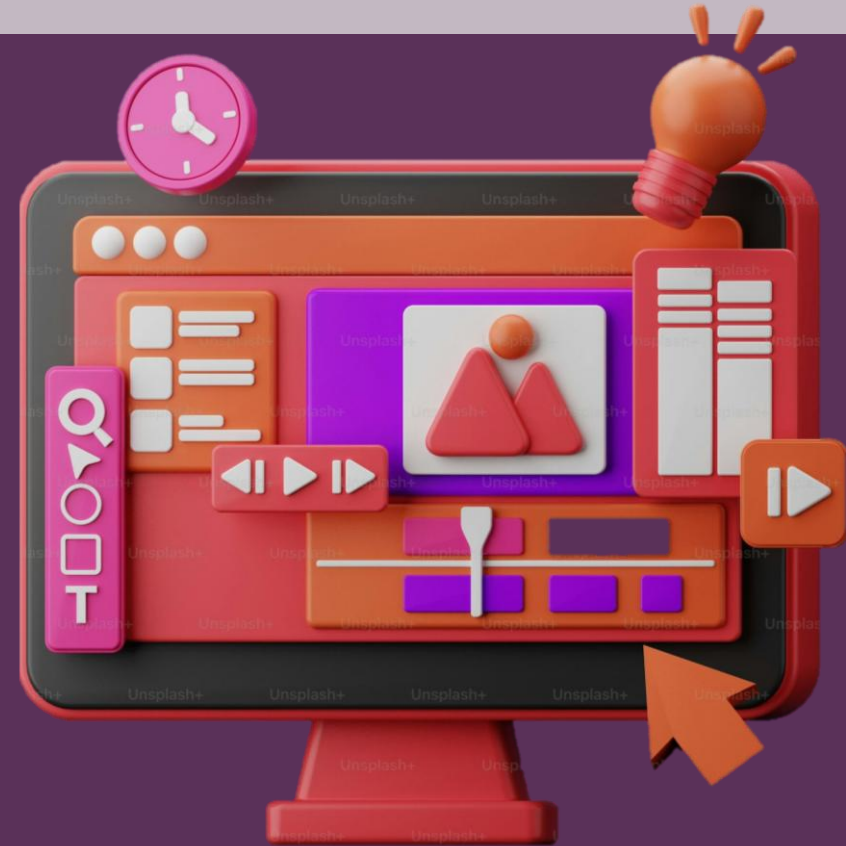


ENGENHARIA DE SOFTWARE

AULA 1 - APRESENTAÇÃO DA DISCIPLINA



WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR



OLÁ!
EU SOU WALTER TRAVASSOS

Entusiasta de novas tecnologias e metodologias inovadoras de ensino aprendizagem.

Você pode me encontrar:

- Instagram: @prof.waltertravassos
- E-mail: travassoswalter@gmail.com
- WhatsApp: (83) 98882-0007

♦ OLÁ! MEU NOME É ♦
**WALTER
TRAVASSOS**



Psicólogo Clínico

Abordagem Centrada na Pessoa – ACP

Eye Movement Desensitization and Reprocessing - EMDR

CRP 13/10.495

Você pode me encontrar:

- Instagram: **@psi.waltertravassos**
- WhatsApp: **(83) 98882-0007**

Mini Currículo

- Graduado em Sistemas para Internet (IFPB - 2012)
 - Mestre em Ciência da Computação (UFPE - 2014)
 - Especialista em Tecnologias e Educação a Distância (Unidoctum – 2020)
 - Graduado em Psicologia (Unipê - 2021)
 - Especialista em Psicologia Comportamental e Cognitiva (Dom Alberto – 2021)
 - Especialista em Tratamento dos Transtornos de Ansiedade e Síndrome do Pânico (Facuminas – 2021)
 - Formação na Abordagem Centrada na Pessoa (ACPBR – 2022), Focalização (Focalização Brasil - 2023) e EMDR (Trauma Clinic - 2023).
 - Especialista em Psicologia Humanista: Abordagem Centrada na Pessoa (Facuminas – 2023)
-
- Professor de cursos técnicos desde 2007.
 - Na graduação desde 2012.
 - **30 disciplinas** diferentes na graduação.
 - Professor de alguns módulos na pós-graduação.

QUAIS SÃO SUAS
EXPECTATIVAS
PARA ESTA
DISCIPLINA?



