

PROCESSOS DE SOFTWARE

RAFAEL PRIKLADNICKI

2023/2 - Aula 2

Em 2019...

**Como será o
mundo em 2029?**



Praça São Pedro



Praça São Pedro



Em 2020...

**Como será o
mundo em 2021?**



**“Os analfabetos do século XXI
não serão os que não
souberem ler ou escrever, mas
os que não souberem aprender,
desaprender e reaprender.”**

(Alvin Toffler)

FUTURO

**"Como eu tenho dito muitas vezes, o futuro já chegou.
Só não está uniformemente distribuído."**

William Gibson
30/11/1999

A Natureza da Nova Economia é VUCA!

Volatility



VOLATILIDADE

A velocidade e a frequência das mudanças demandam rápidas reações diante de cenários dos quais não temos controle.

Uncertainty



INCERTEZA

Novos cenários não familiares e resultados cada vez mais imprevisíveis demandam a tomada de decisão com poucas informações.

Complexity



COMPLEXIDADE

Múltiplas interdependências em meio a uma crescente interconectividade global demandam uma visão sistêmica do ambiente.

Ambiguity



AMBIGUIDADE

Múltiplas perspectivas e interpretações dos cenários existentes demandam adaptação às mudanças.

Muitas
mudanças

Volátil

V

Complexo

C

Muitas
variáveis

Muitas
incertezas

U

Incerto

A

Ambíguo

Muitas
interpretações

Resiliência e
propósito

Volátil

V

Complexo

C

Ser
multidisciplinar

Autonomia e
adaptação

U

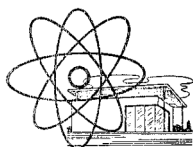
Incerto

A

Ambíguo

Visão
sistêmica

Milagres Eletrônicos do Futuro



Quando nossa anfitriã aparece, depois de trocados os cumprimentos da praxe, explicamos nossa missão e lhe pedimos para nos mostrar como se vive num lar do ano de 2006.

— Pois não — diz-nos a moça. — Tenho todo prazer em lhes mostrar a casa e responder a qualquer pergunta que queiram fazer.

— Bom, para começar, quando chegamos perto da porta, apesar de não termos batido nela, a sra. apareceu naquela tela e convidou-nos a entrar.

— Isso é simples — volveu a moça. — Quando os srs. se aproximaram da porta, um aparelho eletrônico registrou a sua presença e téz soar uma cigarra na cozinha. Eu, então, simplesmente acionei o interruptor de um vídeo-intercomunicador, para vê-los e falar-lhes. Quando os convidei a entrar, um aparelho eletrônico comandado pela voz desarmou a fechadura elétrica e abriu a porta.

ELETRÔNICA POPULAR — 341

OUTUBRO, 1956 — Pág. 21

— A sra. não tem televisor? — indagamos.

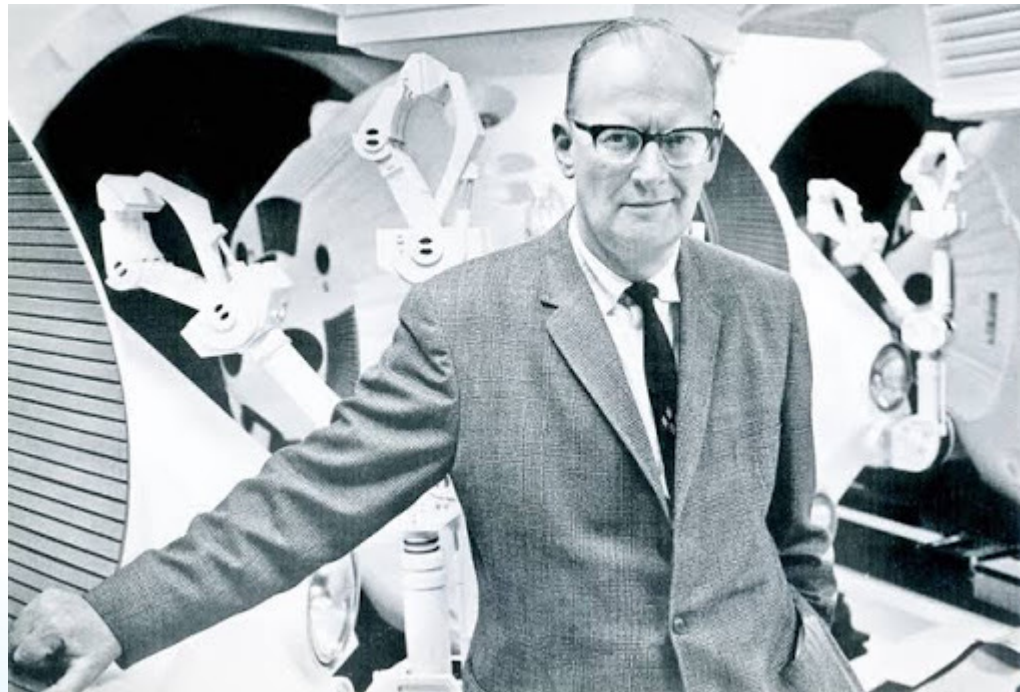
— E' claro que tenho. Este quadro na parede é o nosso televisor — responde ela, apontando para um grande quadro na parede, representando uma paisagem. — Quando não esiamos recebendo um programa, aparece uma paisagem na tela. Os contrôles estão na moldura. Veja!

Ela toca num botão, colocado disfarçadamente na moldura, e a paisagem desaparece, cedendo lugar a um programa de televisão. A imagem é tridimensional e em cores. O som parece emanar de diversos pontos da cena.

— O televisor só tem alguns centímetros de profundidade. Todos os circuitos eletrônicos estão incluídos na moldura, e tem também um gravador automático de programas — explicou ela. — Se eu quero assistir a um determinado programa, mas estou na ocasião, ocupada com as crianças, viro uma chave e gravo o programa para vê-lo mais tarde quando tiver tempo. O receptor de televisão também funciona como um jornal ou uma revista.

