Fundamentos de Desenvolvimento de Software

Ricardo Costa

rac2@cesar.school

C.E.S.A.R school

Professor

Ricardo Costa

- Doutor em Ciência da Computação – UFPE
- Analista do Banco Central
- Engenheiro de Software (desde o ano 2000)
- Contatos:
  - Email: rac2@cesar.school
  - Slack: @Ricardo Costa

A Disciplina - Ementa

- ~~Fundamentos de Desenvolvimento de Software~~
- ENGENHARIA DE SOFTWARE
  - Motivação
  - Introdução a Engenharia de Software
  - Processos de Software
  - Metodologias Ágeis
  - Requisitos
  - Modelagem e Projetos de Software
  - Arquitetura de Software
  - Testes
  - DevOps

Objetivos

- Compreender as etapas do desenvolvimento de software
- Entender as diferenças entre metodologias tradicionais e ágeis.
- Entender os conceitos básicos da engenharia de requisitos.
- Entender conceitos de modelagens de sistemas.
- Compreender os tipos de arquiteturas de sistemas, bem como suas principais aplicações.
- Conhecer alguns dos principais padrões arquiteturais.
- Conhecer conceitos básicos sobre testes de software.
- Aprender como aplicar processos de Integração e Implantação Contínua.

C.E.S.A.R school

Conteúdo Programático

Módulo I

- Motivação
- Introdução a Engenharia de Software
- Processos de Software
- Metodologias Ágeis
- Requisitos

Módulo II

- Modelagem e Projetos de Software
- Arquitetura de Software
- Padrões Arquiteturais
- Testes
- DevOps
- Programador pragmático

Metodologia

- Aulas teóricas para discussão dos conceitos
- Aulas práticas para exercitar conceitos
- Avaliações ao final de cada módulo
- Seminários e/ou palestras

Monólogos ou diálogos?

Depende de vocês!

Avaliação de Aprendizado

Módulos I e II

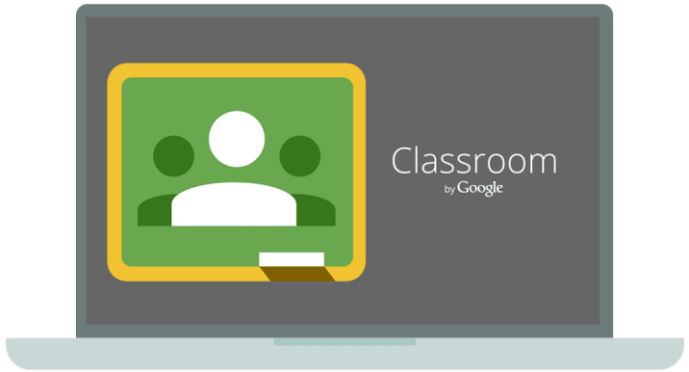
- Nota = A + AP
- A = Avaliações (40%)
- AP = Atividade Prática (60%)

Nota Final

- Nota final = (Nota 1 + Nota 2) / 2

C.E.S.A.R school

Canais oficinais



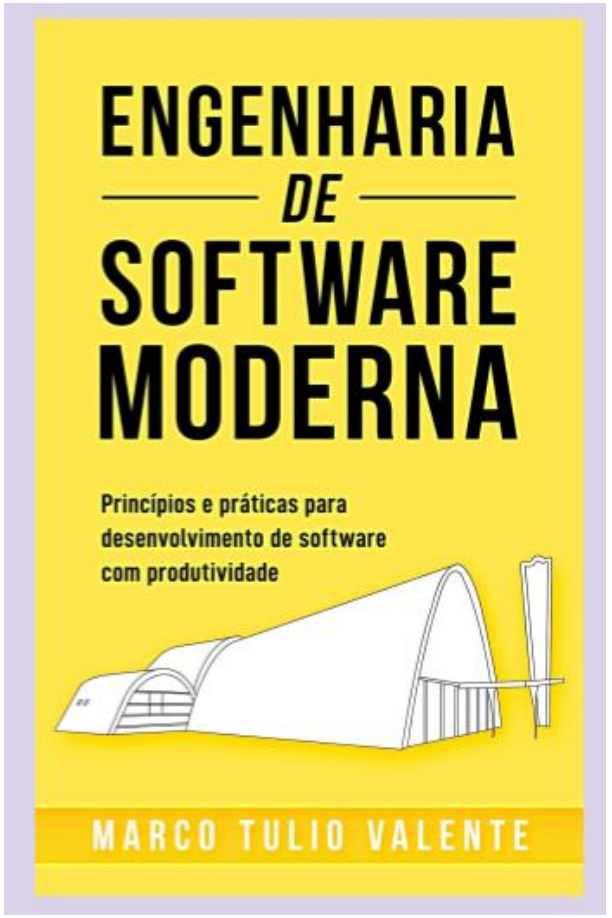
3ltginz (FDS 2024.1 - A)

43qguuy (FDS 2024.1 - B)