Conteúdo Programático

Ementa

Princípios da Engenharia de Software. Modelos de ciclo de vida. Processos de software. Requisitos de software. Padrões de projetos de software. Projeto da interface com o usuário. Planejamento e Estimativas. Documentação. Qualidade e métricas. Gerenciamento de mudanças e configurações. Manutenção e evolução de software. Engenharia de Software para sistemas Web. Tópicos atuais de Engenharia de Software. A Pesquisa e o futuro da Engenharia de Software.

UNIDADE – I

DATA	HORÁRIO	LOCAL	TEMA	RECURSO(S) DIDÁTICO(S)	DOCENTE(S)
Terça 12/08/25	07h50min 11h40min	Sala de Aula	0. Apresentação da disciplina e cronograma de aulas. Acolhida discente.	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 19/08/25	07h50min 11h40min	Sala de Aula	1. Introdução à Engenharia de Software	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 26/08/25	07h50min 11h40min	Sala de Aula	2. Modelos de Ciclo de Vida do Software	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 02/09/25	07h50min 11h40min	Sala de Aula	3. Processos de Software	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 09/09/25	07h50min 11h40min	Sala de Aula	4. Requisitos de Software	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 23/09/25	07h50min 11h40min	Sala de Aula	5. Documentação de Software (Definição de Escopo e Especificação de Requisitos)	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 30/09/25	07h50min 11h40min	Sala de Aula	1ª Verificação de aprendizagem	Prova impressa	Walter Sarinho

UNIDADE – II

Terça 07/10/25	07h50min 11h40min	Sala de Aula	6. Atividade Prática e Curricularização da Extensão	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 14/10/25	07h50min 11h40min	Sala de Aula	7. Planejamento e Estimativas de Projetos	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 21/10/25	07h50min 11h40min	Sala de Aula	8. Gerenciamento de Riscos	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 28/10/25	07h50min 11h40min	Sala de Aula	9. Gerenciamento de Pessoas	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 04/11/25	07h50min 11h40min	Sala de Aula	10. Projeto da Interface com o Usuário	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 11/11/25	07h50min 11h40min	Sala de Aula	2ª Verificação de aprendizagem	Prova impressa	Walter Sarinho

UNIDADE – III

Terça 18/11/25	07h50min 11h40min	Sala de Aula	11. Sustentabilidade do Software (Gerenciamento de Mudanças e Configurações; e Manutenção e Evolução de Software)	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 25/11/25	07h50min 11h40min	Sala de Aula	12. Qualidade e Métricas de Software. Engenharia de Software para Sistemas Web. Tópicos Atuais e Futuro da Engenharia de Software. 13. Padrões de Projetos de Software	Projektor Multimídia, quadro branco, pincel, simulações.	Walter Sarinho
Terça 02/12/25	07h50min 11h40min	Sala de Aula	3ª Verificação de aprendizagem	Prova impressa	Walter Sarinho
Terça 09/12/25	07h50min 11h40min	Sala de Aula	Reposição	Prova impressa	Walter Sarinho
Sexta 12/12/25	07h50min 11h40min	Sala de Aula	EXAME FINAL	Prova impressa	Walter Sarinho

AVALIAÇÃO



A Avaliação é Somativa...

1ª Unidade (0,00 a 10,00):

