



*Serviço Nacional de Aprendizagem Industrial*  
**PELO FUTURO DO TRABALHO**

SÉRIE TI - SOFTWARE

# LÓGICA DE PROGRAMAÇÃO

FUNDAMENTOS  
DE SOFTWARE



## **CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI**

*Robson Braga de Andrade*  
Presidente

### **GABINETE DA PRESIDÊNCIA**

*Teodomiro Braga da Silva*  
Chefe do Gabinete - Diretor

### **DIRETORIA DE EDUCAÇÃO E TECNOLOGIA - DIRET**

*Rafael Esmeraldo Lucchesi Ramacciotti*  
Diretor de Educação e Tecnologia

### **SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL - SENAI**

*Robson Braga de Andrade*  
Presidente do Conselho Nacional

### **SENAI – Departamento Nacional**

*Rafael Esmeraldo Lucchesi Ramacciotti*  
Diretor-Geral

*Julio Sergio de Maya Pedrosa Moreira*  
Diretor-Adjunto

*Gustavo Leal Sales Filho*  
Diretor de Operações



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

SÉRIE TI - SOFTWARE

# LÓGICA DE PROGRAMAÇÃO

FUNDAMENTOS DE SOFTWARE



© 2021. SENAI – Departamento Nacional

© 2021. SENAI – Departamento Regional de Santa Catarina

A reprodução total ou parcial desta publicação por quaisquer meios, seja eletrônico, mecânico, fotocópia, de gravação ou outros, somente será permitida com prévia autorização, por escrito, do SENAI.

Esta publicação foi elaborada pela equipe de Educação a Distância do SENAI de Santa Catarina, com a coordenação do SENAI Departamento Nacional, para ser utilizada por todos os Departamentos Regionais do SENAI nos cursos presenciais e a distância.

**SENAI Departamento Nacional**

Unidade de Educação Profissional e Tecnológica - UNIEP

**SENAI Departamento Regional de Santa Catarina**

Gerência de Educação

**SENAI**

Serviço Nacional de  
Aprendizagem Industrial  
Departamento Nacional

**Sede**

Setor Bancário Norte • Quadra 1 • Bloco C • Edifício Roberto  
Simonsen • 70040-903 • Brasília – DF • Tel.: (0xx61) 3317-  
9001 Fax: (0xx61) 3317-9190 • <http://www.senai.br>

# Lista de Ilustrações

---

Figura 1 - Processo de desenvolvimento do software .....	10
Figura 2 - Ciclo de vida de um software .....	14
Figura 3 - Modelo Cascata .....	16
Figura 4 - Modelo Prototipagem .....	17
Figura 5 - Modelo concorrente .....	18
Figura 6 - Modelo incremental .....	19
Figura 7 - Modelo de desenvolvimento ágil .....	20
 Tabela 1 - Categorias de tamanho de softwares .....	 12



# Sumário

---

Fundamentos de Software.....	9
Apresentação .....	9
Definição .....	9
Estrutura .....	10
Histórico de Desenvolvimento do Software.....	10
Características de Um Software .....	11
Aplicações dos Softwares .....	12
Desenvolvimento de Software .....	13
Processo de Software .....	13
Ciclo de Vida/Definição e Importância .....	14
Aplicação na Indústria .....	21
Exemplo .....	21
Palavra do Docente.....	23
Referências.....	24





# Fundamentos de Software



## APRESENTAÇÃO

Seja bem-vindo ao conteúdo de Fundamentos de Software!

Existem pessoas que costumam associar o termo software apenas com o código dos programas de computador, porém esta é uma visão muito restritiva. Os softwares vão além dessa perspectiva, pois, além da parte de programação, os dados, as configurações relacionadas e toda a parte de documentação técnica devem ser considerados como partes integrantes de um software.

Entre as tecnologias conhecidas, sem sombra de dúvida, o desenvolvimento de software é uma das que está em alta no mundo corporativo.

Já em 1950, foi previsto que os programas seriam indispensáveis para as empresas, bem como para a ciência e a engenharia. Além disso, pressupunha-se que softwares seriam fundamentais para o processo de desenvolvimento de tecnologias inovadoras (como a engenharia genética), e realmente isso aconteceu.