#fds-2024-1

C.E.S.A.R school

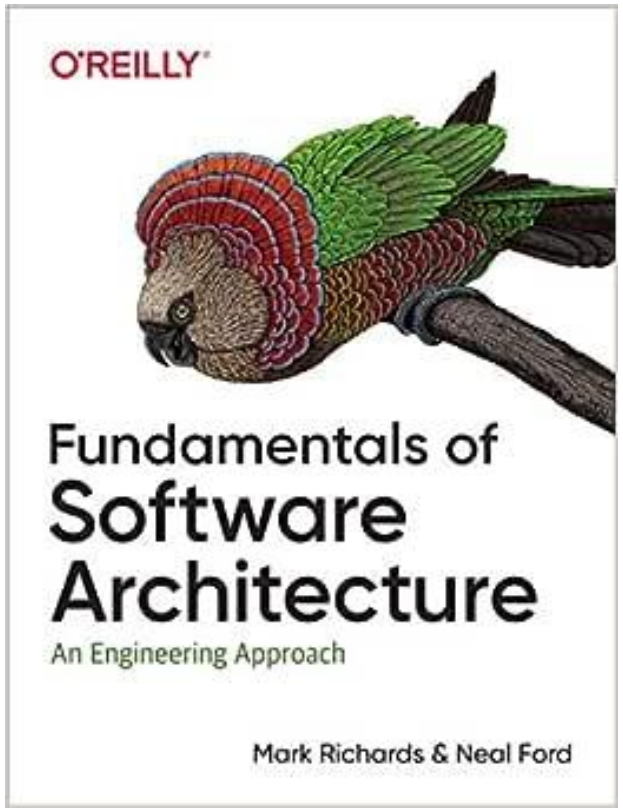
Bibliografia Principal



https://engsoftmoderna.info/

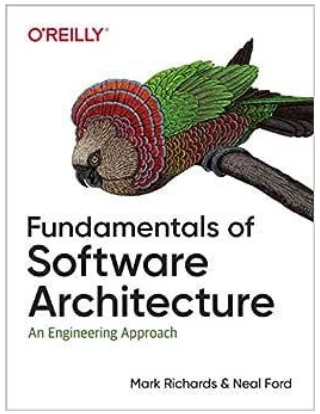
C.E.S.A.R school

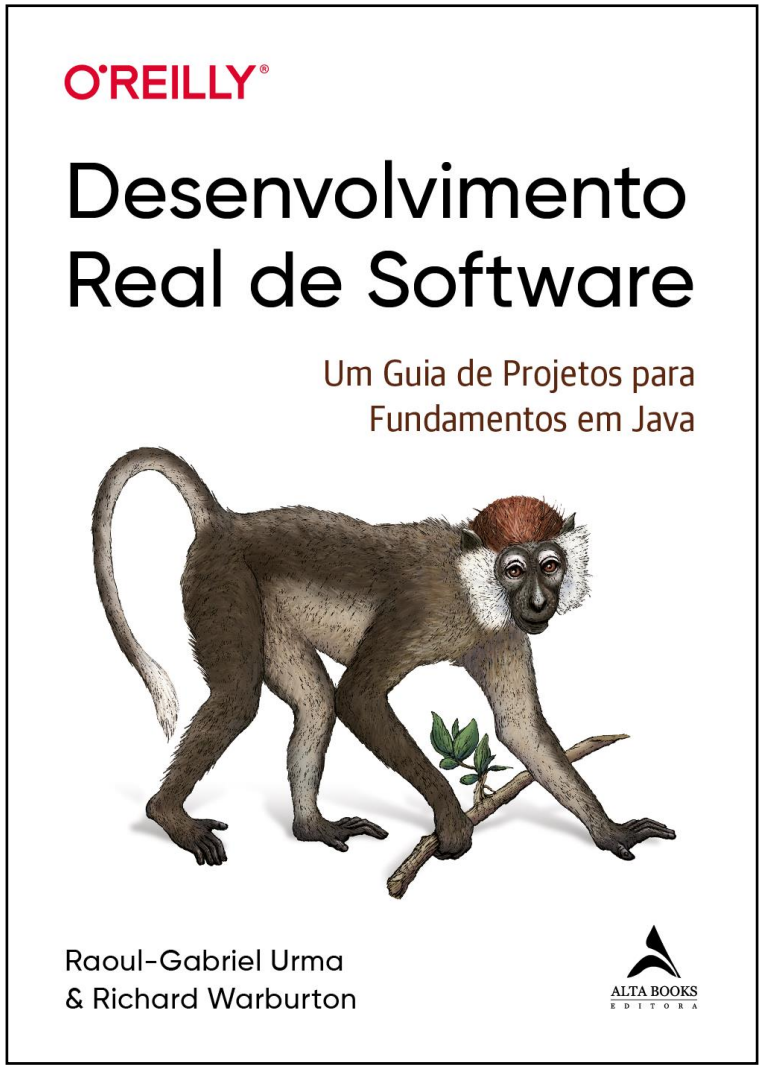
Bibliografia adicional



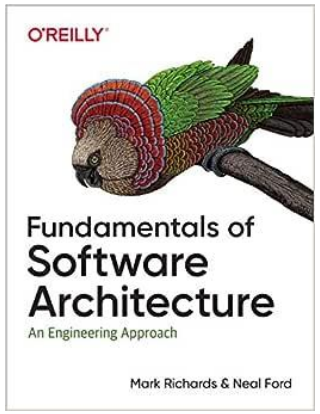
C.E.S.A.R school

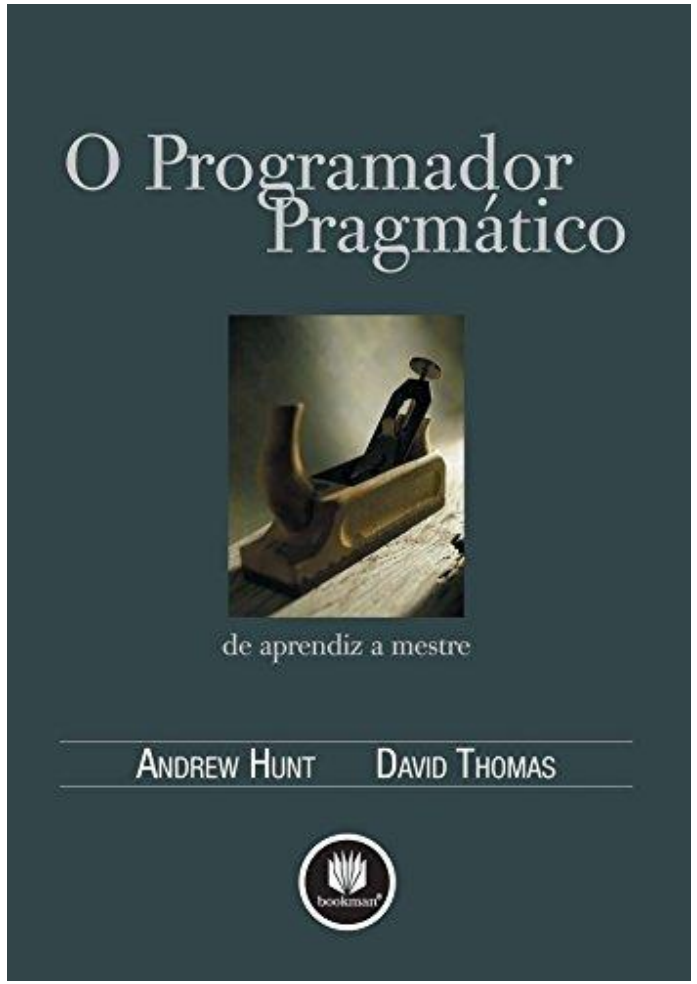
Bibliografia adicional

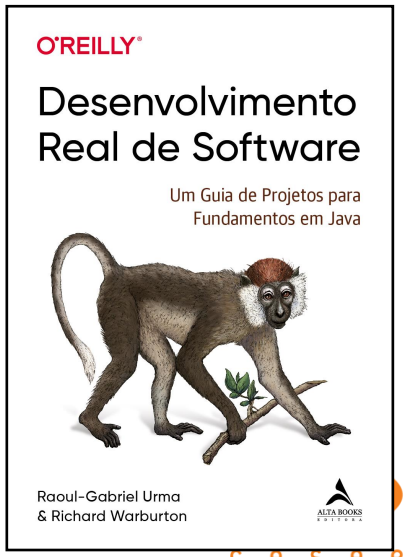




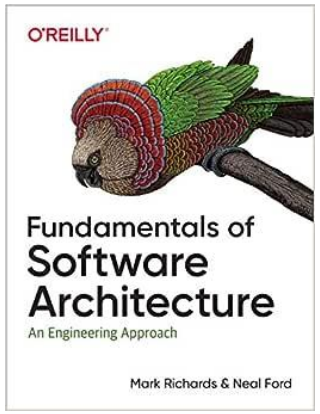
Bibliografia adicional

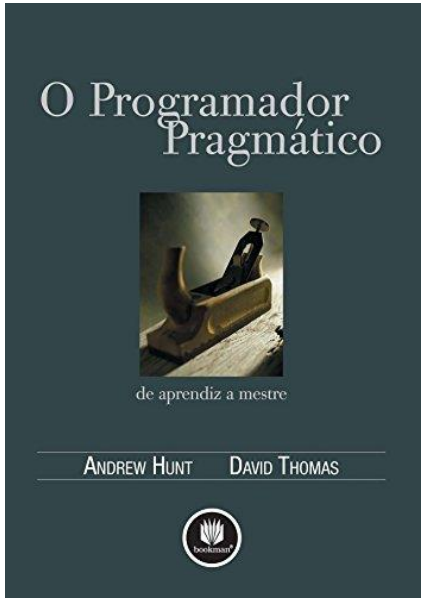


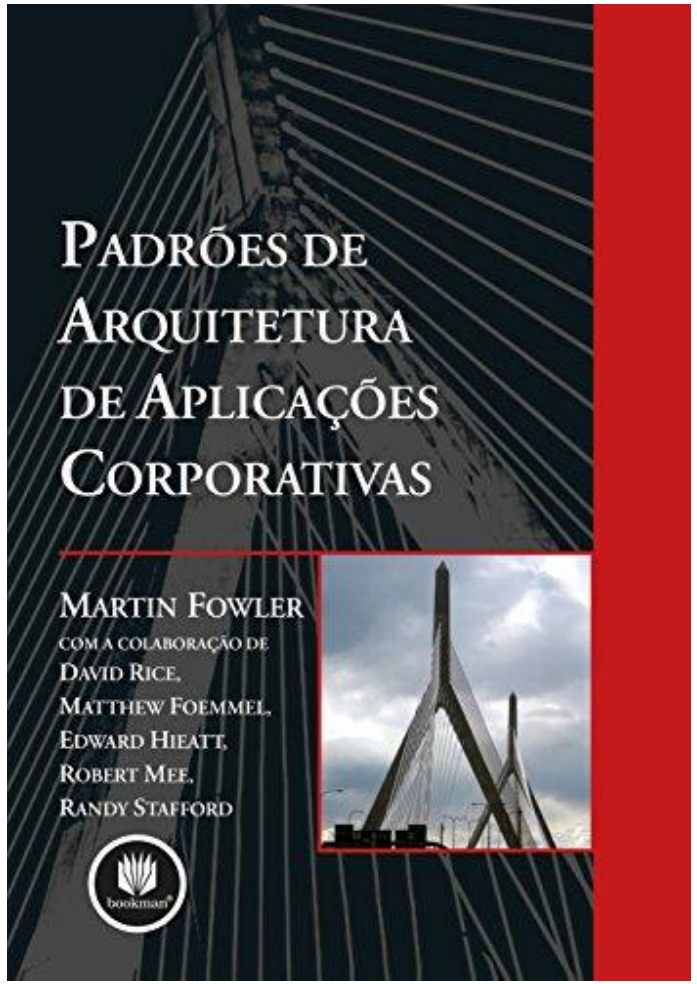