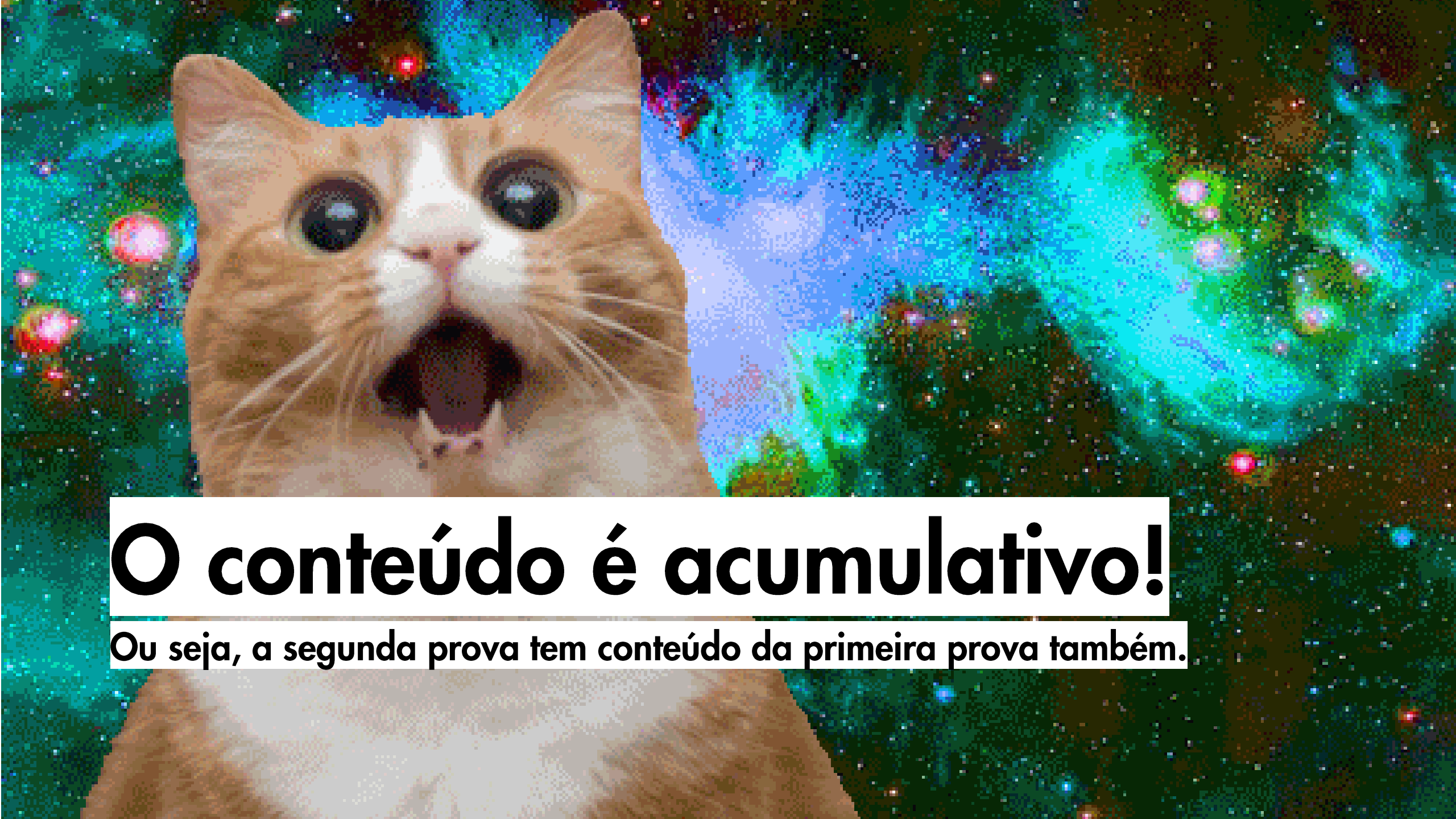
- Avaliação Escrita (8,00) e Atividade em sala (2,00).

2ª Unidade (0,00 a 10,00):

- Avaliação Escrita (8,00) e Atividade em sala (2,00).

3ª Unidade (0,00 a 10,00):

- Avaliação Escrita (6,00), Atividade em sala (2,00) e Prática Integradora (2,00).



O conteúdo é acumulativo!

Ou seja, a segunda prova tem conteúdo da primeira prova também.

Nota Final Mínima:
7,00

Recomendações para assistir as aulas remotas ou presenciais

- Apenas assista a aula (processo psicológico da atenção).
- Esteja presente e participe.
- Seja cordial na exposição de suas ideias.



Bibliografia

<https://plataforma.bvirtual.com.br/>



Sommerville, Ian.
Engenharia de Software.
10. ed. Pearson. 2011.

Bora começar?

Alinhar algumas questões sobre Software

- O que é um software?
- O que um software pode fazer?
- Que tipo de problemas um software pode resolver?
- Que problemas ele não consegue resolver? Por quê?
- É difícil criar um software? Por quê?

Introdução a Engenharia de Software

A Engenharia de Software...

É uma área da engenharia e da computação voltada à especificação, desenvolvimento, manutenção e criação de **software**, com a aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade.