O desenvolvimento de software começa quando um profissional, ou um time, o concebe e o constrói, e prossegue com fases posteriores de manutenção e aperfeiçoamento ao longo do tempo. No universo do desenvolvimento de software, há programas que são executados em computadores de diversos tamanhos e arquiteturas.

Dedique-se a esse incrível universo e bons estudos!

## DEFINIÇÃO

Os softwares são programas que podem rodar em computadores, tablets, notebooks e outros dispositivos eletrônicos. Além do código fonte em suas diversas formas, também a documentação associada faz parte do processo de desenvolvimento de um produto de software.

Software, de maneira mais simples, também pode ser definido como um conjunto modificável de comandos que permitem ao computador a realização de diversas tarefas.

Software é um produto conceitual e lógico do sistema de computação.

## ESTRUTURA

### HISTÓRICO DE DESENVOLVIMENTO DO SOFTWARE

A linha do tempo a seguir mostra as principais fases (comumente chamadas de eras) relacionadas à evolução do software ao longo dos anos.

#### Processo de Desenvolvimento do Software

##### • Histórico da Evolução do Software:

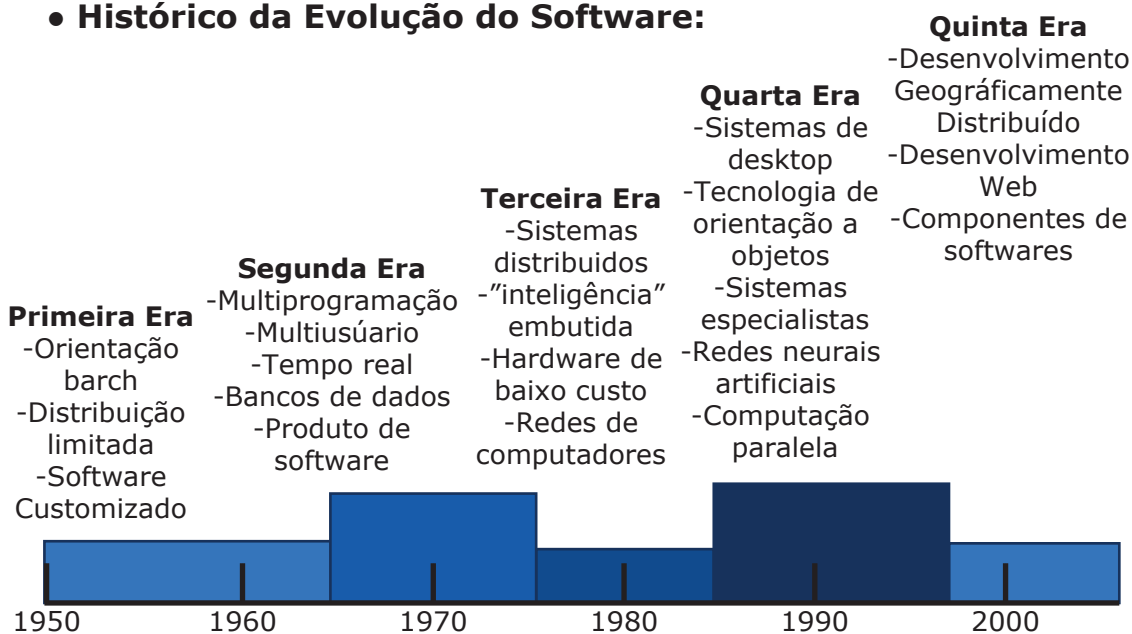


Figura 1 - Processo de desenvolvimento do software  
Fonte: Fernandes-Filho (2009)

**a) Primeira Era** – Os primeiros programas eram desenvolvidos em *batch* (execução sequencial sem intervenção humana; a não ser para iniciar o processo). Sua distribuição era muito limitada devido à dificuldade com recursos. Nesse momento, os softwares eram customizados, pois ainda não havia softwares de uso geral que pudessem ser comercializados.

**b) Segunda Era** – Os sistemas computacionais passaram a suportar uma execução de múltiplos programas simultaneamente (sistemas multiprogramáveis). Também passou a ser possível o uso de sistemas por múltiplos usuários (multiusuários). Essa era também se caracteriza pelo início da comunicação em tempo real, produtos de software e banco de dados.

**c) Terceira Era** – Nessa era, os programas passam a ser distribuídos com inteligência embutida, com hardware mais barato e com redes de computador.

