



(ADS)
***2 Gerenciamento e
Projeto de Software***

Prof. Dr. Eliseu LS

O que seria mesmo um Projeto ?

PROJETO - Trata-se de um empreendimento único que deve apresentar um início e um fim claramente definidos e que, conduzido por pessoas possa atingir seus objetivos respeitando os parâmetros de prazo, custo e qualidade (MENEZES, 2001).

Trata-se de esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. Ou seja, um projeto é tudo aquilo que precisamos realizar para gerar algo novo: seja uma casa, um sistema informatizado, um estudo/pesquisa, um trabalho de conclusão de curso, uma contratação ou um compra importante. Lembre-se que ser “temporário” significa que os projetos devem ter um início e um término definidos; não significa de curta duração.

Gerência de Projetos

Um projeto têm que ser gerenciado em suas várias fases metodológicas, o controle da comunicação, da gestão de pessoas, da gestão do escopo, gestão de insumos ou recursos tecnológicos, gestão financeira, enfim um empreendimento é alcançado através de planejamento e esforço coletivo controlado por um gerente de projetos habilidoso.

Procede daí o termo cunhado "Gestão de Projetos" ou "Gerenciamento de Projetos", com direito até de certificações profissionais distribuídas por empresas credenciadas que habilitam certos profissionais no uso de diversas tecnologias e metodologias.

Softwares disponíveis no mercado

No mercado existem muitas ferramentas disponíveis, a lista a seguir contém algumas que podem ser encontradas para uso, a mais popular no mundo acadêmico é o MS-Project.

- **Microsoft Project**
- **Jira**
- **Trello**
- **Microsoft Planner**
- **CA Clarity**
- **Asana**
- **Monday**
- **Dynamics 365 Projects Operations (Antigo Dynamics 365 Project Service Automation -PSA)**
- **Bitrix 24**
- **Gantt Project**
- **Artia**

Responsabilidades do Gerente de Projeto

| TIPOS DE GERENCIAMENTO | DESCRIÇÃO |
|------------------------|---|
| LIDERANÇA | Dirige a integração e liderança entre as equipes do projeto |
| ESCOPO | Determina o alcance do projeto, os limites e as principais funções do projeto |
| TEMPO | Define e controla o cronograma das entregas. |
| CUSTOS | Determina os custos de cada etapa do projeto e a origem da verba. |
| QUALIDADE | Gerencia e monitora a gestão de qualidade do projeto conferindo cada etapa. |
| RECURSOS HUMANOS | provê a mão-de-obra necessária. |
| COMUNICAÇÕES | Estabelece paradigmas de comunicação entre os membros das equipes. |
| RISCOS | Monitora e controla os riscos do projeto. |
| SUPRIMENTOS | Provê os insumos necessários. |

NORMATIZAÇÕES

| | |
|-----------------------|---|
| ISO10006: 1997 | A ISO 10006:1997 (International Organization for Standardization) Organização internacional de normalização, é uma norma internacional europeia r que fornece diretrizes para a aplicação da gestão da qualidade em projetos. É aplicável a organizações que trabalham em projetos de complexidade variável, de pequeno ou grande porte, de curta ou longa duração, sendo um projeto individual fazer parte de um programa ou portfólio de projetos, em diferentes ambientes, e independentemente do tipo de produto/serviço ou processo envolvido. (Origem em Londres) |
| PRINCE2 | PRINCE2 (Projects in Controlled Environments) Projetos em Ambientes controlados, trata-se de um método estruturado de gerenciamento de projetos e programa de certificação de profissionais. (Origem no Reino Unido). |
| PMBOOK | A Guide to the Project Management Body of Knowledge (Guia PMBOK®). Guia para o conjunto de conhecimentos em gerenciamento de projetos, trata-se de uma estrutura de gerenciamento de projetos publicada pelo Project Management Institute (PMI). O Guia PMBOK® é o framework de gerenciamento de projetos mais utilizado no mundo. (Origem Montreal Canadá) |

CERTIFICAÇÕES

1. PMP: Project Management Professional
2. CAPM: Certified Associate in Project Management
3. CSM: Certified ScrumMaster
4. CompTIA Project+
5. PRINCE2 Foundation/PRINCE2 Practitioner
6. CPMP: Certified Project Management Practitioner
7. Associate in Project Management
8. MPM: Master Project Manager
9. PPM: Professional in Project Management
10. PMITS: Project Management in IT Security
11. Certified Project Director
12. IAPM Certified Project Manager

Fonte:

<https://itforum.com.br/noticias/12-certificacoes-de-gestao-de-projetos-para-ter-em-mente/> acessado em 21/08/2023;

NORMA USADA NO BRASIL

Para que seja possível garantir um software de qualidade e mais isento possível de erros, é imprescindível seguir algumas normas e padrões. Estes fazem parte de um conjunto de documentos elaborados por uma entidade denominada International Organization for Standardization (ISO), sendo conhecidos mundialmente como Normas ISO. Essa entidade desenvolveu uma norma denominada **ISO/IEC 12207**, especificamente para aplicação no desenvolvimento de software.