Práticas e Tecnologias

- Atualmente, essas tecnologias e práticas englobam linguagens de programação, banco de dados, ferramentas, plataformas, bibliotecas, padrões de projeto de software, processo de software e qualidade de software.
- Além disso, a engenharia de software deve **oferecer mecanismos para se planejar e gerenciar o processo de desenvolvimento de um sistema computacional de qualidade** e que atenda às necessidades de um requisitante de software.

Utilizamos modelos abstratos

Os fundamentos científicos para a engenharia de software envolvem o uso de **modelos abstratos** e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades.

No entanto, a maior dificuldade para produzir um sistema computacional é que...

**Software é
produto
intangível**



A black and white photograph of an early computer room. Several people are working on large, dark-colored cabinets that house the computer's components. The room is filled with these cabinets, and the people are dressed in mid-20th-century attire. The scene illustrates the physical nature of early computing technology.

**No início...
O desenvolvimento
de softwares era
muito obsoleto.**

No início...

- Faltava eficiência, ou seja, desenvolver em menos tempo, gastando menos.
- É nesse cenário que surgem os Processos de Desenvolvimento de Softwares.

- Mas o que são esses processos?
- Para que servem?



Processos de Desenvolvimento de Software (PDS)

- São modelos que abordam como desenvolver software de forma eficiente em diferentes formatos.
- Os modelos vão possibilitar mensurar e comparar resultados (inclusive entre os modelos).
- Assim, um processo de desenvolvimento de software é um conjunto de atividades, parcialmente ordenadas, com a finalidade de obter um produto de software.

Processos de Desenvolvimento de Software (PDS)

- O PDS é estudado dentro da área de Engenharia de Software.
- É considerado um dos principais mecanismos para se obter software de qualidade e cumprir corretamente os contratos de desenvolvimento.
- É uma das respostas técnicas mais adequadas para resolver a **Crise do Software**.

É normal não saber fazer algo tão bem feito quando estamos fazendo pela 1ª vez



- As pontes romanas da antiguidade eram estruturas muito ineficientes.
- Para os padrões modernos, elas usavam muita pedra, e como resultado, despendiam muito trabalho para serem construídas.
- Através dos anos, nós aprendemos a construir pontes mais eficientemente, usando pouco material e menos trabalho para realizar a mesma tarefa.

A crise do software

- Foi um termo cunhado para descrever as dificuldades enfrentadas no desenvolvimento de software no fim da década de 60.
- A complexidade dos problemas, a ausência de técnicas bem estabelecidas e a crescente demanda por novas aplicações começavam a se tornar um problema sério.
- Foi nessa época, mais precisamente em 1968, que ocorreu a Conferência da OTAN sobre Engenharia de Software (NATO Software Engineering Conference) em Garmisch, Alemanha.
- O principal objetivo dessa reunião era estabelecer práticas mais maduras para o processo de desenvolvimento, por essa razão o encontro é considerado hoje como o nascimento da disciplina de Engenharia de Software.



**O que é um
software bom?**



QUALIDADE DE SOFTWARE

<https://youtu.be/n8sAGdxmsaQ>

Gerência e Qualidade de Software

https://youtu.be/XiG_Gz-sv48



Não há Bala de Prata para o Desenvolvimento de Software

<https://iopub.org/n%C3%A3o-h%C3%A1-bala-de-prata-para-o-desenvolvimento-de-software-5c6e73ecd28c>

<http://worrydream.com/refs/Brooks-NoSilverBullet.pdf>

No Silver Bullet

- Em 1986 foi publicado um dos grandes clássicos da Engenharia de Software, o artigo “No Silver Bullet — Essence and Accident in Software Engineering” (Não há bala de prata — Essência e Acidente na Engenharia de Software) por Frederick P. Brooks
- Não há nenhuma metodologia individual de desenvolvimento, em tecnologia ou técnicas de gestão, que por si só prometa um incremento ainda que de uma ordem de grandeza dentro de uma década em produtividade, em confiabilidade, em simplicidade”.
- Ou seja, não há uma bala de prata que sozinha resolva os problemas do Software.