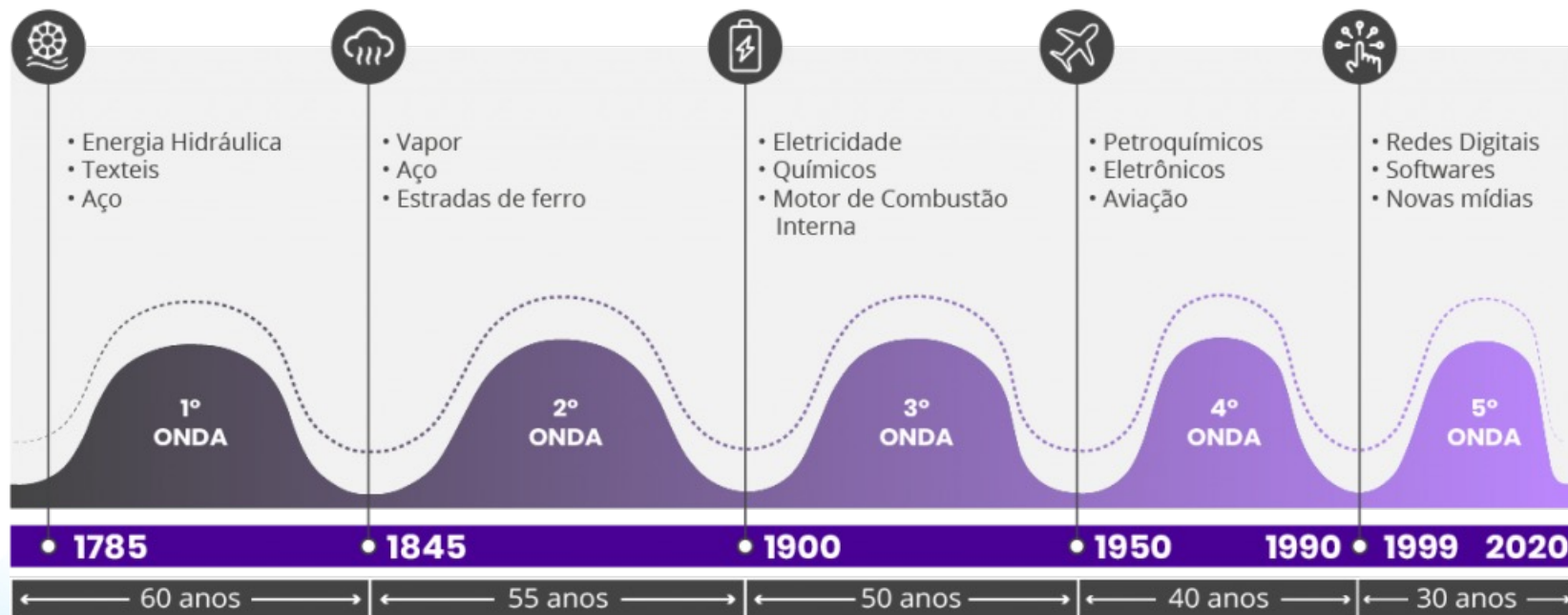
Regressando à nossa era, refletimos um momento sobre o que vimos. Portentosos milagres, sem dúvida alguma, mas podemos ver as suas sementes lançadas hoje em dia. Temos rádio-receptores e gravadores de bolso. Já foram realizadas experiências com baterias solares e atômicas. Já se gravaram programas de televisão em fita magnética. Computadores gigantesco efetuam milagres em matéria de cálculos, muito além das possibilidades humanas. E já se cozinham alimentos com o radar. Talvez não tenhamos viajado tão longe no tempo como pensávamos!

Arthur C. Clarke, 1974 Internet e PCs



As Ondas de Schumpeter

O processo de inovação é cada vez mais rápido em tempos de Internet



As ondas de Schumpeter e seus marcos Para Schumpeter, os negócios vivem ondas de inovação, que surgem e desaparecem. No século 18, a primeira leva inovadora veio com a energia hidráulica, a indústria têxtil e o tratamento do aço. Os ciclos eram longos, duravam de 40 a 60 anos - agora encurtaram.



Leaders

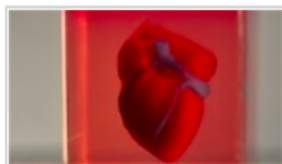
May 6th 2017 edition >

Regulating the internet giants

The world's most valuable resource is no longer oil, but data

<https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>

Health & Science



Human heart made by 3D printer

 facebook

 print

 send to friend

 comment

Breakthrough in medicine: Israeli researches print live heart

Researchers believe within a decade, it will be possible to print all human organs using tissue taken from patients so that organ transplants will be a thing of the past.

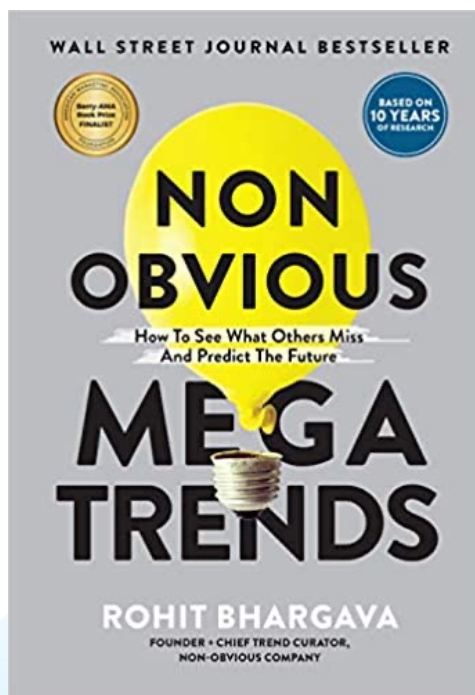
Rotem Elizera | Published: 04.15.19 , 14:45



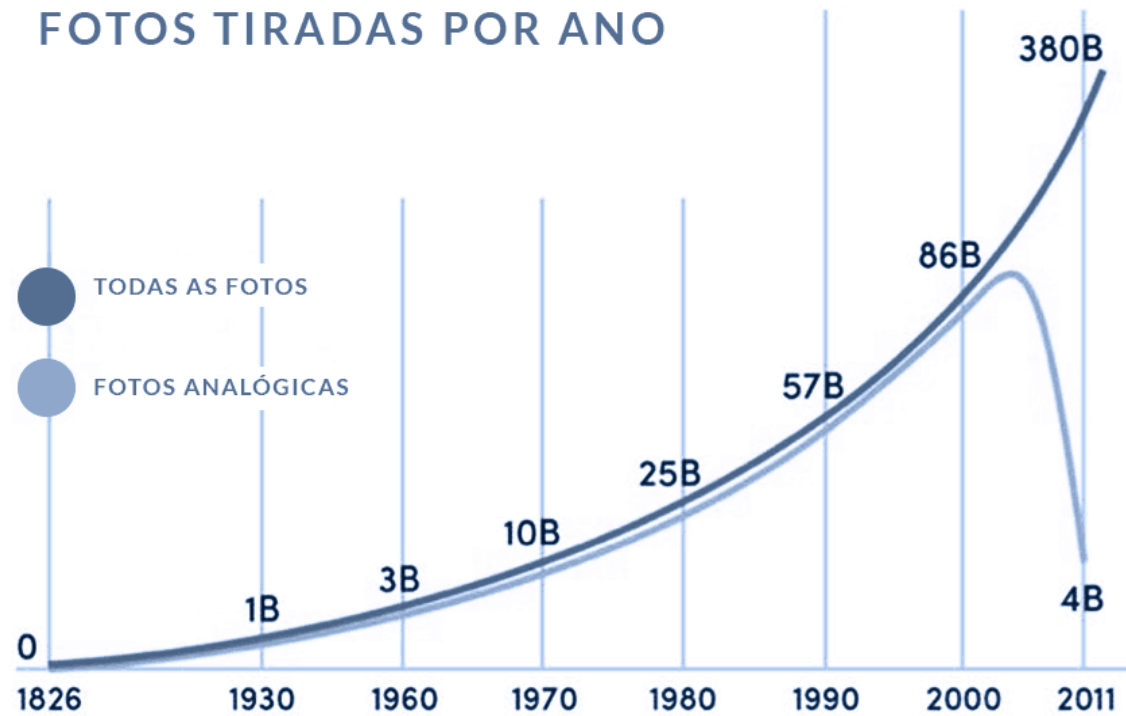
For the first time ever, a live heart was printed and presented as part of a study at the Tel Aviv University, and in the Advanced Science journal.