No Brasil, ela foi publicada pela Associação Brasileira de Normas Técnicas (ABNT) como **NBR ISO/IEC 12207**.

NBR ISO/IEC 12207 & CICLO DE VIDA

"Outro importante ponto presente na NBR ISO/IEC 12207 é a estruturação do ciclo de vida em diversos processos, que, por sua vez, são formados por um grande número de atividades, como ilustra a Figura 4.1." (ALVES, 2017, p.78)

CICLO DE VIDA - Pode ser é definido pelas diversas fases e etapas do processo de criação do projeto que resultará em um serviço ou produto. O Software pode ser encarado tanto como um produto e também como um serviço, para tanto existem diferentes formas de cobrança para o produto ou serviço software.

"RISCO TÉCNICO"

RISCO TÉCNICO: Nesse sentido, o risco técnico em software é compreendido como uma medida de probabilidade e severidade dos efeitos adversos essenciais ao processo de desenvolvimento de software que não suprem os requisitos planejados e as funções específicas (CHITTISTER e HAIMES, 1993; ZMUD, 1980).

QUANDO OCORRE ? Um risco técnico pode acontecer em qualquer das fases metodológicas na criação do software que vai desde a falta de verba, falta de profissionais qualificados, falha no tempo da entrega ou na forma de entrega do software, etc.

.

CAUSAS QUE AUMENTAM OS RISCOS TÉCNICOS

- 1- Falta de Monitoramento e Controle.
- 2- Falta de Sessões de Planejamento com o Time.
- 3– Gerente incompetente
- 4– Falta de Apoio Geral à Cultura de **Projetos**
- 5– Nenhuma Gestão de **Riscos**
- 6– Estimativas irreais, expectativas erradas do projeto
- 7– Falta de verba
- 8 - Comunicação Ruim
- 9 - Excesso de horas trabalhadas
- 10 - Falta de Escopo
- 11 - Cronograma mal feito ocasionando atrasos na entrega do software
- 12 - Falta de testes adequados para garantir a qualidade
- 13 - Pouca ou nenhuma manutenção

Fonte: <https://nerus.com.br/blog/processos/exemplos-riscos-projeto-software/>
acessado em 21/08/2023.



(ADS)
Modelos para Projetos
de Software

Prof. Dr. Eliseu LS

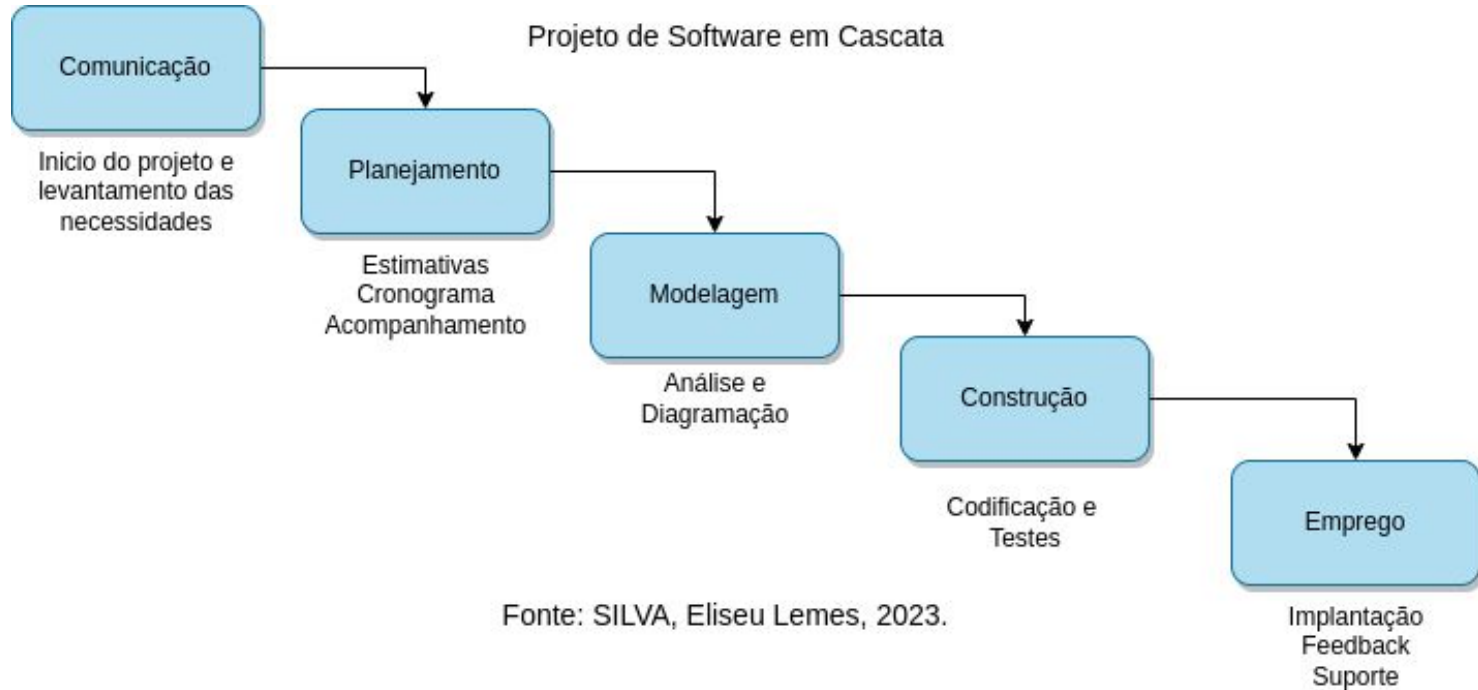
Modelo de Projeto de Software

Dos vários modelos existentes, alguns se destacam sendo que todos são assistidos pelas fases metodológicas já explicadas, porém, segundo Pressman (2011), cada um deles dá uma ênfase diferente a essas atividades e podem definir um fluxo de processo diferenciado.