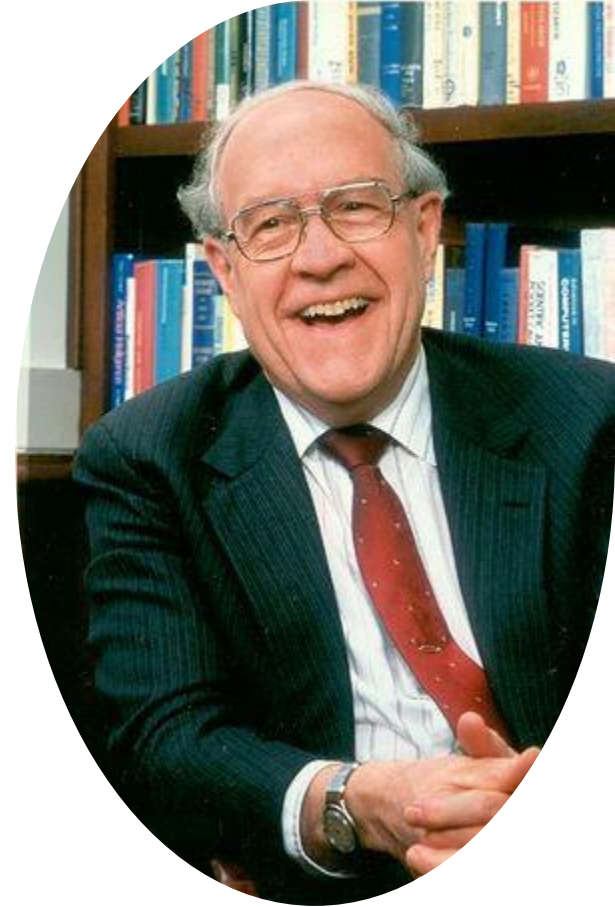
Zeitgeist

- É um termo alemão cuja tradução significa espírito da época, espírito do tempo ou sinal dos tempos.
- Significa, em suma, o conjunto do clima intelectual e cultural do mundo, numa certa época, ou as características genéricas de um determinado período de tempo.
- Na época do artigo, o filme Um Lobisomem Americano em Londres já era um sucesso em bilheteria. Se fosse nos dias de hoje, talvez uma alternativa para a bala de prata fosse a bala do Colt da série Supernatural.

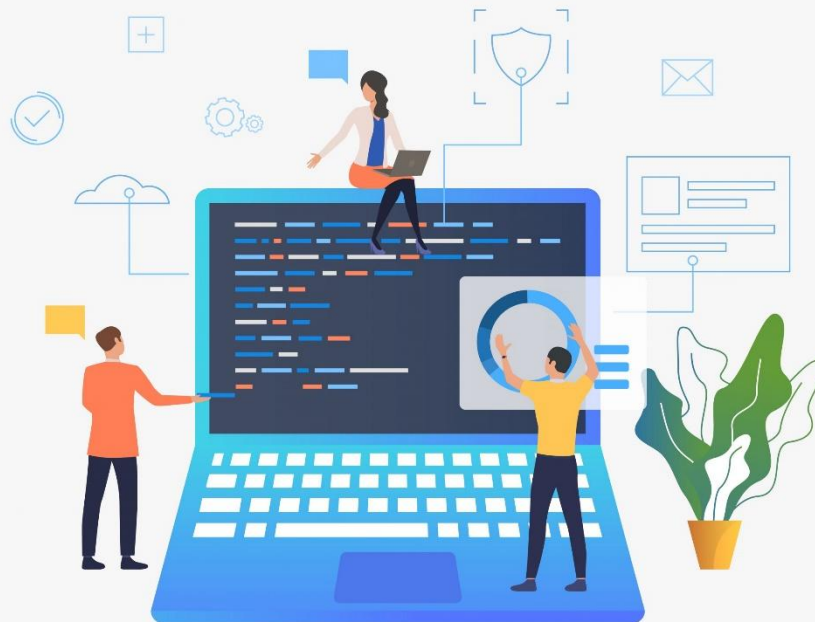


No Silver Bullet

- Neste artigo, o autor Fred Brooks faz uma brilhante — e também estranha — analogia entre o Software e o Lobisomem, alegando que estes seres se transformam, inesperadamente, de coisas familiares à seres horripilantes.
- A diferença é que para o Lobisomem, basta uma bala de prata para se resolver o problema.
- Com relação ao Software, os horrores seriam os prazos perdidos, orçamentos estourados e produtos com falhas, só que diferentemente do Lobisomem, não há uma única bala de prata que resolva esses problemas.



Os problemas essenciais do software dizem respeito as suas próprias características inerentes.