•Follow Ynetnews on [Facebook](#) and [Twitter](#)

The study opens a window into the future medicine which, they predict, will be free of organ transplantation and rejection, and include specifically printed organs using tissue taken by the patient himself.



FOTOS TIRADAS POR ANO



EXPONENTIAL TECHNOLOGIES IN THE CURVE

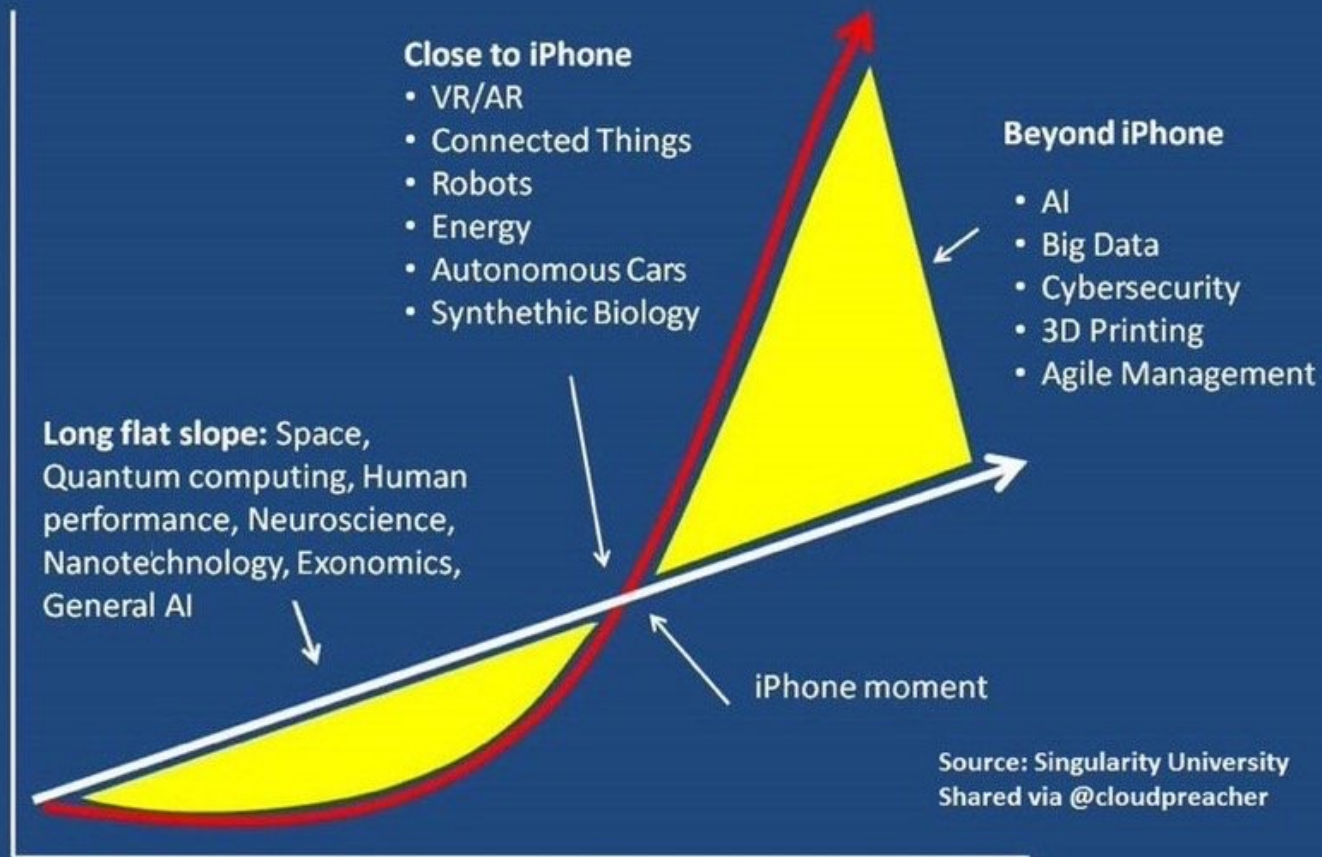


Figure 1. What *appears* to be happening

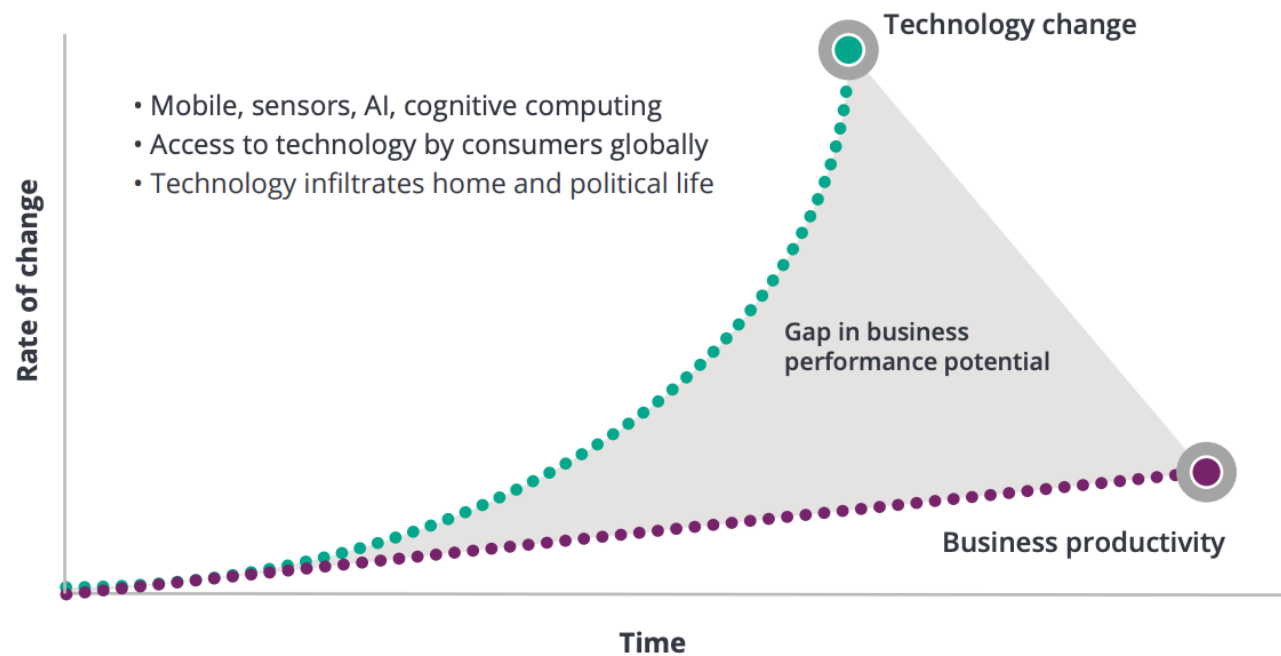
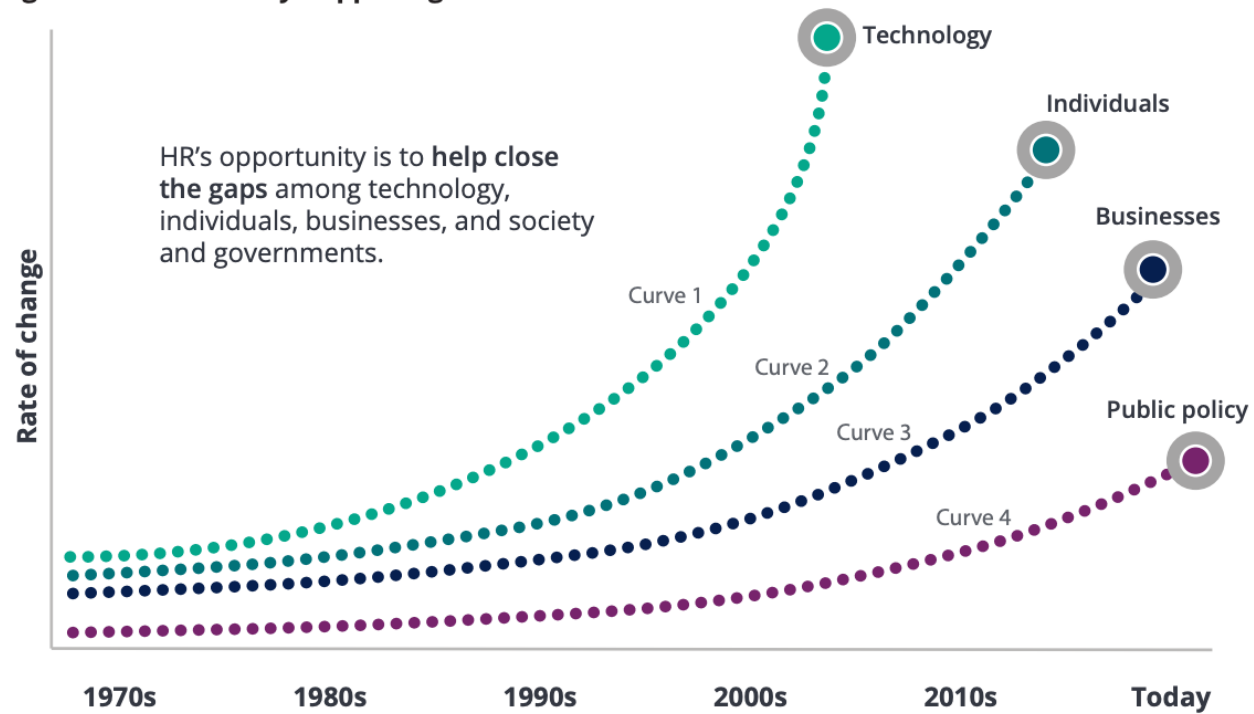


Figure 2. What is *really* happening



Deloitte University Press | dupress.deloitte.com

Mudança de Era

- Escrita
- Revolução agrícola
- Revolução industrial
- Revolução digital (sociedade do conhecimento)
- Revolução pós-digital (tudo é tecnologia)

Paradigma Industrial

- Linear, Segmentado, Repetitivo, Previsível
- Comando controle
- Pensamento industrial - fábrica, linha de montagem
- Lógica da escassez (preciso ter os melhores e não dou acesso a eles - só se pagar)

Paradigma Digital