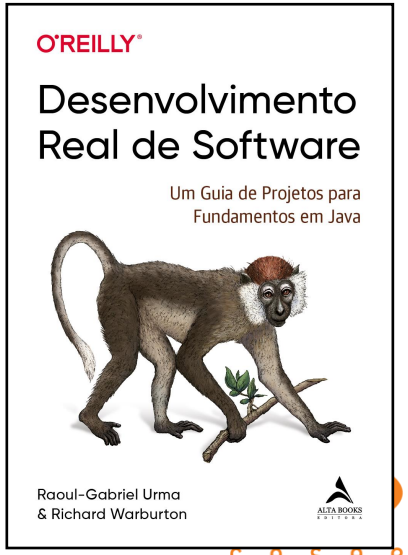


Bibliografia adicional

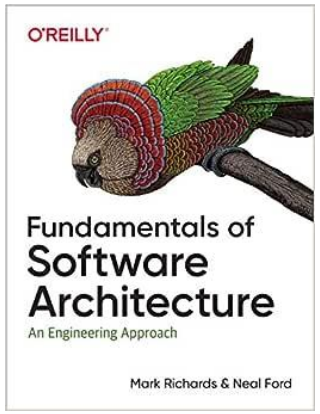


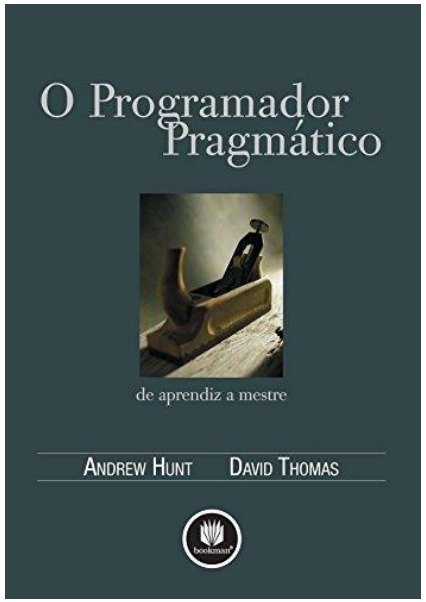


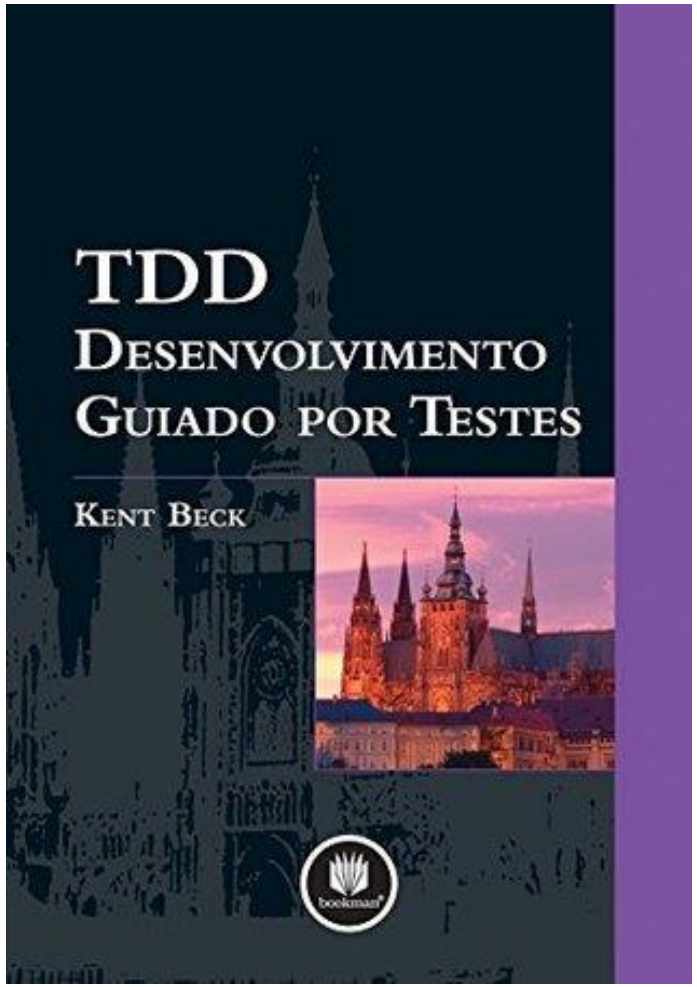


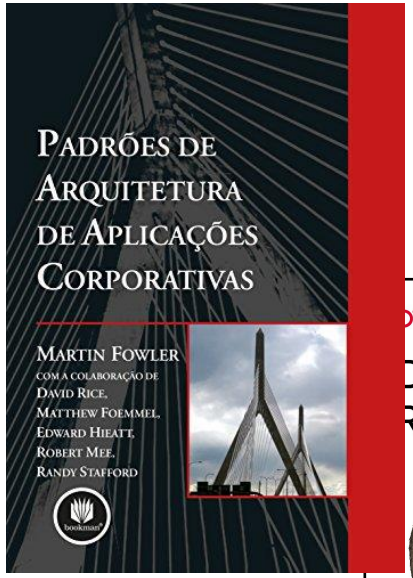


Bibliografia adicional

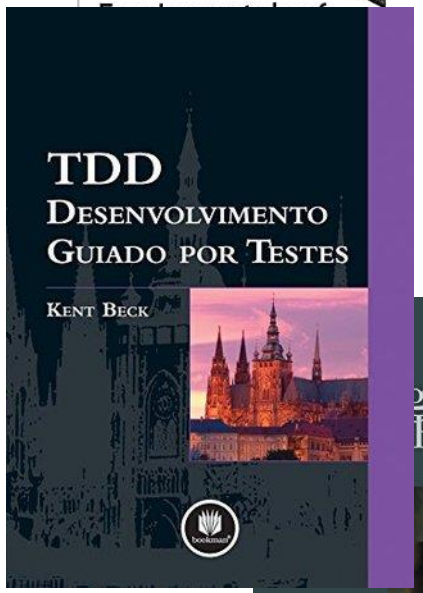


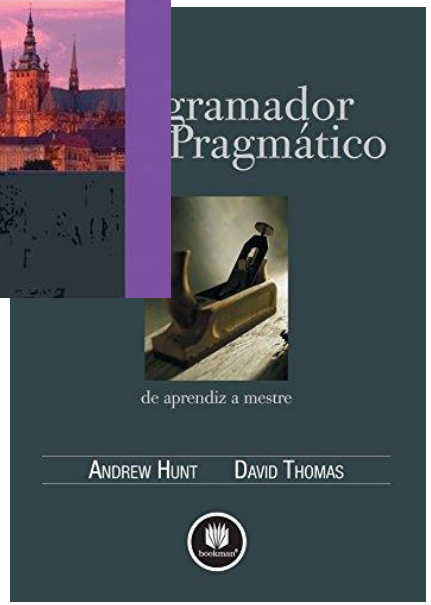


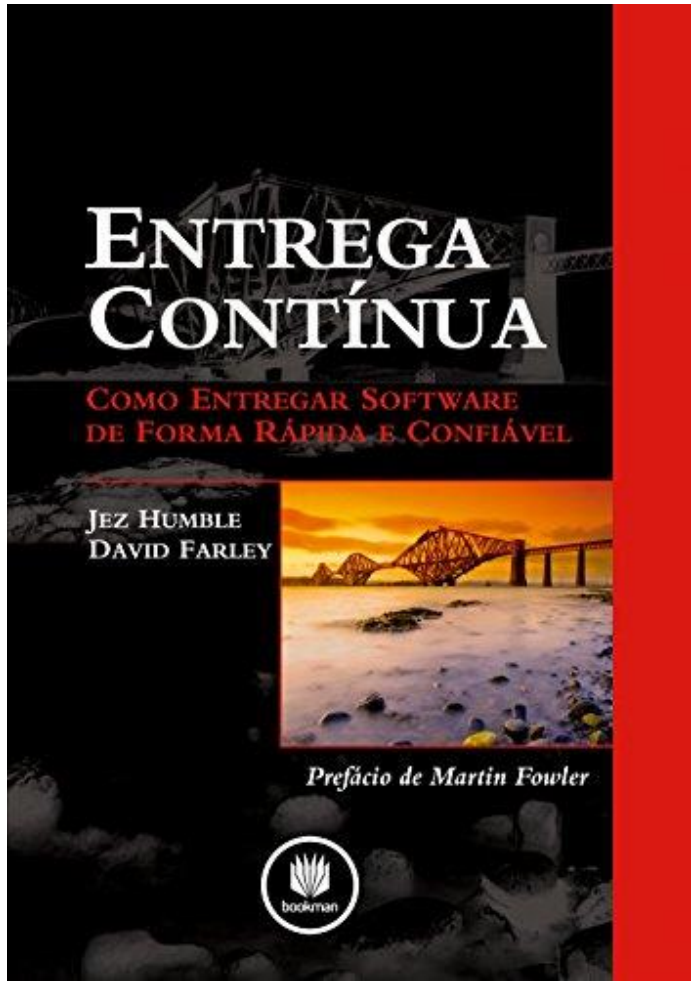


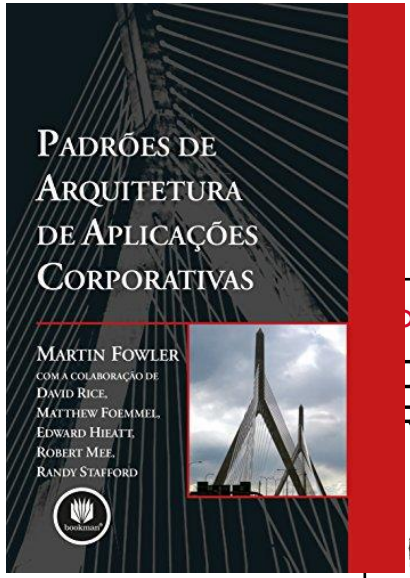


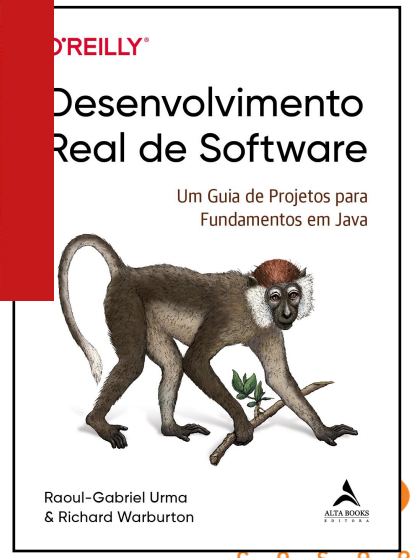
Bibliografia adicional



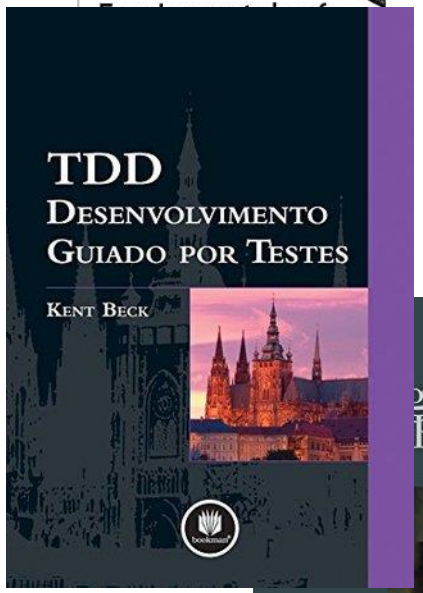


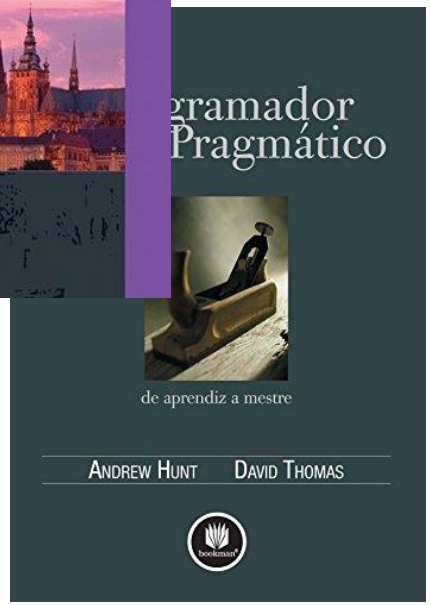


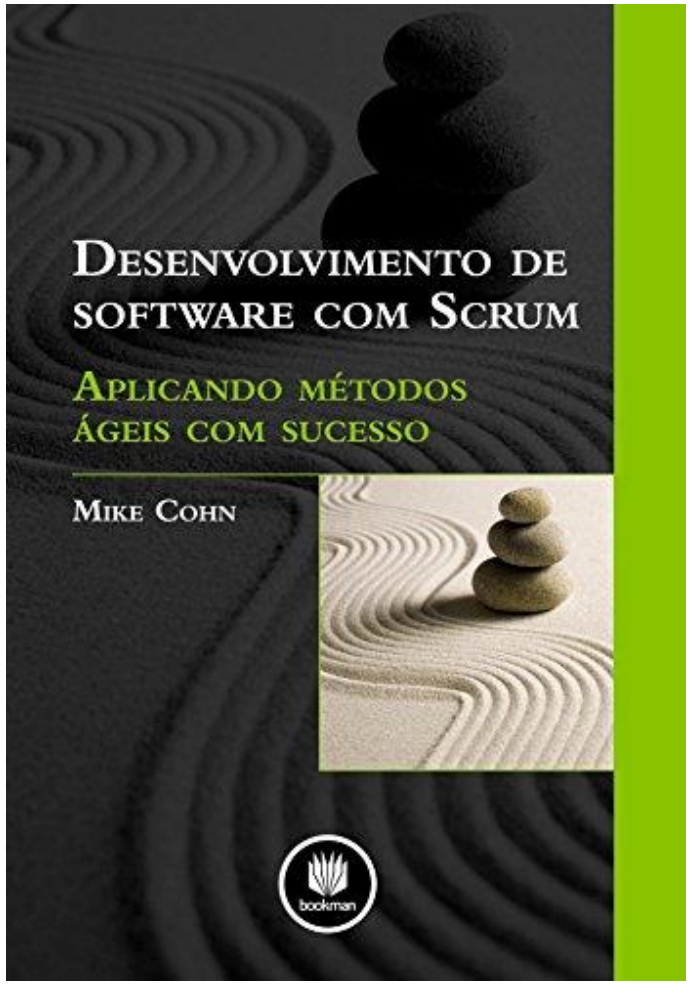


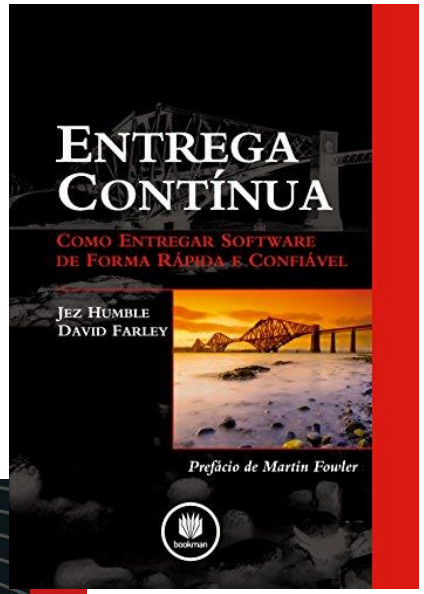


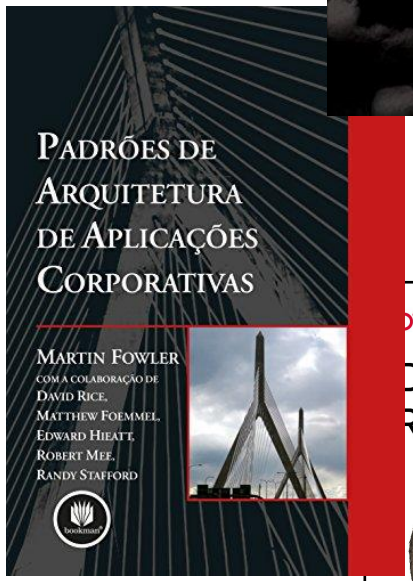
Bibliografia adicional



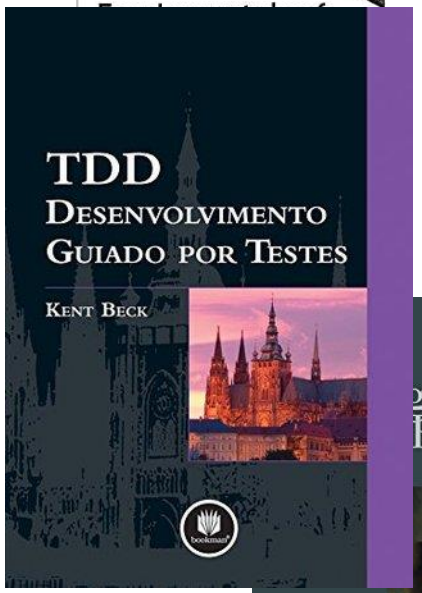


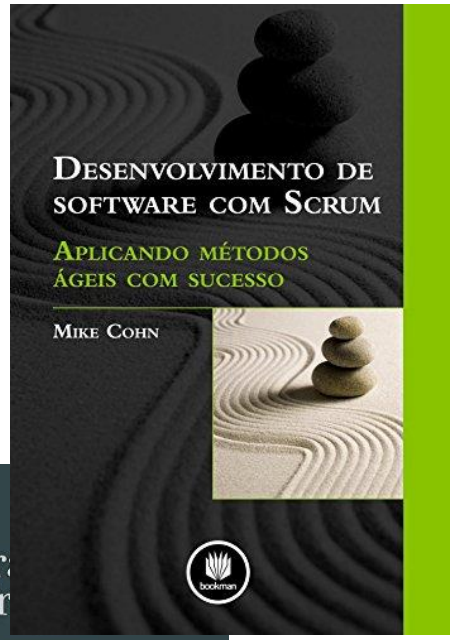


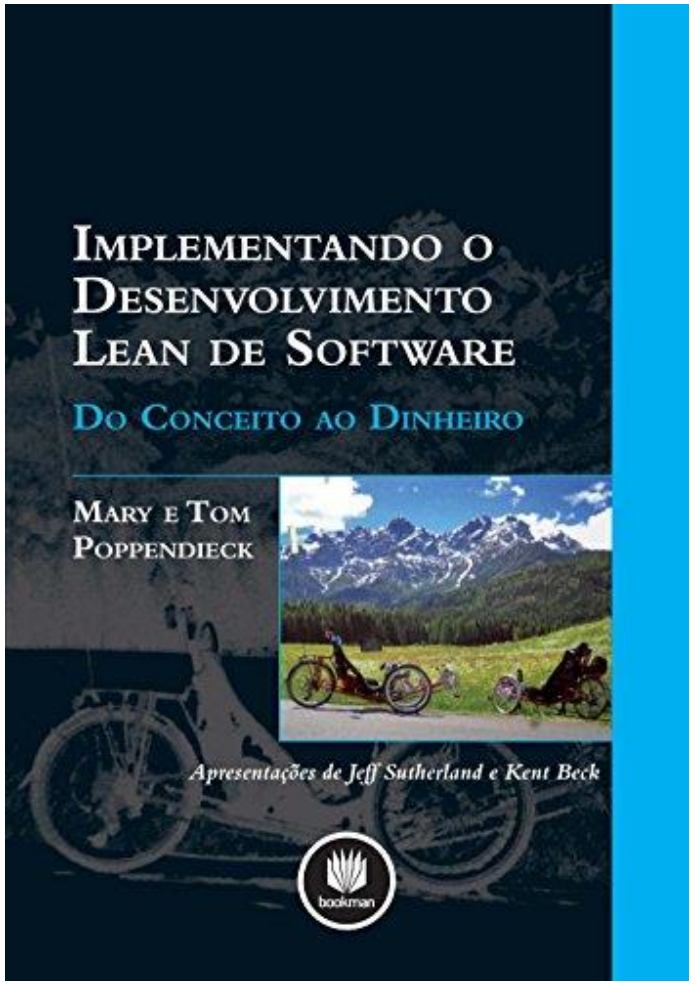


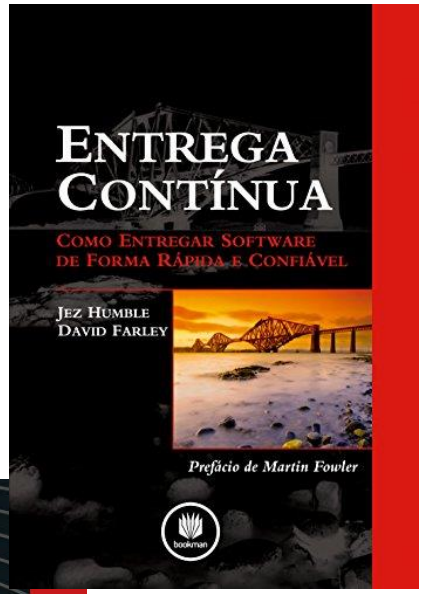


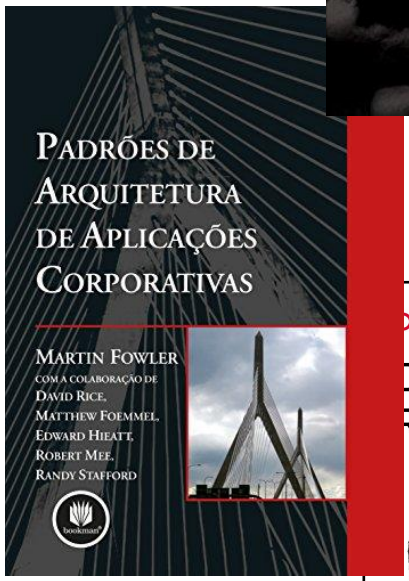
Bibliografia adicional











Informações Importantes

- Horário das aulas:
  - Segundas e Quartas - 8:15 às 10:15 (Turma B) ou 10:30 às 12:30 (Turma A)

- Serão realizadas duas chamadas por aula
  - O momento da chamada será o mais conveniente para o andamento da aula

- Faltas **NÃO** serão abonadas, o total não deve ultrapassar **25%**
  - Qualquer tipo de ausência (incluindo doença) conta como falta

C.E.S.A.R school

Integração com demais disciplinas

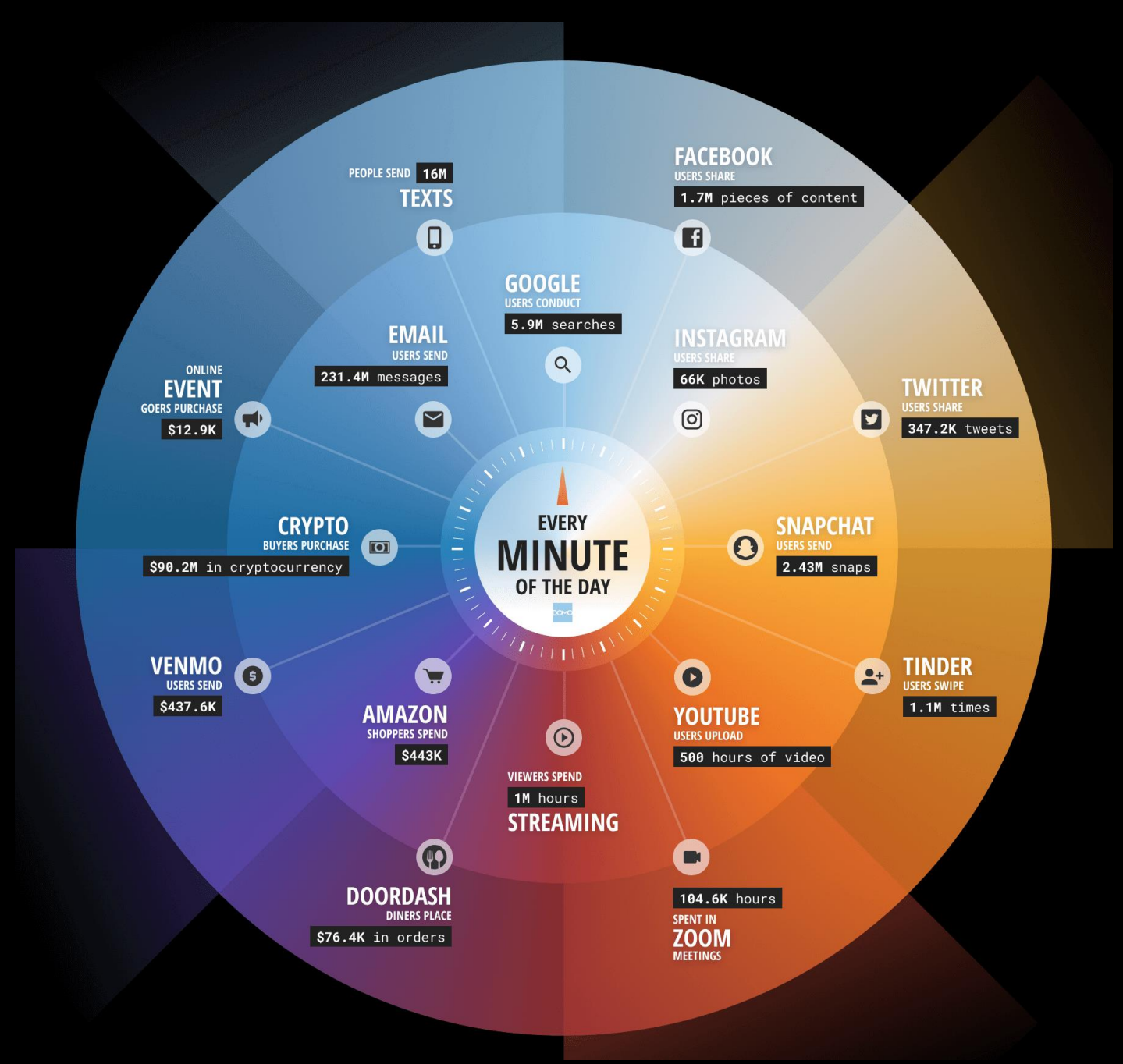
- FP2
  - Os conceitos relacionados ao gerenciamento de projetos serão abordados com maior profundidade na disciplina de FP2.
  - Essa disciplina se restringirá a apresentar como são conduzidos os projetos com metodologias ágeis.
- Projeto 2
  - O que você aprender e aplicar em FDS será aproveitado na disciplina de Projeto 2.

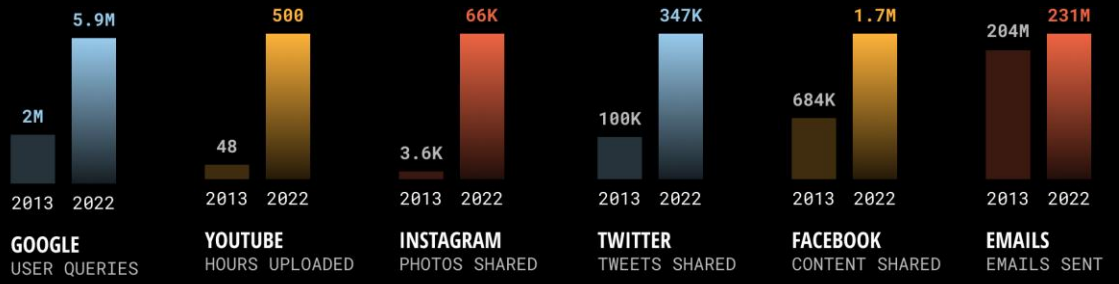
Como desperdiçar essa disciplina

- Faltar aula (física ou mentalmente)
- Não fazer o próprio trabalho
- Trabalhar sozinho, não envolvendo os colegas
- Resistir a novas ferramentas/técnicas
- Ignorar o processo, desenvolver da forma que quiser
- Trabalhar por/para pontos e não para aprender
- Supor que apenas assistir as aulas é suficiente para "passar"
- Multitarefa (email, mensagens instantâneas, etc.) durante a aula

C.E.S.A.R school

Por que estudar “Engenharia de Software”?

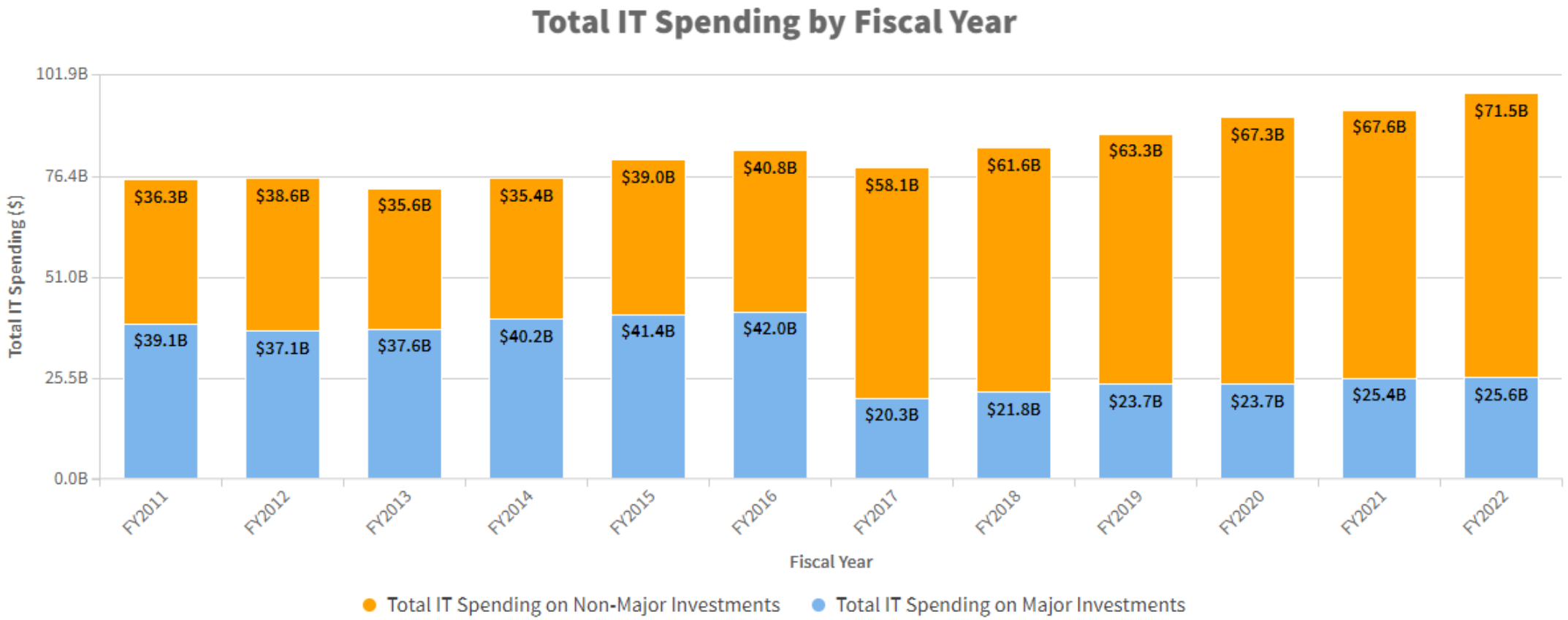




https://www.domo.com/data-never-sleeps

C.e.S.A.R school

Evolução de gastos em TI pelo governo dos EUA



https://itdashboard.gov/drupal/summary/000

C.E.S.A.R school

Custos dos problemas em Software

2003

$59.5 bilhões / ano

Apenas EUA

Fonte: Departamento de Comércio - NIST

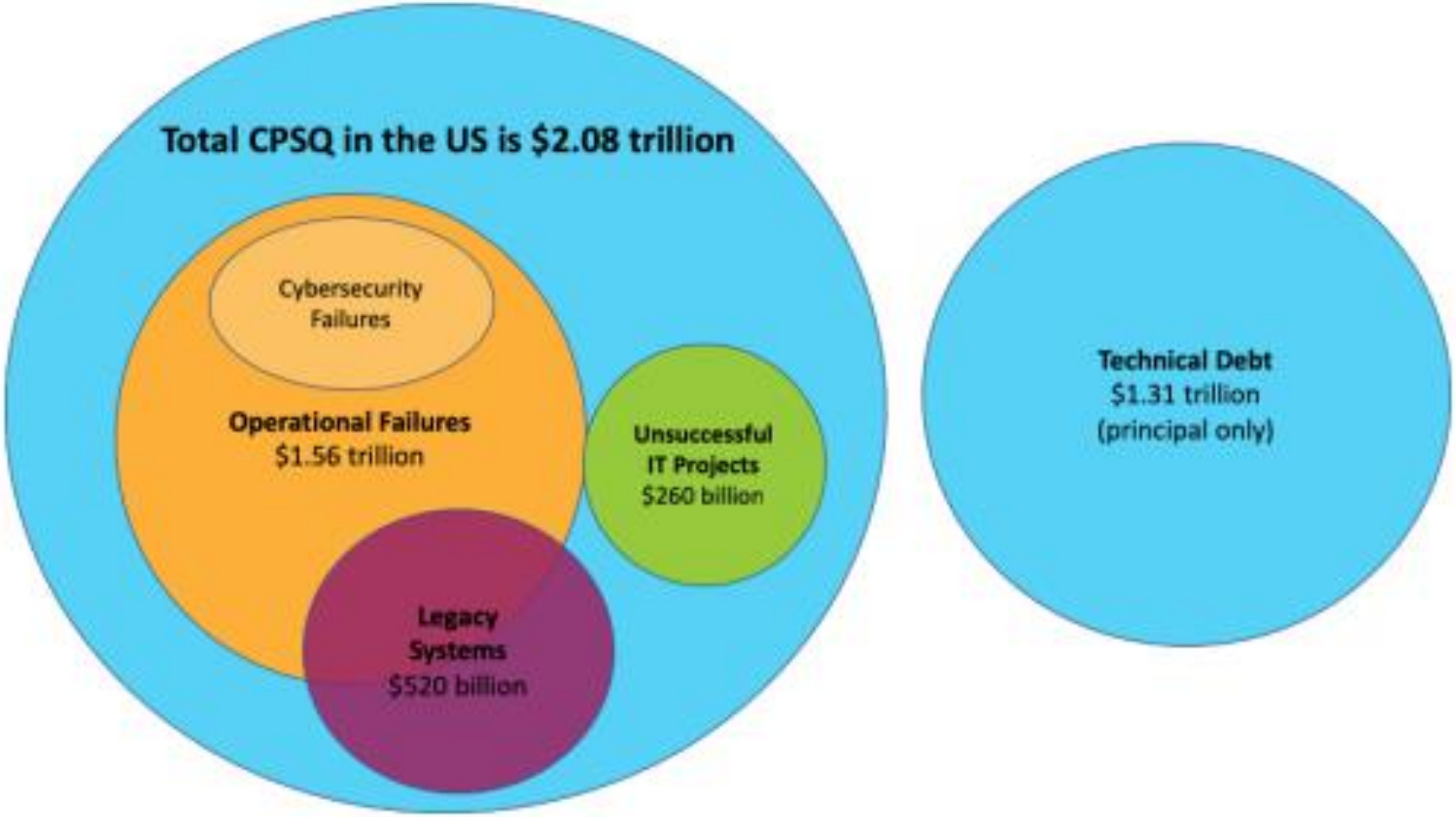
2013

$312 bilhões / ano

Em todo o mundo

Fonte: Universidade de Cambridge

Custo de problemas em Software - 2020



https://www.it-cisq.org/pdf/CPSQ-2020-report.pdf

C.E.S.A.R school

Epic Software Failures

- Ariane 5 (1996)

https://www.youtube.com/watch?v=5tJPXYA0Nec

https://www.youtube.com/watch?v=W3YJeoYgozw

C.e.S.A.R school

Epic Software Failures

- Therac-25 (1985)
  - Aparelho de radioterapia
  - “Condição de corrida” fazia com que ele fosse configurado para aplicar doses letais de radiação
  - Ao menos 5 morreram

Epic Software Failures

- Desconexão da rede da AT&T (1990)
  - Bug no software de controle dos switches de longa distância da AT&T causou a desconexão e reboot de 114 switches a cada 6 segundos.
  - Aproximadamente 60.000 pessoas ficaram sem conexão de longa distância por 9 horas.
  - Causado por um comando **`break`** que deveria finalizar um comando **`if`**, mas infelizmente finalizou o comando **`switch`** que continha o **`if`**

Epic Software Failures

- Instituto Nacional do Câncer – Panamá (2000)
  - O software de planejamento de terapia criado pela Multidata Systems International errou o cálculo do tempo de exposição apropriado para os pacientes de radioterapia.
  - Ao menos 8 pacientes morreram e 20 receberam doses capazes de causar sérios problemas de saúde.
  - Durante o processo de configuração, a direção em que um “buraco” era desenhado alterava a dosagem
    - Se o “buraco” fosse desenhado em uma direção, a exposição era calculada corretamente
    - Se o “buraco” fosse desenhado em outra direção, a exposição recomendada era o dobro da necessária

Epic Software Failures

- Nike & i2 Technologies (2000)
  - Software de planejamento de oferta e demanda produzido pela i2 Technologies solicitou:
    - Milhares de Air Garnett a mais do que poderiam ser vendidos.
    - Milhares de Air Jordan a menos do que o necessário.
  - Prejuízo de mais de $100MI em vendas não realizadas
  - Ações da empresa caíram 20%
  - Queixas da Nike: Software muito lento, de difícil integração e com alguns bugs.

Epic Software Failures

- St. Mary’s Mercy Medical Center (2003)
  - Erro de software declarou 8.500 pacientes **vivos** como **mortos**
  - Todos eles haviam sido atendidos no período de 25/10/2002 a 11/12/2002
  - O sistema ainda notificou incorretamente a Seguridade Social dos EUA, as seguradoras dos pacientes e os próprios pacientes
  - Falha durante a atualização da versão do sistema de gerenciamento de pacientes trocou o código de “teve alta” pelo código de “morreu”

Epic Software Failures

- Inauguração do Terminal 5 do Heathrow (2008)
  - O terminal contava com um novo sistema para gestão de bagagens “melhor e mais robusto”
  - Antes de inaugurar o terminal, os engenheiros testaram o sistema com mais de 12.000 bagagens de teste
  - NENHUM teste incluía a operação de remoção de bagagens manualmente
    - Passageiros costumam esquecer itens importantes dentro das malas (documentos, itens de valor etc.)
  - O sistema começou a funcionar erraticamente e se desligou
  - Nos 10 dias seguintes mais de 42.000 bagagens não viajaram com seus donos e mais de 500 voos foram cancelados

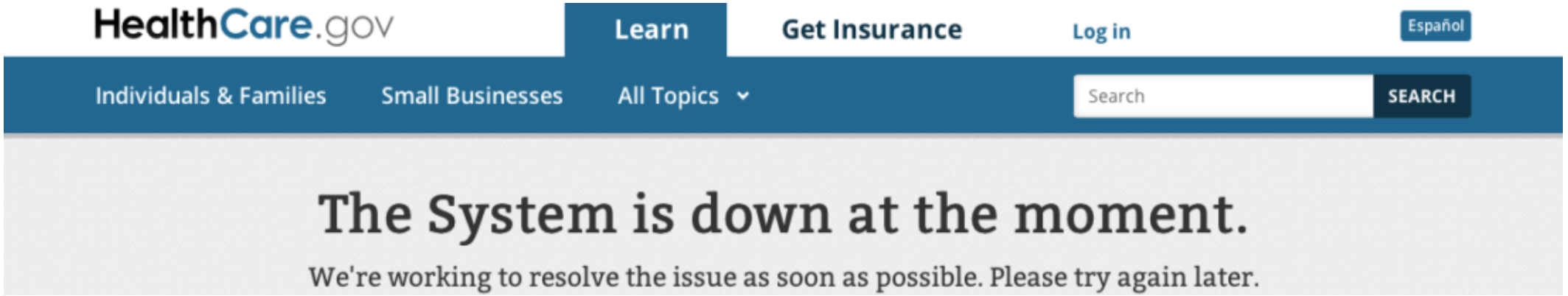
Epic Software Failures

- Knight Capital Group (2012)
  - Algoritmo de trading de ações comprou no “valor ofertado” (offer) e vendeu no “valor pedido” (bid) 150 diferentes ações
  - Cerca de 2.400 operações por minuto
  - As ações da empresa caíram 62% em um único dia
  - O prejuízo foi de **$440MI** em **30 minutos**

O caso HealthCare.gov

"I'm going to try and download every movie ever made, and you're going to try to sign up for Obamacare, and we'll see which happens first"

de Jon Stewart (The Daily Show)
para Kathleen Sebelius (Secretary of HHS)



https://digital.hbs.edu/platform-rctom/submission/the-failed-launch-of-www-healthcare-gov/

C.E.S.A.R school

O caso HealthCare.gov

2h

Tempo que o site ficou no ar após o lançamento

6

Número de pessoas que conseguiram finalizar o processo no 1º dia

$93.7M

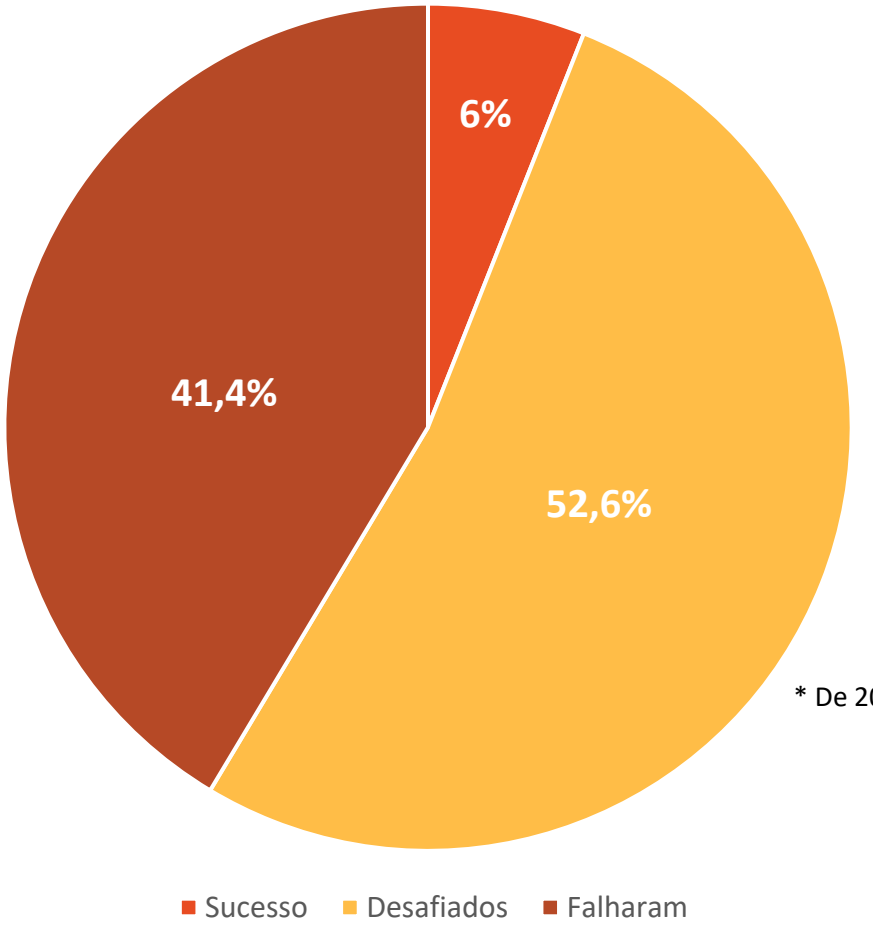
Orçamento do projeto

$1.7B

Custo final do projeto

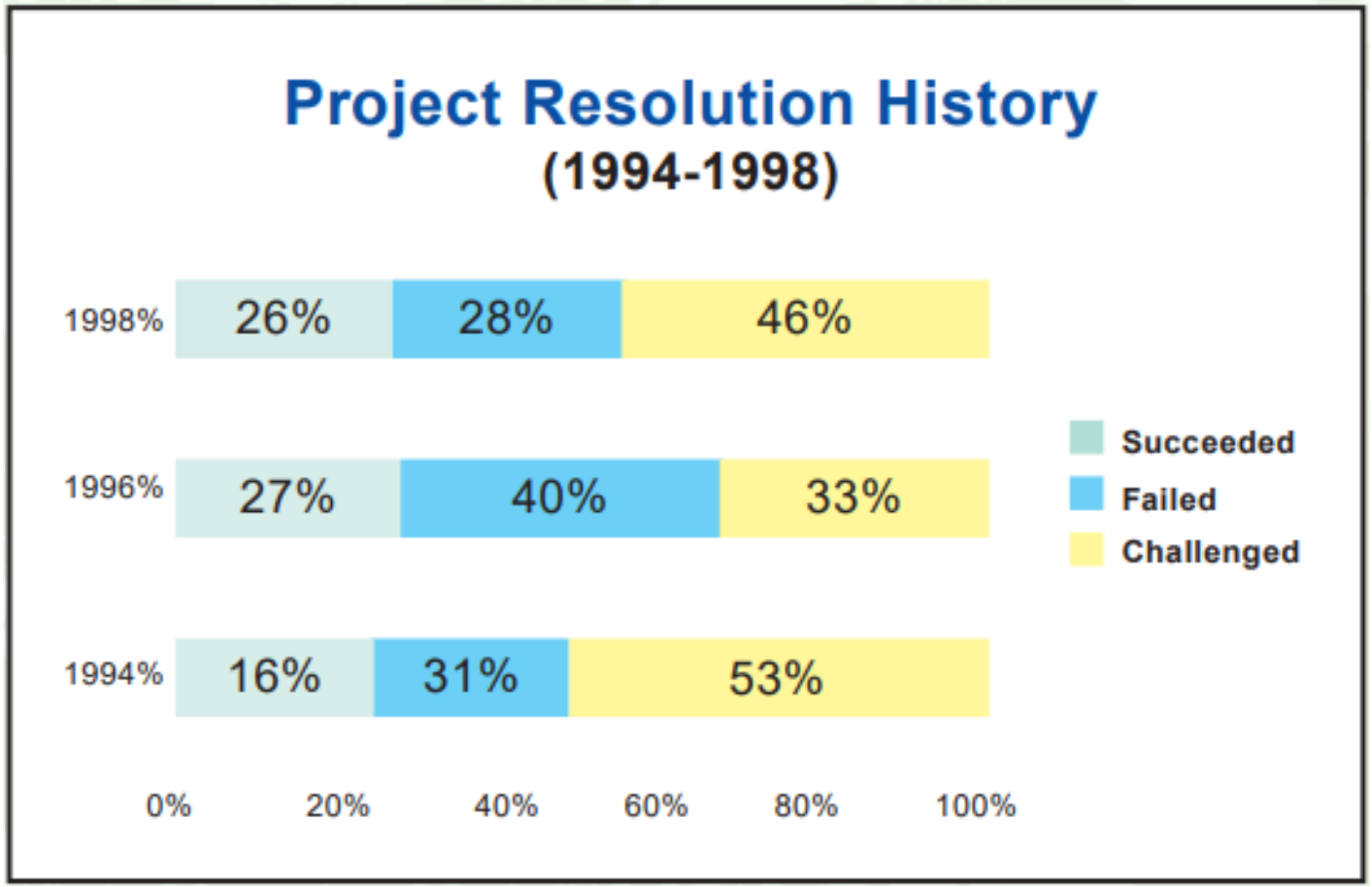
C.E.S.A.R school

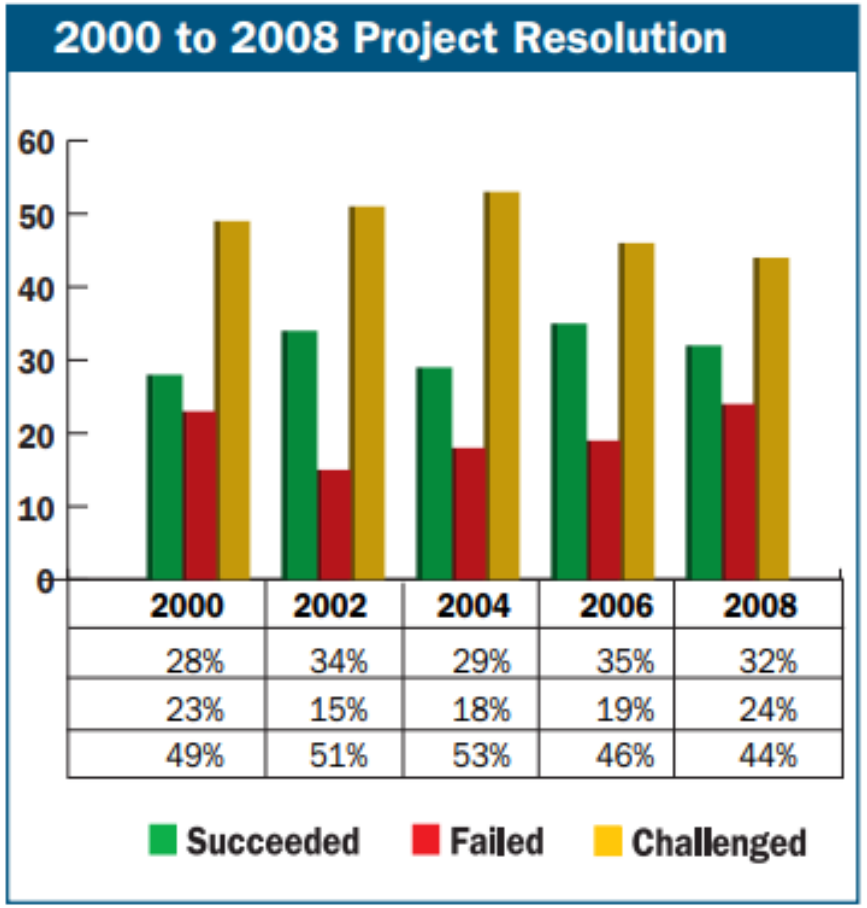
Projetos de TI no governo dos EUA



* De 2005 a 2015

Standish Group's CHAOS Report





C.e.S.A.R school

Standish Group's CHAOS Report

TRADITIONAL RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

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

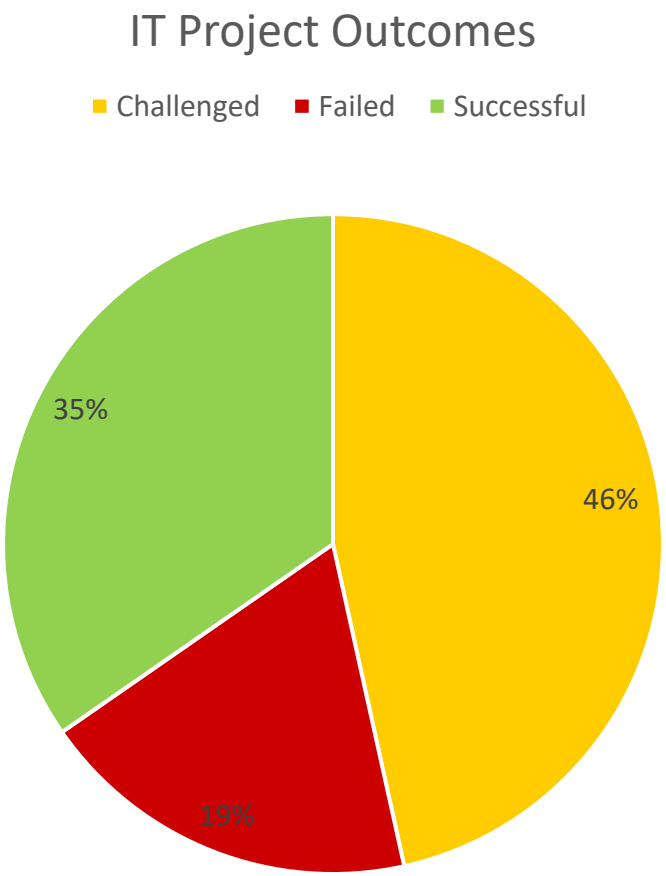
MODERN RESOLUTION FOR ALL PROJECTS

	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.

C.e.S.A.R school

CHAOS 2020: Beyond Infinity

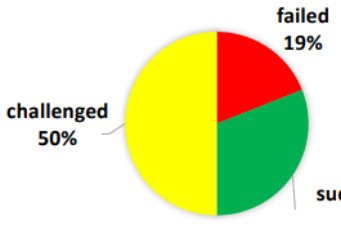


C.e.S.A.R school

Standish Group's CHAOS Report

Project Success
Quick Reference Card

Based on CHAOS 2020: Beyond Infinity Overview. January'2021, QRC by Henny Portman



Modern measurement
(software projects)

Good Sponsor, Good Team, and Good Place are the only things we need to improve and build on to improve project performance.

The Good Place is where the sponsor and team work to create the product. It's made up of the people who support both sponsor and team. These people can be helpful or destructive. It's imperative that the organization work to improve their skills if a project is to succeed. This area is the hardest to mitigate, since each project is touched by so many people. Principles for a Good Place are:

- The Decision Latency Principle
- The Emotional Maturity Principle
- The Communication Principle
- The User Involvement Principle
- The Five Deadly Sins Principle
- The Negotiation Principle
- The Competency Principle
- The Optimization Principle
- The Rapid Execution Principle
- The Enterprise Architecture Principle

Successful project Resolution by Good Place Maturity Level:

highly mature	50%
mature	34%
moderately mature	23%
not mature	23%

The Good Team is the project's workhorse. They do the heavy lifting. The sponsor breathes life into the project, but the team takes that breath and uses it to create a viable product that the organization can use and from which it derives value. Since we recommend small teams, this is the second easiest area to improve. Principles for a Good Team are:

- The Influential Principle
- The Mindfulness Principle
- The Five Deadly Sins Principle
- The Problem-Solver Principle
- The Communication Principle
- The Acceptance Principle
- The Respectfulness Principle
- The Confrontationist Principle
- The Civility Principle
- The Driven Principle

Successful project Resolution by Good Team Maturity Level:

highly mature	66%
mature	46%
moderately mature	21%
not mature	1%

The Good Sponsor is the soul of the project. The sponsor breathes life into a project, and without the sponsor there is no project. Improving the skills of the project sponsor is the number-one factor of success – and also the easiest to improve upon, since each project has only one.
Principles for a Good Sponsor are:

- The Decision Latency principle
- The Vision Principle
- The Work Smart Principle
- The Daydream Principle
- The Influence Principle
- The Passionate Principle
- The People Principle
- The Tension Principle
- The Torque Principle
- The Progress Principle

Successful project Resolution by Good Sponsor Maturity Level:

highly mature	67%
mature	33%
moderately mature	21%
not mature	18%

C.e.S.A.R school

Standish Group's CHAOS Report

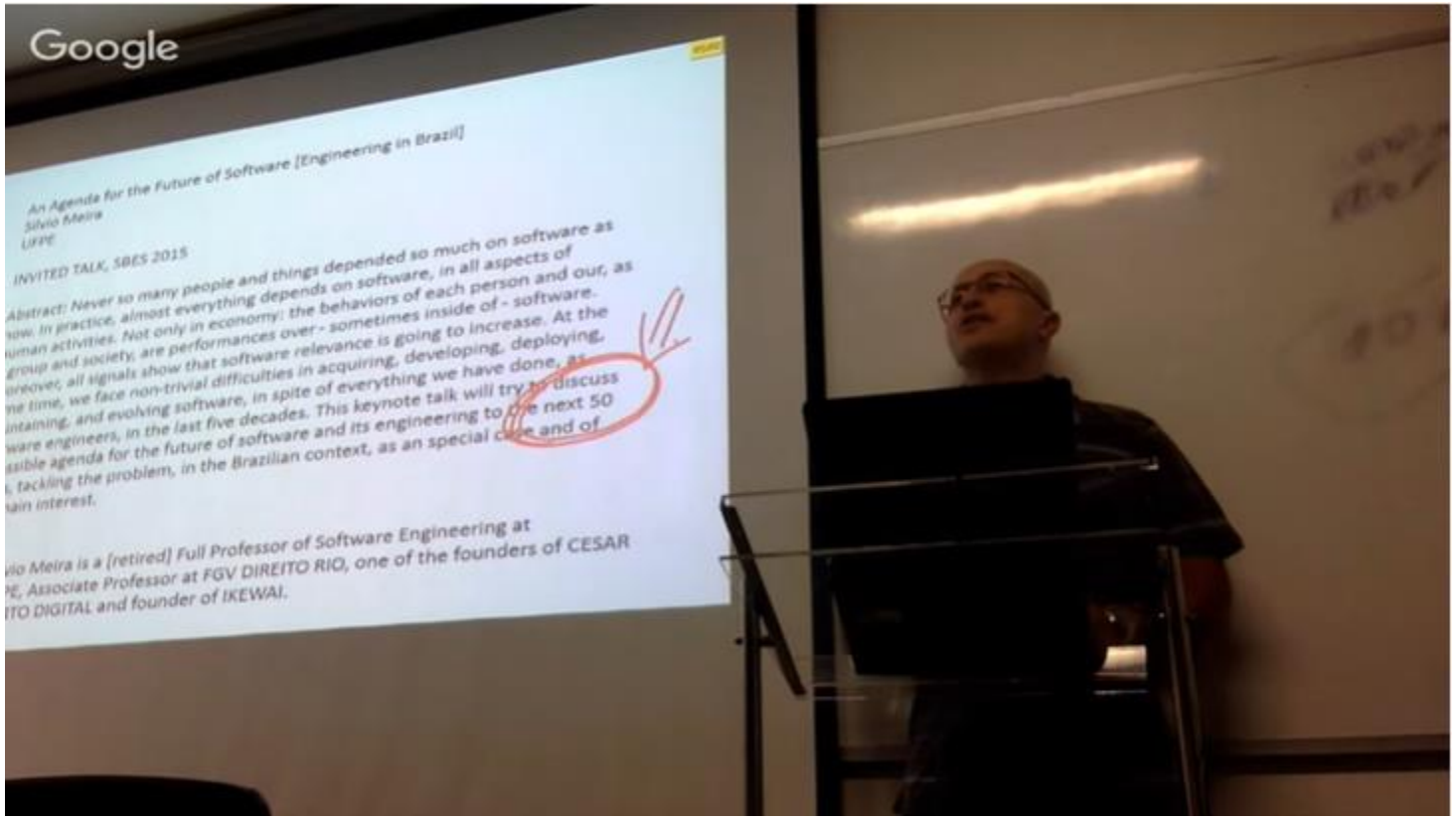
CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

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.

C.e.S.A.R school

An Agenda for the Future of Software [Engineering in Brazil]



https://www.youtube.com/watch?v=eujhiejLL7c

C.E.S.A.R school

Bem-vindos e aproveitem a jornada