1. Complexidade

- No desenvolvimento Software **não existem dois elementos repetidos ou idênticos**, quando existem eles são transformados em uma coisa só na forma de funções.
- Essa característica é um grande desafio na hora de **escalar** o Software para desafios maiores, uma vez que não basta o uso dos mesmos componentes em tamanho maior, como muitas vezes acontece com o Hardware, construções, automóveis e outras máquinas, onde os elementos repetidos superabundam.
- Dessa forma, não existe um crescimento linear para o Software.



-FX-
AHS
DOUBLE FEATURE
8.25

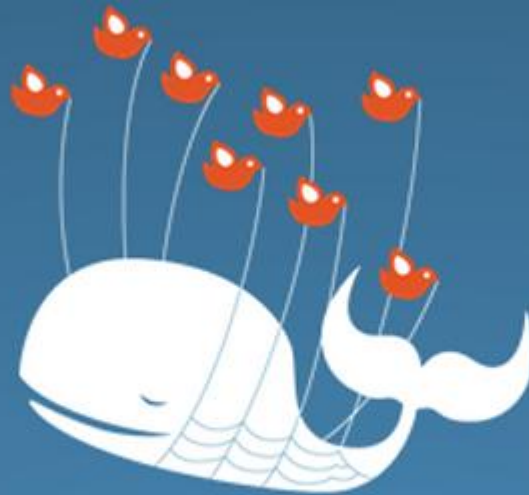
**1 mulher pode fazer
1 filho em 9 meses,
no entanto...**

**9 mulheres não conseguem fazer 1
filho em 1 mês.**

2. Conformidade

- Não é só o Software que sofre com os problemas de complexidade, a física por exemplo também sofre, no entanto, diferentemente da física, o Software **não possui leis imutáveis as quais se possa agarrar.**
- O Software **precisa se adaptar a todo tipo de interface, sistema ou instituição existente**, e se não bastasse, estes supostos padrões mudam de tempos em tempos de acordo com a moda do momento.
- Como diz o autor, as vezes não é nem por necessidade, mas simplesmente porque os padrões são desenvolvidos por pessoas e não por Deus, como no caso da física e as leis da natureza.

**Lembram da baleia do Twitter?
Ela impulsionou o uso de SGBDs
NoSQL para sistemas Web.**



Twitter is over capacity.

3. Alterabilidade

- O Software é constantemente sujeito a **pressões por mudança**. Isso se deve a sua característica abstrata que lhe confere a impressão de que mudanças são fáceis de se fazer, muito diferente daquilo que ocorre com os produtos manufaturados do mundo físico.
- Para tal, basta se comparar a quantidade de recalls que acontecem no setor automobilístico com a quantidade de atualizações na indústria de Software.
- Esta característica é em parte benéfica por permitir uma “rápida” evolução dos sistemas, entretanto, paga-se o preço para tal.



- Ansiedade e transtornos mentais relacionados ao trabalho vem se tornando um desafio entre as empresas de tecnologia.
- De acordo com o International Journal of Social Sciences*, os desenvolvedores de software têm uma chance consideravelmente maior de experimentar fadiga, esgotamento, ansiedade, e estresse, em comparação aos colegas que realizam tarefas mecânicas.
- Deteriorar a saúde mental não apenas ameaça o bem-estar dos funcionários, mas também a produtividade geral.

*[http://www.ijhssi.org/papers/v3\(2\)/Version-2/G0322052056.pdf](http://www.ijhssi.org/papers/v3(2)/Version-2/G0322052056.pdf)

