressman*, Makron Books
(2) Software Engineering: Theory and Practice, *Shari Pfleeger*. Prentice Hall
Sommerville, Addison Wesley, 1989.
(3) Aderência do iRUP à norma NBR ISO/IEC 12207. Alcides Calsavara - PUC-PR, Cristina Ângela Filipack Machado - GPT, Sheila dos Santos Reinehr - BANESTADO, Robert Carlisle Burnett - PUC-PR
http://www.pr.gov.br/batebyte/edicoes/2000/bb104/software.htm
(4) A Rational development process: white paper.
http://www.rational.com/products/ rup/prodinfo/whitepapers/
(5) Material dos Profs.:
- Edson Murakami - UPIS - SI - 2004
- Reinaldo Bianchi - FEI – Elétrica, 2000
- Alfredo Lanari-UCBD
http://www.ec.ucdb.br/~alanari/engsoft/engweb03.ppt
- Hermano Perrelli - UFPE -2003 Centro de Informática
http://www.cin.ufpe.br/~if683/slides/visao-geral-rup.ppt
- Disciplina: Estudo do RUP. Autor: Sérgio C. B. Soares, Orientação: Augusto Sampaio, Paulo Borba, iRUP - DI / UFPE - 1999

Bibliografia Complementar

- Curso de Pós-Graduação em Tecnologias da Informação. Prof.a Letícia C. P. Fendt - CEULJI - 2003/1
http://inf.ulbrajp.com.br/~leticia/pos/1AnaliseProjeto.ppt

(6) Sena Informática
http://www.sena.com.br/portal/page?_pageid=40,38142,40_38144&_dad=portal&_schema=PORTAL

(7) Java com Tendência no Desenvolvimento de Software. SoftExport/IBM NATI
http://www.unifor.br/hp/nati/docs/java.ppt

(8) http://www.paradygma.com.br/comp/artigos/rup.pdf

Glossário

- Baselines:Conjuntos de artefatos de software com versões compatíveis.
- Artefatos:São todos os modelos, códigos, documentos e requisitos.
- Worker:Quem executa as atividades dentro do processo de desenvolvimento de software.
- Milestones: São marcos de referência nas quatro fases do RUP.
- Stakeholdes: Pessoas envolvidas no processo de desenvolvimento de software.
- Processo: Um processo define quem está fazendo o quê, quando e como alcançar um determinado objetivo.
- Workflow: Agrupam atividades Logicamente.
- Workflow de Processo: Modelagem de Negócio, Requisitos, Análise e Projeto.
- Workflow de Suporte: Gerenciamento de Configuração, Gerenciamento de Mudanças, Gerenciamento de Projeto