- Não linear, Conectado, Multidisciplinar
- Exponencialmente imprevisível
- Pensamento digital: colaboração, exponencial, disrupção
- Lógica da abundância (disponibilizo o que tenho e uso o que os outros disponibilizaram)

Paradigma Digital

- Não linear, Conectado, Multidisciplinar
- Exponencialmente imprevisível
- Pensamento digital: colaboração, exponencial, disrupção
- Lógica da abundância (disponibilizo o que tenho e uso o que os outros disponibilizaram)

Objetivos

1. Entender a ideia de um processo de software
2. Compreender os conceitos e modelos de processos de software
3. Conhecer as atividades fundamentais do processo de engenharia de requisitos de software, desenvolvimento de software, testes e evolução
4. Entender por que os processos devem ser organizados de maneira a lidar com as mudanças nos requisitos e projeto de software

PROCESSES THAT ARE APPLIED TO PRODUCING E.G.



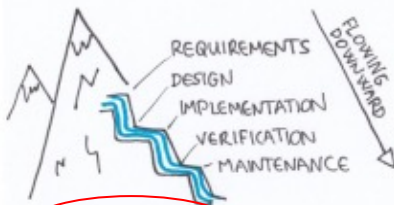
ARE NOT EFFECTIVE FOR PRODUCING DIGITAL PRODUCTS & SERVICES ...