4. Invisibilidade

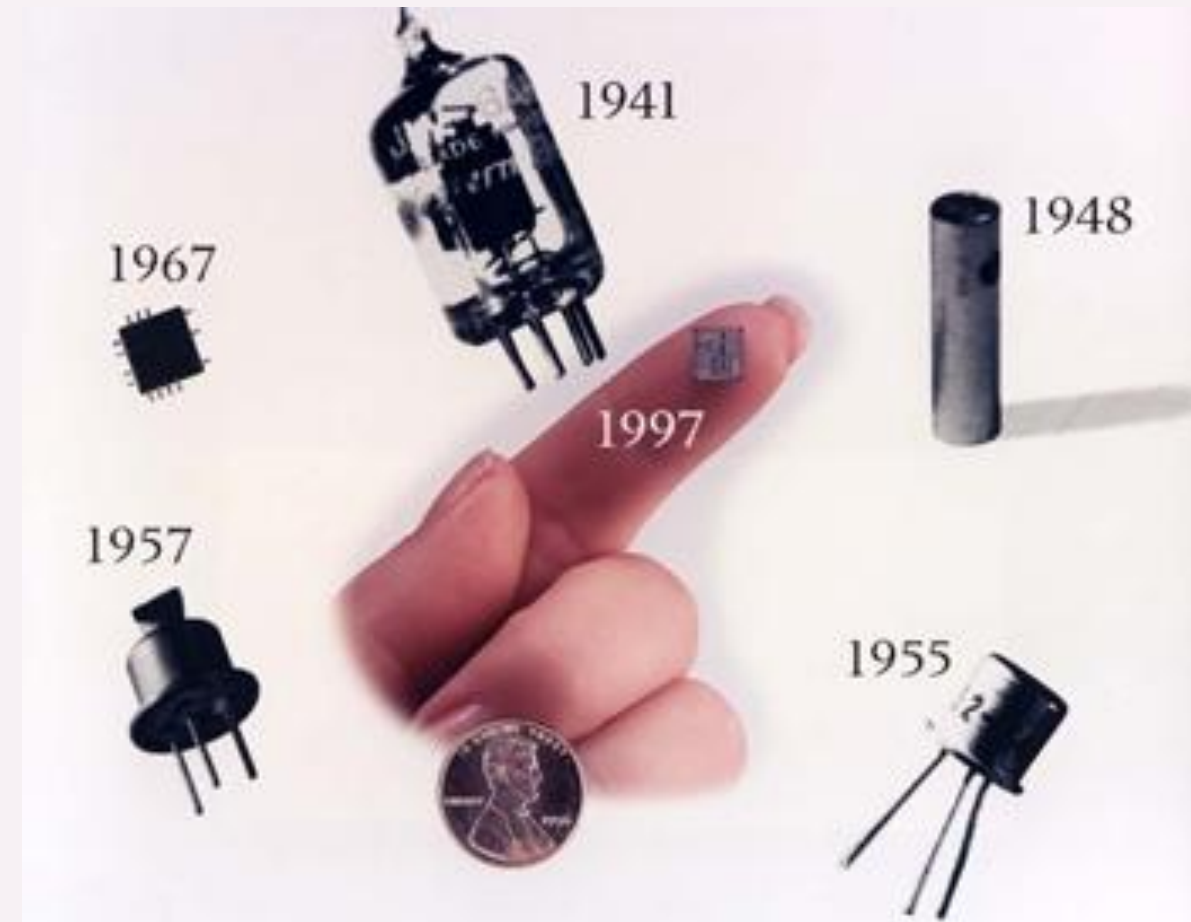
- Essa é talvez a característica mais intuitiva, acredito que todos já ouviram aquela famosa expressão de que o Software é aquilo que se xinga.
- Pois é, o Software é de fato invisível, não é possível representá-lo a partir de formas geométricas ou outras formas compreendidas pelo senso comum.
- Esta característica representa um grande desafio para o projeto de sistemas e a sua representação para terceiros.

Software, em geral,
não tem sentimentos.
Não adianta xingar!



Por fim...

Considerando-se todas essas características de **forte cunho abstrato**, não é difícil de se concluir que **difícilmente existirá uma única solução** que alavanque a produtividade, a confiabilidade e a simplicidade do **Software**, assim como foram os transistores para a eletrônica, por exemplo.



Lista de Exercícios 1

Algumas questões de concurso

Ano: 2025 Banca: CESPE / CEBRASPE Órgão: TRF - 6ª REGIÃO Prova: CESPE / CEBRASPE - 2025 - TRF - 6ª REGIÃO - Analista Judiciário – Área: Apoio Especializado – Especialidade: Análise de Sistemas de Informação

Julgue o item a seguir, no que se refere à engenharia de *software* e à análise de requisitos.

Para garantir a manutenibilidade, a arquitetura do sistema deve ser projetada para adotar componentes grandes e centralizados, facilitando futuras modificações e atualizações.

 Certo

Errado

Em um projeto de desenvolvimento de software para um cliente do setor público, o analista enfrenta dificuldades para entender os requisitos funcionais e não funcionais dos usuários. Qual abordagem seria mais eficaz para identificar e esclarecer os requisitos, minimizando o risco de retrabalho e desalinhamento com as expectativas?