**d) Quarta Era** – Sistemas de desktop, primeiras plataformas com orientação a objetos, sistemas especialistas, nascimento das redes neurais artificiais e computação paralela.

**e) Quinta Era** – Desenvolvimento geograficamente distribuído, rompante do desenvolvimento web e componentes de software.

## CARACTERÍSTICAS DE UM SOFTWARE

Conheça agora as características principais de um software e o que distingue o software (lógico) do hardware (físico). O software não sofre desgaste, mas pode se desatualizar com o tempo. Outra característica do software é que boa parte é feita sob medida.

No processo de desenvolvimento de software, é comum haver uma sequência de procedimentos que resultam na elaboração de diversos documentos. Um dos principais documentos é fruto da análise de requisitos e irá determinar as condições para que um software seja aceito.

A seguir, serão apresentadas algumas características adicionais dos softwares. Acompanhe.

**a) Invisibilidade** – Software é invisível por natureza.

**b) Complexidade** – Software é um dos mais complexos produtos construídos por seres humanos.

**c) Mutabilidade** – O software está sempre mudando (em evolução permanente, tanto no que se refere à sua forma de construção quanto pelo fato de os próprios produtos de software trazerem aperfeiçoamentos a cada nova versão).

**d) Conformidade** – Dependendo do ambiente para o qual um software é desenvolvido, é o programa que precisa se adaptar aos requisitos do ambiente.

Na tabela a seguir, é possível verificar uma classificação de software a partir do seu tamanho em linhas de código. Cada linha corresponde a uma categoria e apresenta o tamanho da equipe de desenvolvedores necessária para realizar a codificação (programação) do software, bem como o tempo de médio de desenvolvimento e, por último, o tamanho do programa.

### CATEGORIAS DE TAMANHO DE SOFTWARE

Categoria	Tamanho da Equipe	Duração	Tamanho da Fonte (linhas de código)
Trivial	1	1-4 semanas	500
Pequeno	1	1-6 meses	1000 a 2000
Médio	2-5	1-2 anos	5 mil a 50 mil
Grande	5-20	2-3 anos	50 mil a 100 mil
Muito grande	100-200	4-5 anos	1 milhão
Extremamente grande	2000-5000	5-10 anos	1 a 10 milhões

Ex: O Win 95; teve 11 milhões de linhas e 200 programadores  
O Netscape teve 3 milhões de linhas e 120 programadores.

Tabela 1 - Categorias de tamanho de softwares  
Fonte: Auxiliadora (s.d.)

## APLICAÇÕES DOS SOFTWARES

As aplicações de softwares são múltiplas. Entretanto, normalmente cada software resolve um conjunto de problemas bem definidos em uma determinada área de conhecimento, um domínio. Portanto, para que seja possível o desenvolvimento de um software, é importante identificar qual o domínio e levantar os requisitos. Quanto mais profundo for o levantamento de requisitos, melhor tende a ser o resultado.

Pode-se dividir as aplicações de software em:

- a) Software Básico** – São programas que dão apoio a outros softwares. Sua principal característica é interação com o hardware, assim proporcionando compartilhamento de recursos e distribuição para multiusuários e interfaces. Exemplo: sistemas operacionais e drivers de dispositivos.
- b) Software de Tempo Real** – Esses programas fazem o controle e a análise de eventos externos em tempo real. Exemplo: jogos iterativos de computador.
- c) Software Comercial** – É disparado o tipo de programa mais conhecido; são aplicações de uso comercial. Exemplo: pacote Office, cuja proprietária é a Microsoft.
- d) Software Científico e de Engenharia** – Sua principal característica é executar algoritmos de processamento de base numérica com elevada precisão. Exemplo: sistemas de manufatura.

- e) Software Embutido** – É o software integrado a produtos eletrônicos. Exemplo: software embutido em equipamento eletrônico, como micro-ondas.
- f) Software de Computador Pessoal** – São softwares para uso no computador pessoal. Exemplo: planilhas eletrônicas e processadores de textos.
- g) Software de Inteligência Artificial** – Sua característica é o uso de algoritmos não numéricos para a resolução de problemas complexos. Exemplo: sistemas de reconhecimento de voz.

## DESENVOLVIMENTO DE SOFTWARE

Os processos de software abrangem um conjunto de três elementos fundamentais: métodos, ferramentas e procedimentos, que servem para projetar, construir e manter grandes sistemas de software de forma profissional.

### MÉTODOS

Os métodos servem para mostrar as especificidades de “como fazer” para desenvolver um software. No como fazer, entram os elementos de projeto, a definição, o planejamento, a análise de requisitos, a estrutura dos dados, os algoritmos, a codificação e o teste de manutenção.

### FERRAMENTAS

As ferramentas dão apoio à automatização dos métodos. Existem atualmente várias ferramentas para sustentar cada método. Pode-se citar, por exemplo, as ferramentas de apoio à edição de código, como o VSCode ou o Eclipse, e as ferramentas para gestão de repositórios de software, como o GitLab.

### PROCEDIMENTOS

Os procedimentos são uma “junção” entre os métodos e as ferramentas citados anteriormente. Eles definem a ordem ou sequência mais adequada para a aplicação de determinado método, utilizando-se determinada ferramenta.

## PROCESSO DE SOFTWARE

De forma geral, o processo de software possui o seguinte conjunto de atividades:

- a) Análise de Requisitos:** fase em que o domínio do problema é estudado e os requisitos para a solução são levantados.

- b) Projeto e Especificação** – Nesta fase, os requisitos previamente levantados dão origem ao projeto do sistema, o qual inclui documentos que indicam como o software deverá ser construído.
- c) Implementação** – Fase em que a construção do software propriamente dito ocorre. É aqui que são gerados os artefatos de software codificados em linguagem de programação.
- d) Verificação e Validação** – Fase em que o software implementado é confrontado com os requisitos para validá-lo e verificar se pode ser aceito e entregue.
- e) Implantação** – Fase em que o software é instalado, implantado ou disponibilizado para uso. Diz-se, então, que o software entra “em produção”.
- f) Manutenção** – Fase em que o software, já “em produção”, passa a ser mantido, comumente por efeito de um período de garantia ou por um contrato de serviços. Nesta fase, são corrigidos defeitos (bugs) que eventualmente passaram despercebidos na fase de verificação.

## CICLO DE VIDA/DEFINIÇÃO E IMPORTÂNCIA

A Engenharia de Software é uma disciplina de engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, após sua entrada em produção. A meta principal da Engenharia de Software é desenvolver sistemas com boa relação custo-benefício (CARVALHO, 2021).

## CICLO DE VIDA DO SOFTWARE

Steve McConnell (1997) afirma que um modelo de ciclo de vida de software é uma representação que descreve todas as atividades que tratam da criação de um produto de software. Dessa forma, entende-se como ciclo de vida de softwares o que se refere às fases pelas quais um sistema de software atravessa desde sua concepção até sua retirada de produção.

Observe as fases do ciclo de vida de um software.



Figura 2 - Ciclo de vida de um software  
Fonte: da Autora (2021)

Conforme a ilustração, o ciclo de vida de um software possui 4 grandes áreas: definição, desenvolvimento, operação e retirada. Acompanhe cada uma delas detalhadamente.

- a) Definição** – É momento do reconhecimento da natureza do software. Entre as etapas da definição, têm-se o levantamento da situação atual, o estudo de viabilidade e de levantamento de custos, a proposta de desenvolvimento e o levantamento de requisitos.
- b) Desenvolvimento** – Nesta etapa, são elaborados os projetos, realizadas as implementações, as verificações e as validações.
- c) Operação** – É o momento em que acontece a distribuição, a entrega, a instalação e a configuração. Nesta fase, é comum a empresa desenvolvedora realizar suporte e manutenção do sistema.
- d) Retirada** – Esta é uma fase em que se retira o software de produção, normalmente por tornar-se obsoleto. Em alguns casos, é necessária a colaboração da empresa desenvolvedora para que dados e configurações possam ser migrados para um novo sistema.

## MODELOS DE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Os modelos definem os grupos de processos a ser implementados durante o desenvolvimento de um produto de software.

## MODELO CASCATA

O modelo denominado Cascata (*waterfall*) tem como característica principal o desenvolvimento sequencial (em cascata) e está entre os mais antigos (em desuso atualmente).

Na imagem, observe que cada processo sucede a um processo anterior de forma sequencial (em cascata).

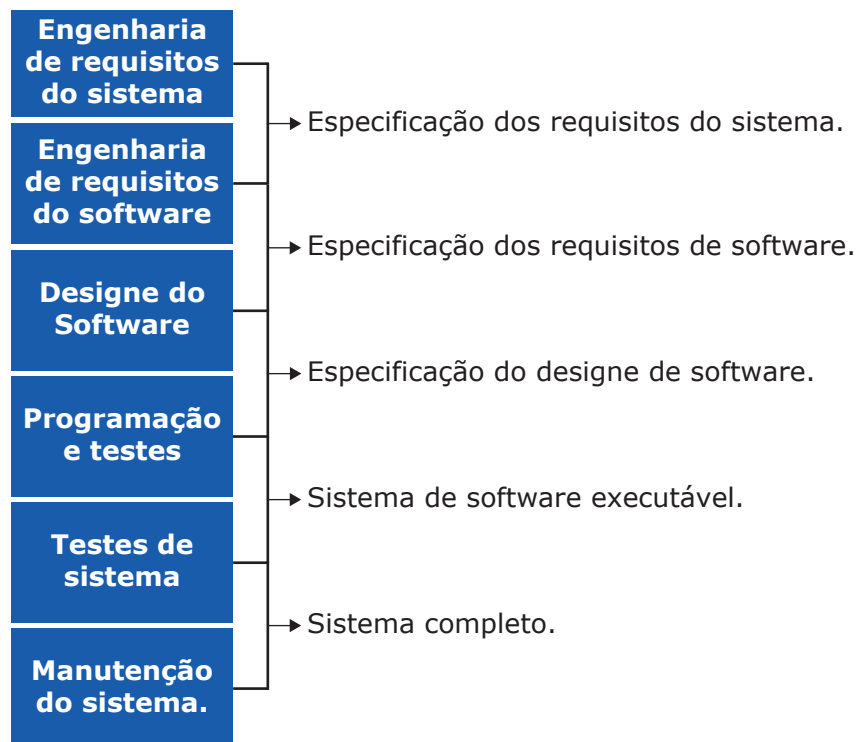


Figura 3 - Modelo Cascata  
Fonte: Carvalho, Batista e Santos (2012)



## MODELO PROTOTIPAÇÃO

Neste modelo, o desenvolvimento é realizado em torno de ciclos de prototipação, muito usado quando o cliente não especificou muito bem os requisitos, além de simular a aparência e possibilitar exercitar algumas funcionalidades básicas desde os primeiros ciclos. A imagem a seguir mostra as fases dos ciclos de prototipação: coleta de requisitos, projeto, construção do protótipo, avaliação e refinamento.



Figura 4 - Modelo Prototipagem  
Fonte: Gomes (2014)

## MODELO CONCORRENTE

A principal característica do modelo concorrente é a organização das atividades em simultâneo, ou seja, algumas atividades acontecem em paralelo com outras, mas em diferentes estados. A imagem mostra como funciona o modelo concorrente.

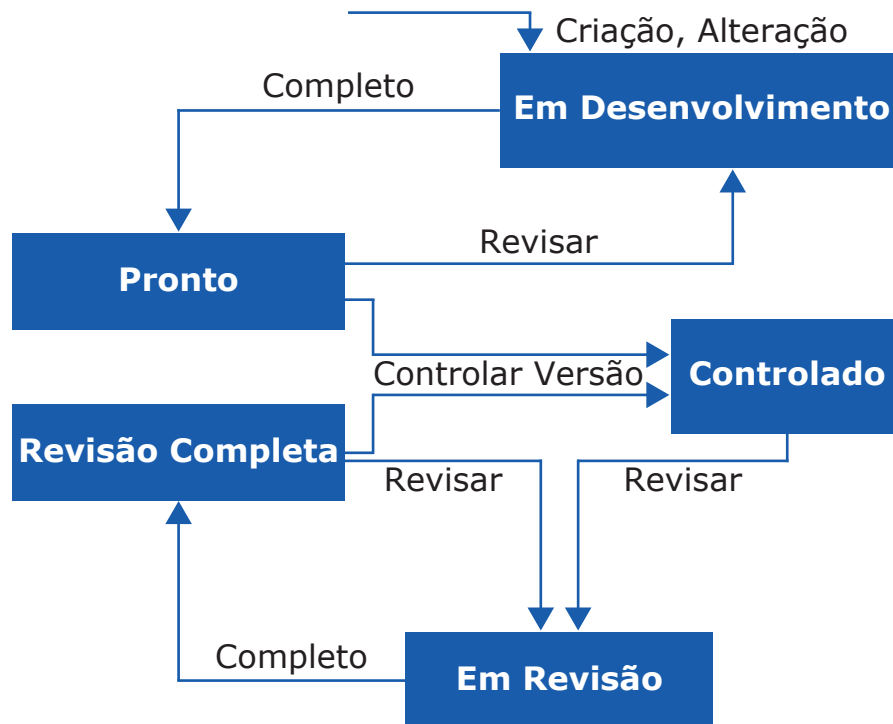


Figura 5 - Modelo concorrente  
Fonte: Reis (2001)

## MODELO INCREMENTAL

A principal característica do modelo incremental é sua formação a partir da mescla entre o modelo cascata e o prototipação. Cada ciclo, também chamado de iteração, gera um incremento, e cada incremento é uma entrega, conforme a imagem a seguir.

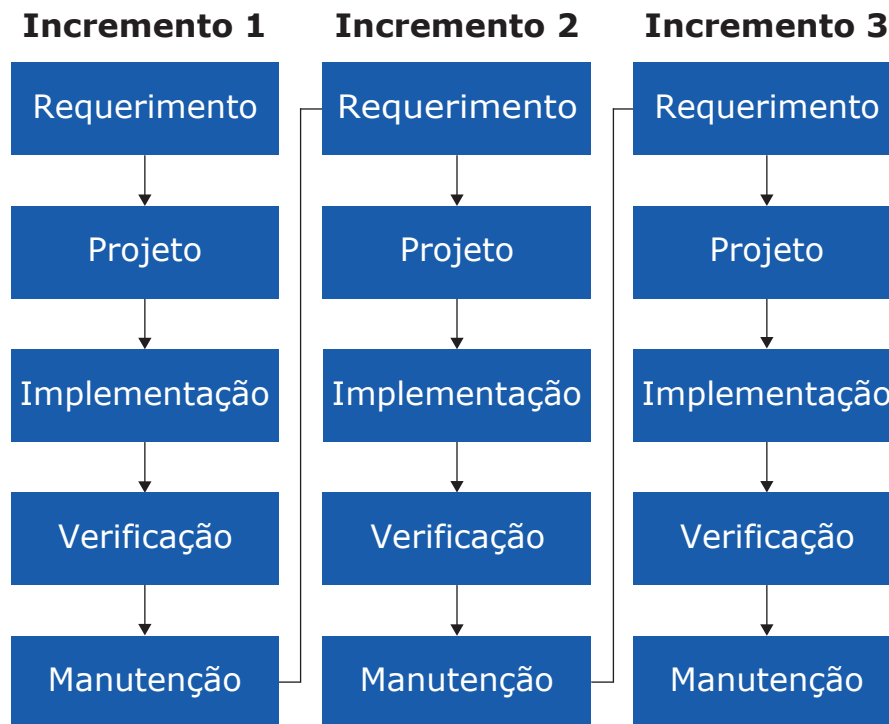


Figura 6 - Modelo incremental  
Fonte: Aerochimps (2015)

## MODELO DE DESENVOLVIMENTO ÁGIL

As principais características do modelo de desenvolvimento ágil são: uma forma mais acelerada no desenvolvimento, com estratégia de fases curtas de desenvolvimento (1 a 4 semanas); comunicação em tempo real com os stakeholders (partes interessadas); comunicação face a face ou verbal (documentada); e reuniões de planejamento e validação.

Um dos modelos de desenvolvimento ágil mais utilizados é o Scrum. A imagem a seguir mostra o fluxo do Scrum em 4 fases principais: iniciação, ideação, criação e evolução. Hoje, o Scrum é modelo de desenvolvimento mais aplicado em indústrias de software.

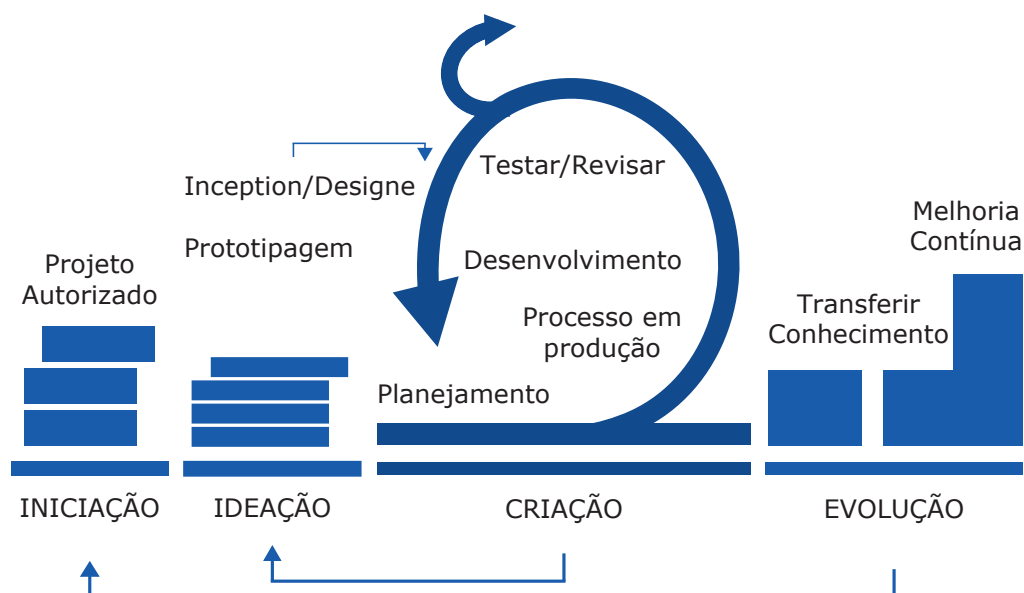
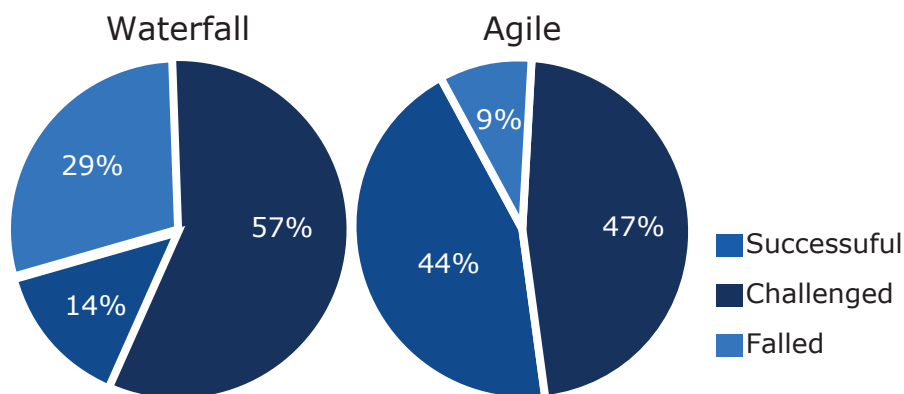


Figura 7 - Modelo de desenvolvimento ágil  
Fonte: Lisi (2018)

## APLICAÇÃO NA INDÚSTRIA

As indústrias de software têm adotado o modelo desenvolvimento ágil como um modelo inovador e bastante produtivo. Em 2014, a empresa VersionOne fez uma pesquisa com diversas empresas e detectou que 94% dos entrevistados utilizam metodologias ágeis para desenvolvimento de software. A imagem abaixo mostra um gráfico comparativo da taxa de sucesso entre aqueles que utilizam metodologias tradicionais (cascata) em relação aos modelos ágeis.



Fonte: Standish Group Chaos Report 2012 *apud* Ettinger (2015)

## EXEMPLO

### Sony

A multinacional Sony, que fabrica equipamentos eletrônicos em larga escala, implantou a metodologia ágil para auxiliar na eficiência de seus processos complexos. As áreas de projetos e desenvolvimento de software utilizam o Scrum como modelo ágil.

Houve um incentivo à cultura ágil na empresa, convencendo os times de que a mudança era essencial para competitividade no mercado e saúde da empresa.

Atualmente, a empresa tem gerenciamento ágil de alto nível e a equipe de projetos da Sony é reconhecida como uma das melhores do mundo nesse segmento.

Para conhecer as 8 metodologias ágeis mais usadas no mercado, acesse <https://neilpatel.com/br/blog/metodologia-agil/> ou aproxime o celular do *QR Code* a seguir.



**SAIBA  
MAIS**



\*\*\*

## PALAVRA DO DOCENTE

---

A construção de um software bem-sucedido, por meio de um modelo de processo que seja ágil e adaptável, tende a levar a um resultado de qualidade e eficácia. Diante disso, milhares de pessoas estão simplificando suas rotinas e processos, o que resulta em maior qualidade de vida. Um simples exemplo disso é o fato de que, hoje, uma pessoa com acesso à internet pode fazer suas compras e pagar suas contas via *internet banking*. Muitos produtos estão disponíveis em e-commerces que utilizam softwares criados para esses fins, evitando espera em filas de banco, o stress do trânsito, além dos gastos com combustível ou transporte, de modo que o tempo poupado pode ser usado para outras coisas. O produto do desenvolvimento de software é a simplificação das transações e a informação resultante que, de algum modo, tornam melhor o mundo do usuário.

## REFERÊNCIAS

---

AEROCHIMPS. **Por que desenvolver iterativamente?** Plataforma Medium, 2015. Disponível em: <https://medium.com/@Aerochimps/por-que-desenvolver-iterativamente-54b9128ac43b>. Acesso em: 28 jul. 2021.

AUXILIADORA, M. Categorias de Tamanhos de Softwares [Apresentação - Slide 70]. In: \_\_\_\_\_. **Introdução à Engenharia de Software**, s.d. Disponível em: <https://slideplayer.com.br/slide/3153647/>. Acesso em: 28 jul. 2021.

CARVALHO, D. **Aula 00/SEFAZ-AL (Auditor de Finanças de Controle) Engenharia de Software - 2021** (Pós-Edital). São Paulo: Estratégia Concursos, 2021. Disponível em: <https://t.ly/yBwO>. Acesso em: 28 jul. 2021.

CARVALHO, F. S.; BATISTA, D. N.; SANTOS, M. **Modelo Cascata, Linear ou Clássico**. Blog da Disciplina Engenharia de Software. Colegiado de Ciência da Computação, Universidade Estadual do Sudoeste da Bahia, 2012. Disponível em: <http://engenhariadesoftwareuesb.blogspot.com/2012/12/fffrrrrr.html>. Acesso em: 28 jul. 2021.

ETTINGER, D. **O crescimento das metodologias ágeis nas organizações**. Site pessoal, 2015. Disponível em: <https://danielettinger.com/2015/11/10/o-crescimento-das-metodologias-ageis-nas-organizacoes/>. Acesso em: 28 jul. 2021.

FERNANDES-FILHO, C. A. Processo de desenvolvimento do software [Imagem JPG]. In: \_\_\_\_\_. **Histórico da Evolução do Software**. Blog Dúvidas de Programação, 2009. Disponível em: <http://duvidasdeprogramacao.blogspot.com/2009/04/introducao-engenharia-de-softwares.html>. Acesso em: 28 jul. 2021.

GOMES, J. L. S. **Aula 1 - Conceitos gerais aplicados no ciclo de vida do software e modelos de processo**. Slides de apresentação – Disciplina Introdução ao Desenvolvimento de Sistemas, Univale, 2014. Disponível em: <https://pt.slideshare.net/janynnegomes/aula-1-conceitos-gerais-aplicados-no-ciclo-de-vida-do-software-e-modelos-de-processo>. Acesso em: 28 jul. 2021.

LISI, E. **5 dicas para implementação da Metodologia Ágil**. Plataforma Lecom, 2018. Disponível em: <https://www.lecom.com.br/blog/5-dicas-da-metodologia-agil/>. Acesso em: 28 jul. 2021.

MCCONNELL, S. (Org.). **The OMG/CommerceNet Joint Electronic Commerce Whitepaper**, EC/97-06-09. Milford: Object Management Group, 1997.

REIS, C. **Caracterização de um Modelo de Processo para Projetos de Software Livre**. 2001. Exame de Qualificação (para Mestrado em Ciências da Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação. USP, São Carlos, 2001. Disponível em: <http://www.async.com.br/~kiko/quali/>. Acesso em: 28 jul. 2021.







*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**