PRODUCT DEVELOPMENT PROCESSES

UX Knowledge Base Sketch #53

DIGITAL PRODUCTS & SERVICES ARE:

- COMPLEX SYSTEMS
- UNPREDICTABLE
- IN CONSTANT CHANGE



WATERFALL-MODEL

- SEQUENTIAL, LINEAR
- NOT MUCH ITERATION

USED IN SOFTWARE DEVELOPMENT FROM THE MIDDLE OF THE 20TH CENTURY

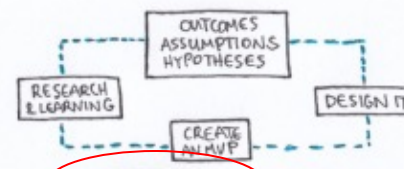
- PROBLEMS:
- NOT FLEXIBLE
 - TOO SLOW
 - KNOWING THE REQUIREMENTS BEFORE THE PROCESS STARTS



AGILE FRAMEWORK

- E.G. SCRUM

- RAPID & FLEXIBLE
 - SHORT CYCLES, SMALLER TIME-FRAMES
 - ADAPTIVE PLANNING
 - ACCOUNTABILITY & REFLECTION
 - CONTINUOUS LEARNING
- PROBLEMS:
- OUTPUT-BASED: SHIPPING FEATURES
 - GREAT VELOCITY
 - BUT: IS THE DIRECTION RIGHT?
 - NO DECISION-MAKING FRAMEWORK



LEAN UX

BY JEFF GOTHHELF & JOSH SEIDEN

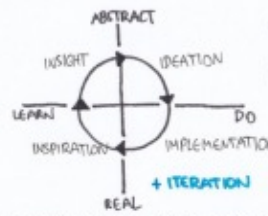
- OUTCOMES INSTEAD OF OUTPUTS: CHANGE IN CUSTOMER BEHAVIOR INSTEAD OF FEATURES
 - DOUBT -> CERTAINTY IN SMALL BATCHES (AGILE!)
 - REFLECTION: ENOUGH EVIDENCE TO HAVE A CYCLE IN THE SAME DIRECTION?
 - REDUCES RISKS & WASTE
- PROBLEMS:
- IF NOT A MATURE PRODUCT, HARDER TO MEASURE CUSTOMER BEHAVIOR
 - DEFINING SUCCESS CAN BE CHALLENGING

IMPORTANT INGREDIENTS:



DESIGN THINKING

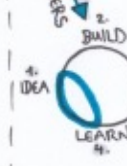
"HUMAN-CENTERED APPROACH C-3 TO INTEGRATE THE NEEDS OF PEOPLE, THE POSSIBILITIES OF TECHNOLOGY, AND THE REQUIREMENTS FOR BUSINESS SUCCESS." (TIM BROWN)



LEAN STARTUP

BY ERIC RIES

BUILD - MEASURE - LEARN FEEDBACK LOOPS
MINIMUM VIABLE PRODUCT (MVP)
GETTING OUT OF THE BUILDING



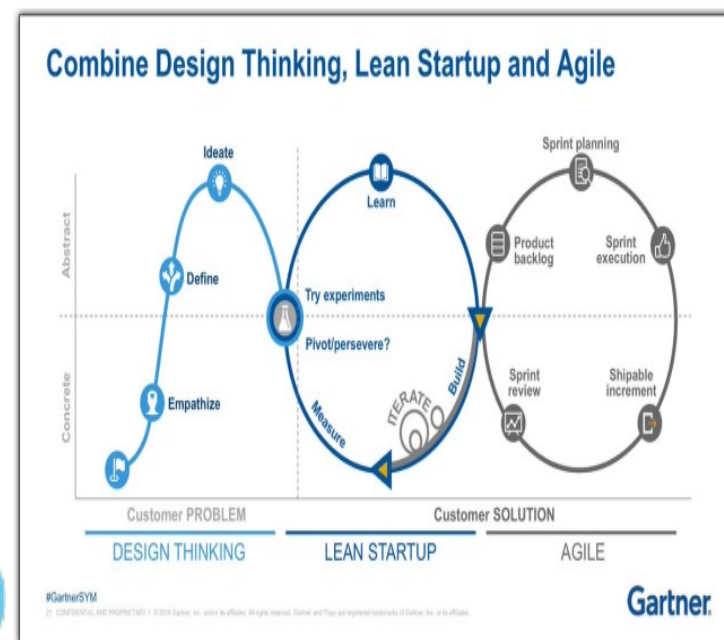
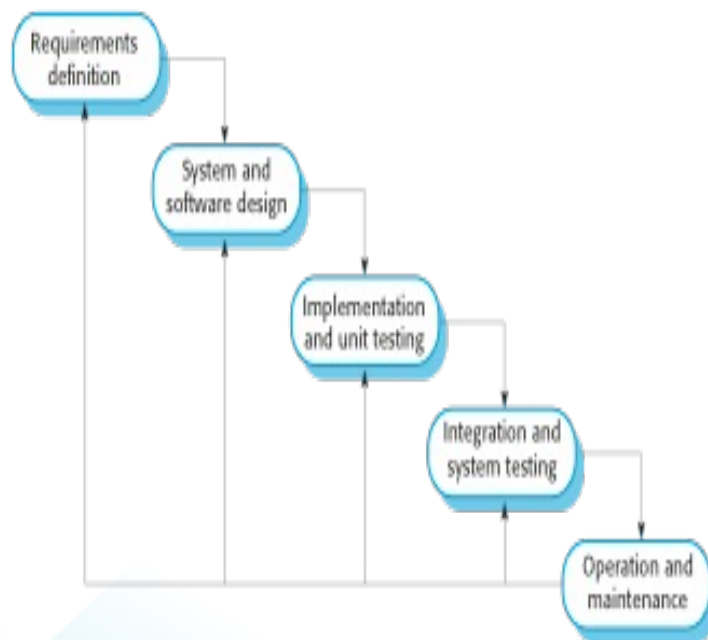
DESIGN SPRINT

"SHORTCUT TO LEARNING WITHOUT BUILDING AND LAUNCHING." (GV.COM/SPRINT)

DUAL TRACK

1. GENERATIVE RESEARCH, DESIGN THINKING -> DESIGN SPRINT
2. LEAN UX + AGILE PRODUCTION

O passado e o futuro



Processo de Software

- Um conjunto estruturado de atividades que levam à produção de um produto de software
- Existem muitos processos de software diferentes, mas todos envolvem
 - Especificação: definir o que o sistema deve fazer (funcionalidades)
 - Projeto e implementação: definir a organização do sistema e implementá-lo
 - Validação: validar que o sistema faz aquilo que os clientes desejam
 - Evolução: alterar o sistema em respostas às mudanças das necessidades dos clientes
- Um modelo de processo de software é uma **representação abstrata de um processo**. Ele apresenta uma descrição de um processo de uma determinada perspectiva.

Processo de Software

- Quando descrevemos e discutimos processos, normalmente falamos sobre atividades desses processos tais como especificar um modelo de dados, projetar uma interface de usuário, etc., e também sobre como organizar essas atividades
- Descrições de processos podem também incluir:
 - Produtos, que são o resultado das atividade do processo
 - Papéis, que refletem as responsabilidades das pessoas envolvidas no processo
 - Pré e pós condições, que são as declarações verdadeiras antes e depois de uma atividade do processo ou da produção de um produto

Prescritivo versus Adaptativo

- Processos dirigidos a planos são aqueles em que **todas as atividades são planejadas com antecedência, e o progresso é avaliado por comparação com o planejamento inicial**
- Em processos ágeis, **o planejamento é gradativo**, e é mais fácil alterar o processo de maneira a refletir as necessidades de mudança dos clientes
- Na prática, a maioria dos processos envolvem elementos de ambas as abordagens
- Não existe processo de software certo ou errado

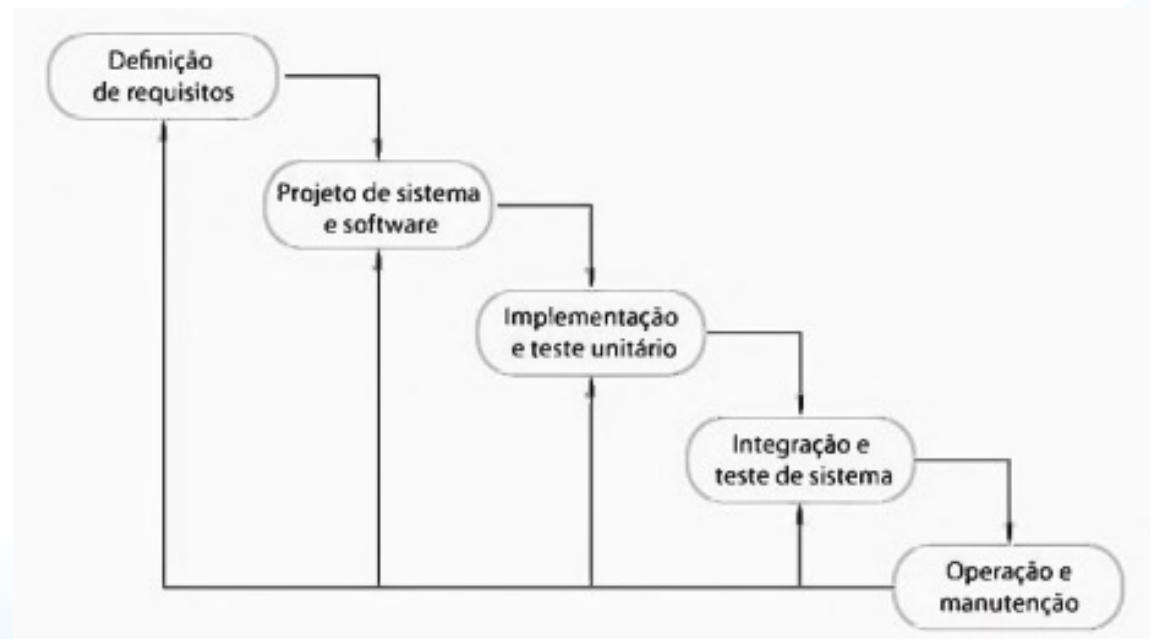
Modelos de Processos

- O Modelo Cascata
 - Modelo dirigido a planos. Separa e distingue as fases de especificação e desenvolvimento
- Desenvolvimento iterativo e incremental
 - Especificação, desenvolvimento e validação são intercaladas
 - Pode ser dirigido a planos ou ágil

Modelos de Processos

- Na prática, a maioria dos sistemas de grande porte são desenvolvidos usando um processo que incorpora elementos de mais de um modelo
- Por exemplo:
 - Combinar algumas das melhores características do modelo em cascata e dos modelos de desenvolvimento incremental/ágil
 - As partes do sistema que são bem compreendidas podem ser especificadas e desenvolvidas por meio de um processo baseado no modelo em cascata
 - É preciso ter informações sobre os requisitos essenciais do sistema para projetar uma arquitetura de software que dê suporte a esses requisitos
 - Os subsistemas dentro de um sistema maior podem ser desenvolvidos com diferentes abordagens

Modelo Cascata



Modelo Cascata

- Modelo linear, dirigido a planos e com as fases bem distintas
 - Análise e definição de requisitos
 - Projeto de sistema e software
 - Implementação e teste unitário
 - Integração e teste de sistema
 - Operação e manutenção

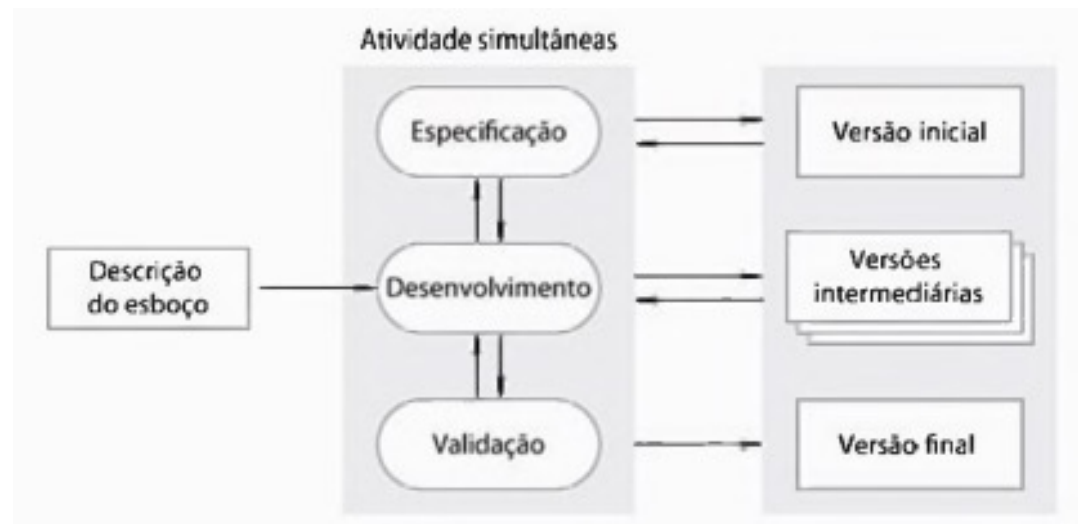
Modelo Cascata

- É consistente com outros modelos de processos de engenharia, e a documentação é produzida em cada fase do ciclo
- O processo torna-se visível e é possível monitorar o progresso de acordo com o plano de desenvolvimento
- Ainda usado quando os requisitos são bem compreendidos e pouco provavelmente venham a ser radicalmente alterados durante o desenvolvimento do sistema

Modelo Cascata

- Maior problema: a divisão inflexível do projeto em fases distintas dificulta o atendimento de mudanças nos requisitos
 - Poucos tipos de sistemas no mundo dos negócios possuem requisitos estáveis