Solicitar ao cliente um documento detalhado com todos os requisitos esperados para o sistema.

Revisar documentações de sistemas similares para levantar requisitos com base em comparações.

Realizar workshops de elicitação com as principais partes interessadas para clarificar e priorizar os requisitos.

Delegar a definição de requisitos para o gerente de projeto, que possui um melhor entendimento do contexto organizacional.

Um projeto de desenvolvimento de software está atrasado e com o orçamento ultrapassado. Após uma análise, o gerente de projetos identificou que a causa principal dos problemas é a falta de padronização e controle nos processos de desenvolvimento, o que leva a retrabalhos, erros e dificuldades na comunicação entre a equipe. Considerando os conceitos de gerenciamento de projetos e gestão de processos, qual seria a ação mais eficaz para solucionar os problemas e melhorar o desempenho do projeto?

Cancelar o projeto e reiniciá-lo do zero, utilizando uma metodologia de desenvolvimento mais adequada e uma equipe mais qualificada.

Mapear, analisar e otimizar os processos de desenvolvimento, definindo padrões, responsabilidades e mecanismos de controle para evitar retrabalhos e erros.

Concentrar esforços na etapa de execução do projeto, aumentando a quantidade de horas trabalhadas pela equipe para recuperar o tempo perdido.

Renegociar o prazo e o orçamento do projeto com o cliente, justificando os problemas e solicitando mais tempo e recursos para finalizar o projeto.

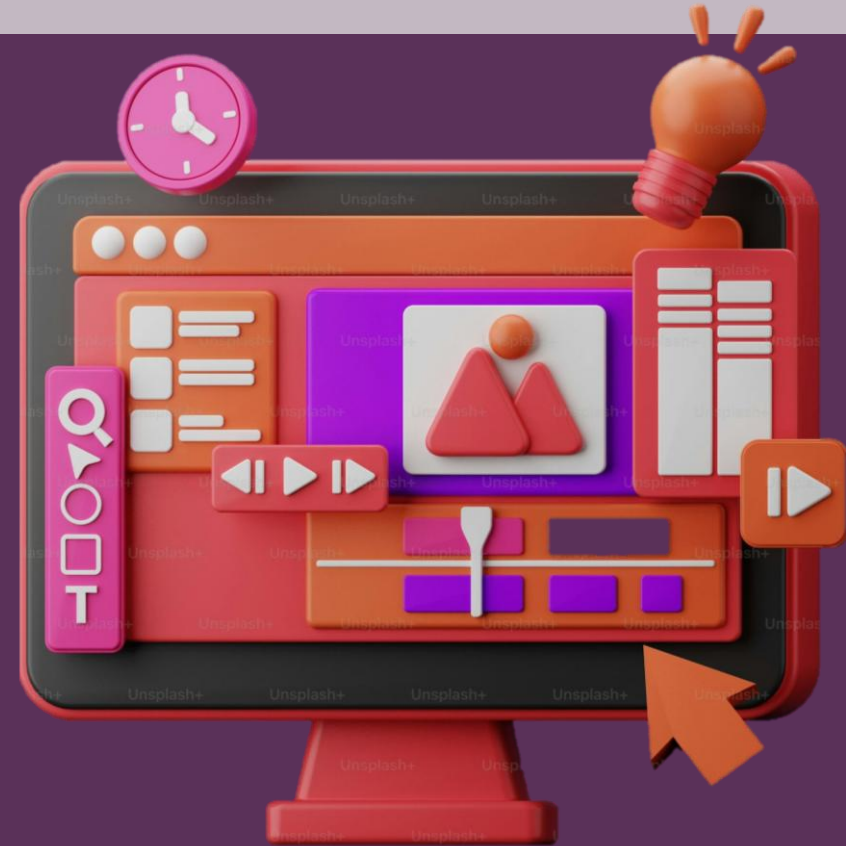
Substituir a equipe de desenvolvimento por profissionais mais experientes, capazes de entregar o projeto com maior rapidez e qualidade.

Palavras Cruzadas

https://www.educolorir.com/crosswords/engenharia_de_software-5b30962f0b6b6489f0a666989999041f

ENGENHARIA DE SOFTWARE

AULA 1 - APRESENTAÇÃO DA DISCIPLINA



WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR