

Fundamentos de Eng. Software



2023

Fundamentos de Eng. Software

O que é Engenharia?

Fundamentos de Eng. Software

“Engenheiro” deriva do latim *in generare*, que significa a faculdade de saber, criatividade.

Fundamentos de Eng. Software

“Engenheiro” data de cerca de 200 a.C. Autor Tertuliano descreve um ataque romano a Catargo com uso de aríate, descrito como *ingenium*, uma invenção engenhosa. Pessoas que desenvolviam engenhos inovadores eram conhecidas como *ingeniator*.

Fundamentos de Eng. Software

No Brasil

- * Engenheiros estrangeiros são trazidos para ensinar técnicas de fortificação e artilharia para o exército colonial
- * Em 1792 é fundada a Real Academia de Artilharia, Fortificação e Desenho
- * Em 1810 por ordem do Príncipe D. João, a Real Academia é substituída pela Academia Real Militar.
- * Da Academia Real Militar descende a Escola Polytechnica do Rio de Janeiro, que atualmente, Escola Politécnica da UFRJ.

Fundamentos de Eng. Software

Engenharia como disciplina acadêmica surge no século XVIII, com caráter militar.

Fundamentos de Eng. Software

Engenharia é a **aplicação do conhecimento científico, econômico, social e prático**, com o intuito de **inventar, desenhar, construir, manter e melhorar estruturas, máquinas, aparelhos, sistemas, materiais e processos**. É também profissão em que **se adquire e se aplicam os conhecimentos matemáticos e técnicos** na criação, aperfeiçoamento e implementação de utilidades que realizem uma função ou objetivo.

Fundamentos de Eng. Software

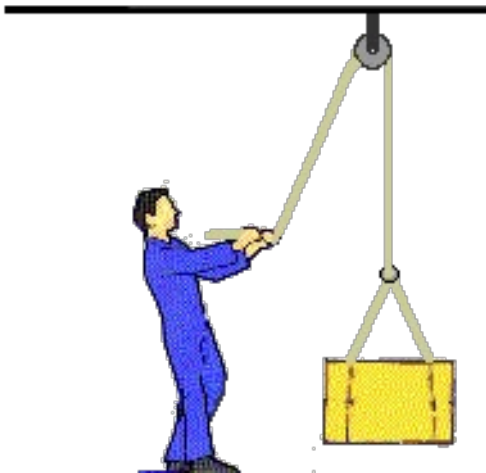
Nos processos de criação, aperfeiçoamento e complementação, a engenharia conjuga os vários conhecimentos especializados no sentido de viabilizar as utilidades, **tendo em conta a sociedade, a técnica, a economia e o meio ambiente.**

Fundamentos de Eng. Software

Exemplos de obras antigas que se encaixem nos conceitos modernos de engenharia?

Fundamentos de Eng. Software

Exemplos de obras antigas que se encaixem nos conceitos modernos de engenharia?



Fundamentos de Eng. Software

Comitê de Certificação de Engenharia e Tecnologia dos Estados Unidos (1982)

Engenharia é a profissão na qual o conhecimento das ciências matemáticas e naturais, obtido através do estudo, experiência e prática, é aplicado com julgamento no desenvolvimento de novos meios de utilizar, economicamente, os materiais e forças da Natureza para o benefício da humanidade.

Fundamentos de Eng. Software

Engenharia de Software:

- Termo utilizado pela primeira vez em 68, pelo prof. Friederich Bauer na primeira conferência dedicada ao assunto pelo NATO Science Comitee para discutir a “Crise do Software”.
- Houve o entendimento de que software é um produto que precisava de processos de produção.
- No Brasil Eng. Soft. Sempre foi uma disciplina. A área não era consolidada nem nos EUA.
- Consolidação a partir de 2016 (ACM) e no Brasil, aprovação pelo CNE em 2012 e homologada em 2016

Fundamentos de Eng. Software

Engenharia de Software:

- Engenharia de Software é uma disciplina de engenharia relacionada a todos os aspectos, desde os estágios iniciais de especificação até a sua manutenção depois que este entrar em operação.
- Desafios da Engenharia de Software – Estar à altura do aumento da diversidade, demandas para redução de tempo e desenvolvimento de software digno de confiança.

Fundamentos de Eng. Software

O que é um software?

Software segundo a disciplina:

Programas de computador e documentação associada.

Fundamentos de Eng. Software

O que é um sistema computacional?

- Software
- Hardware
- Pessoas
- Banco de Dados
- Documentação
- Procedimentos

Fundamentos de Eng. Software

Objetivos da Engenharia de Software?

- Por que o software demora tanto para ser concluído?
- Por que os custos de produção são tão elevados?
- Por que não é possível detectar todos os erros antes que o software seja entregue ao cliente?
- Por que é tão difícil medir o progresso durante o processo de desenvolvimento de software?

Fundamentos de Eng. Software

Eng. de Software x Ciência da Computação

- C.C. – se relaciona a teorias e fundamentos.
- E. S. - está relacionada com a prática de desenvolvimento e entrega do software útil.

Fundamentos de Eng. Software

Eng. de Software x Eng. De Sistemas

- E.Sist. – se relaciona a todos os aspectos de desenvolvimento de sistemas computacionais, incluindo hardware, software e engenharia de processos.
- E. Soft. - é considerada parte deste processo.

Fundamentos de Eng. Software

Engenheiro de Software é um profissional com sólidos conhecimentos em:

- Ciência da Computação
- Matemática
- Produção

Fundamentos de Eng. Software

Histórico de Desenvolvimento de Software

Década de 50

Fortran e Cobol

Década de 60

Grande demanda

Crise do software

Décadas seguintes

evolução das linguagens e paradigmas

Fundamentos de Eng. Software

Alguns conceitos:

- Processo de software: Conjunto de atividades com objetivo é o desenvolvimento e evolução do software.
- Modelo de processo de software – Representação simplificada de um processo de software.
- Custos da engenharia de software – 60% desenvolvimento e 40% de testes. Para softwares sob encomenda, custos de evolução usualmente superam o de desenvolvimento.

Fundamentos de Eng. Software

Alguns conceitos:

- Métodos da engenharia de software - Abordagem estruturada para desenvolvimento incluindo modelos de sistema, notações, regras.
- Bom software - funcionalidade e desempenhos exigidos satisfeitos, fácil de manter, confiável e usável.

Fundamentos de Eng. Software

Importância da correta reutilização de código.

- Atualmente: comum a entrega de com atraso, erros e excedendo orçamentos previstos.
- Requisitos mudam constantemente.
- Reinvenção da roda: kit de algoritmos não compartilhado entre programadores.

Fundamentos de Eng. Software

Componentes cruciais para o sucesso da Engenharia de Software Clássica.

- Comunicação
- Testes
- Manutenção

Engenharia de Software

Comunicação

- Complexidade e custos de comunicações de um projeto aumentam com o quadrado do número de desenvolvedores enquanto o trabalho aumenta linearmente.

Fundamentos de Eng. Software

Manutenção

- Custo total de manutenção de um software largamente utilizado é tipicamente 40% ou mais do custo de desenvolvimento.

Fundamentos de Eng. Software

Testes

- Uma alteração em qualquer parte do código requer o teste da modificação e de toda parte do código que interage com o código modificado.

Fundamentos de Eng. Software

Gerentes de qualidade ficam loucos:

Quando ocorrem modificações não controladas no código fonte.

Gerentes de projetos ficam loucos:

Quando novos desenvolvedores ganham acesso ao fonte de um projeto em andamento. O custo de tempo é alto. Adaptação a nuances, estilo de código, atrasam o projeto.

Fundamentos de Eng. Software

Exemplo de documento real sobre estilo de desenvolvimento do linux, adotado por diversos projetos.

http://git.kernel.org/?p=linux/kernel/git/torvalds/linux.git;a=blob_plain;f=Documentation/CodingStyle;hb=HEAD

Fundamentos de Eng. Software

Chapter 1: Indentation

Tabs are 8 characters, and thus indentations are also 8 characters. There are heretic movements that try to make indentations 4 (or even 2!) characters deep, and that is akin to trying to define the value of PI to be 3.

Rationale: The whole idea behind indentation is to clearly define where a block of control starts and ends. Especially when you've been looking at your screen for 20 straight hours, you'll find it a lot easier to see how the indentation works if you have large indentations.

Now, some people will claim that having 8-character indentations makes the code move too far to the right, and makes it hard to read on a 80-character terminal screen. The answer to that is that if you need more than 3 levels of indentation, you're screwed anyway, and should fix your program.

In short, 8-char indents make things easier to read, and have the added benefit of warning you when you're nesting your functions too deep. Heed that warning.

Fundamentos de Eng. Software

The preferred way to ease multiple indentation levels in a switch statement is to align the "switch" and its subordinate "case" labels in the same column instead of "double-indenting" the "case" labels. E.g.:

```
switch (suffix) {  
    case 'G':  
    case 'g':  
        mem <<= 30;  
        break;  
    case 'M':  
    case 'm':  
        mem <<= 20;  
        break;  
    case 'K':  
    case 'k':  
        mem <<= 10;  
    /* fall through */  
    default:  
        break;  
}
```

Fundamentos de Eng. Software

Chapter 3: Placing Braces and Spaces

The other issue that always comes up in C styling is the placement of braces. Unlike the indent size, there are few technical reasons to choose one placement strategy over the other, but the preferred way, as shown to us by the prophets Kernighan and Ritchie, is to put the opening brace last on the line, and put the closing brace first, thusly:

```
if (x is true) {  
    we do y  
}
```


Fundamentos de Eng. Software

Chapter 6: Functions

Functions should be short and sweet, and do just one thing. They should fit on one or two screenfuls of text (the ISO/ANSI screen size is 80x24, as we all know), and do one thing and do that well.

```
int system_is_up(void)
{
    return system_state == SYSTEM_RUNNING;
}
```

Fundamentos de Eng. Software

Chapter 8: Commenting

Comments are good, but there is also a danger of over-commenting. NEVER try to explain HOW your code works in a comment: it's much better to write the code so that the `_working_` is obvious, and it's a waste of time to explain badly written code.

The preferred style for long (multi-line) comments is:

```
/*  
 * This is the preferred style for multi-line  
 * comments in the Linux kernel source code.  
 * Please use it consistently.  
 *  
 * Description: A column of asterisks on the left side,  
 * with beginning and ending almost-blank lines.  
 */
```

Fundamentos de Eng. Software

Como garantir a exatidão do programa?

Testando cada recurso. Dependendo do projeto, teste de cada função. Pessoal treinado exclusivamente.

Como era garantida a gerência de projetos?

Não permitir entrada de novos desenvolvedores.

O gerente monitorava o progresso. Dividia o projeto em duas fases: entrega um subconjunto de funcionalidades na data, e na segunda fase o restante.

Fundamentos de Eng. Software

Standish Group

+ 50.000 projetos no mundo

Projetos monitorados desde 1994 na indústria de desenvolvimento de software

Fundamentos de Eng. Software

Standish Group

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Fundamentos de Eng. Software

Standish Group

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.

Fundamentos de Eng. Software

Standish Group

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000