Iterativo e Incremental



Iterativo e Incremental

- O custo de acomodar as mudanças nos requisitos do cliente é reduzido
 - A quantidade de análise e documentação geradas é menor em relação ao Modelo Cascata
- É mais fácil obter feedback dos clientes sobre o desenvolvimento que foi feito
 - Os cliente podem comentar nas demonstrações do software e acompanham o que está sendo implementado
- É possível obter entrega e implementação rápida de um software útil ao cliente, mesmo se toda a funcionalidade não for incluída
 - Os cliente obtém retorno da utilização do software antes em relação ao Modelo Cascata

Iterativo e Incremental

- O processo como um todo não é visível
 - Os gerentes necessitam de entregas regulares para medir o progresso. Se o sistema é desenvolvido rapidamente, não é economicamente viável produzir documentos que reflitam cada versão do software
- A estrutura do sistema tende a se degradar com a adição dos novos incrementos
 - Ao menos que tempo e dinheiro sejam gastos na refatoração do software, mudanças frequentes tendem a corromper a sua estrutura. Cada novo incremento torna-se mais difícil e oneroso

Iterativo e Incremental

- Os problemas são particularmente críticos para os sistemas de vida-longa, grandes e complexos, nos quais várias equipes atuam
- Esses sistemas necessitam de um *framework* ou arquitetura estável, e as responsabilidades das diferentes equipes precisam ser claramente definidas. Isso deve ser planejado com antecedência, e não desenvolvido de forma incremental
- Como reflexão: é possível desenvolver um sistema de forma incremental e expô-lo aos comentários dos clientes, sem realmente entregá-lo e implantá-lo no ambiente do cliente?

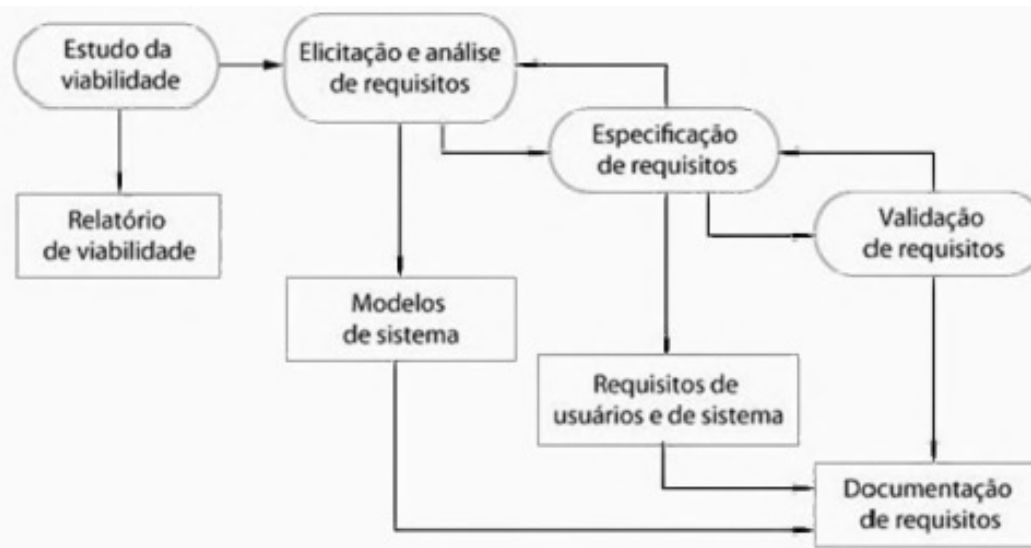
Atividades do processo

- O processo de desenvolvimento de software é uma sequência interconectada de atividades técnicas, de colaboração e gerenciais com o objetivo final de especificar, projetar, implementar e testar um sistema de software
- As quatro atividades básicas de especificação, desenvolvimento, validação e evolução são organizadas de maneiras distintas em diferentes processo de software

Especificação

- A especificação de software ou engenharia de requisitos é o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento
- Atividades do processo de especificação
 - *Estudo de viabilidade*: estimativa acerca da possibilidade de se satisfazerem as necessidades do usuário identificado usando-se tecnologias atuais de software e hardware
 - *Elicitação e análise de requisitos*. Esse é o processo de derivação dos requisitos do sistema
 - *Especificação de requisitos*. É a atividade de traduzir as informações obtidas durante a atividade de análise em um documento que defina um conjunto de requisitos
 - *Validação de requisitos*. Essa atividade verifica os requisitos quanto a realismo, consistência e completude

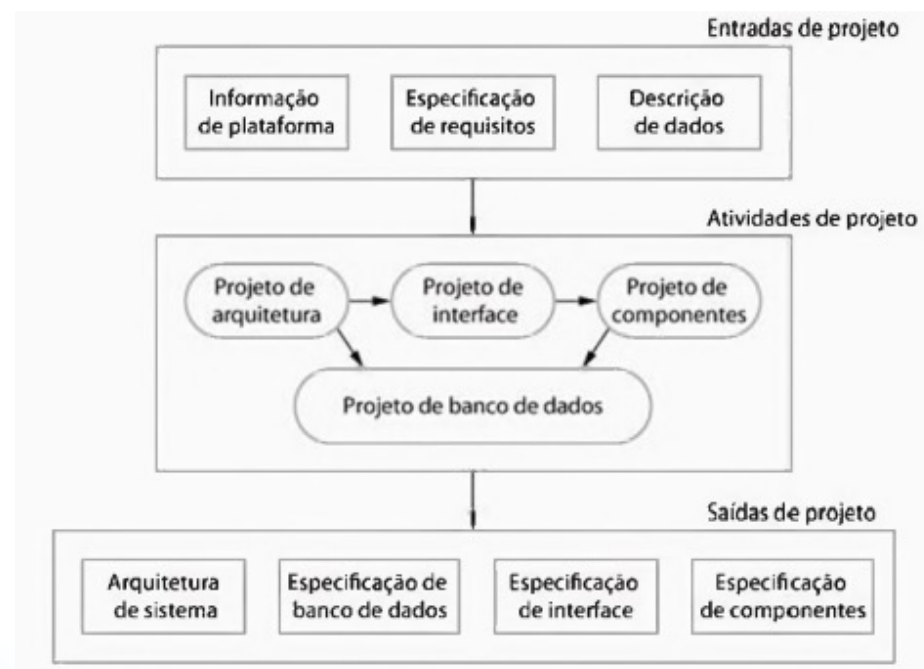
Engenharia de Requisitos



Desenvolvimento (projeto e implementação)

- É o processo de conversão de uma especificação de requisitos em um sistema executável
- Projeto de software
 - Descrição da estrutura do software, dos modelos e estruturas de dados usados pelo sistema, das interfaces entre os componentes do sistema e, às vezes, dos algoritmos usados
- Implementação
 - Transformação dessas estruturas em um programa executável
- As atividades de projeto e implementação de software são diretamente relacionadas e muitas vezes intercaladas

Projeto de software



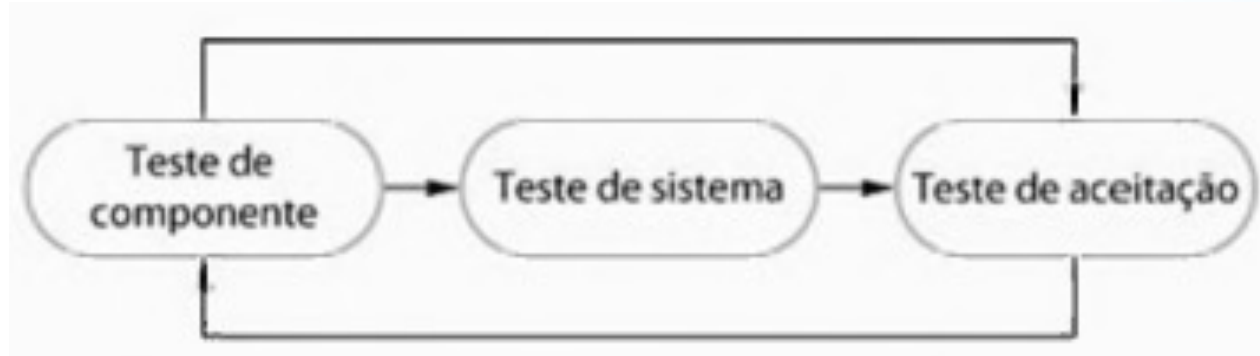
Projeto de software

- *Projeto de arquitetura*
 - Identificar a estrutura geral do sistema, os componentes principais (algumas vezes, chamados subsistemas ou módulos), seus relacionamentos e como eles são distribuídos
- *Projeto de interface*
 - Definir as interfaces entre os componentes do sistema
- *Projeto de componente*
 - *Projetar* o funcionamento dos componentes. Pode ser uma simples declaração da funcionalidade ou uma lista de alterações em um componente reusável
- *Projeto de banco de dados*
 - Projetar as estruturas de dados e como eles devem ser representados em um banco de dados

Validação

- Verificação e validação (V&V), tem a intenção de mostrar que um software se adequa a suas especificações ao mesmo tempo que satisfaz as especificações do cliente
- Pode envolver processos de verificação, como inspeções e revisões em cada estágio do processo de software
- Devido à predominância dos testes, a maior parte dos custos de validação incorre durante e após a implementação
- Teste de programa, em que o sistema é executado com dados de testes simulados, é a principal técnica de validação

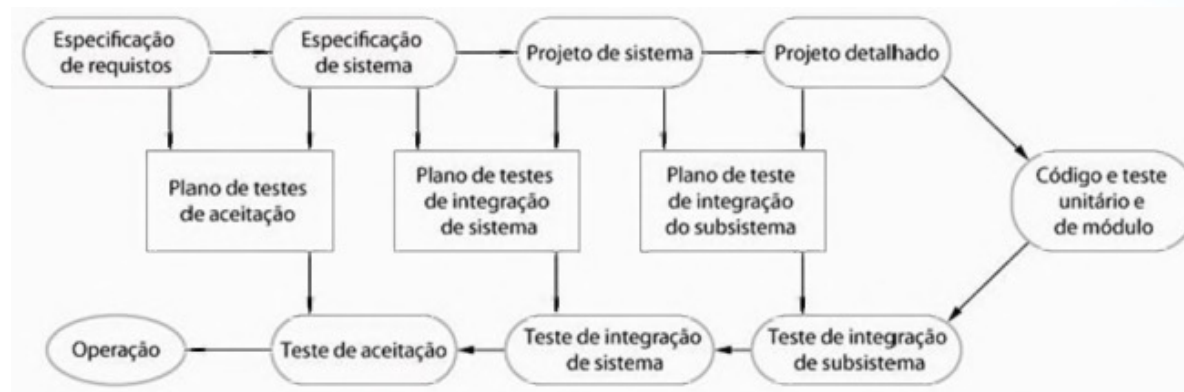
Estágios de teste



Estágios de teste

- Teste de desenvolvimento ou de componentes
 - Os componentes individuais são testados de forma independente
- Teste de sistemas
 - Componentes do sistema são integrados para criar um sistema completo e ser testado
- Teste de aceitação
 - Teste com dados fornecidos pelo cliente. Podem revelar erros e omissões na definição dos requisitos do sistema, pois muitas vezes os dados reais exercitam o sistema de formas diferentes dos dados de teste

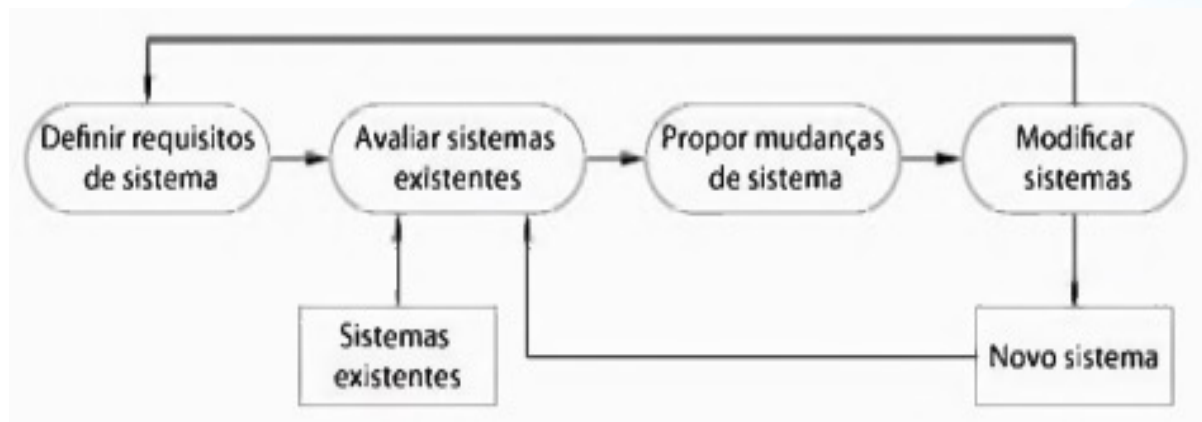
Estágios de teste



Evolução de software

- Por natureza, o software é flexível e pode mudar
- Os requisitos do sistema mudam, ao mesmo tempo que o negócio que adquiriu o sistema responde a pressões externas e mudam as prioridades de gerenciamento
- Embora existe uma distinção entre desenvolvimento e evolução, isso é cada vez menos relevante na medida em que mais e mais sistemas são totalmente novos

Evolução de software



Referências