Também segundo Bezerra (2007), parece ser um consenso na comunidade de desenvolvimento de software o **fato de que não existe o melhor processo de desenvolvimento**, aquele que melhor se aplica a todas as situações de desenvolvimento. Segundo o autor, cada processo tem suas particularidades em relação ao modo de arranjar e encadear as atividades de desenvolvimento.

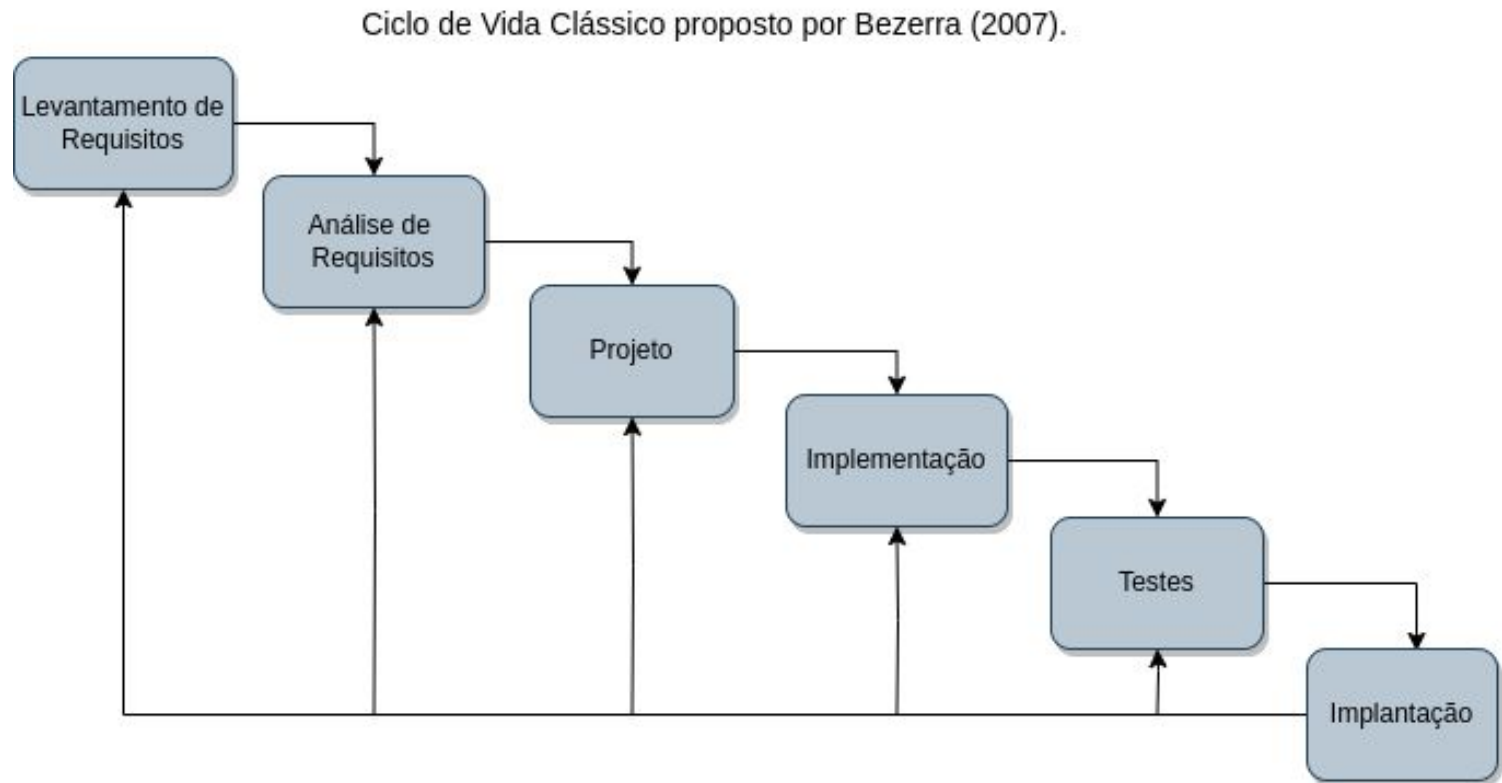
Modelo em Cascata (Clássico)

Processo tradicional de desenvolvimento de software também conhecido como linear ou sequencial é o mais comuns dos modelos utilizados no mercado de desenvolvimento hierarquizado.



Ciclo de vida - Bezerra (2007)

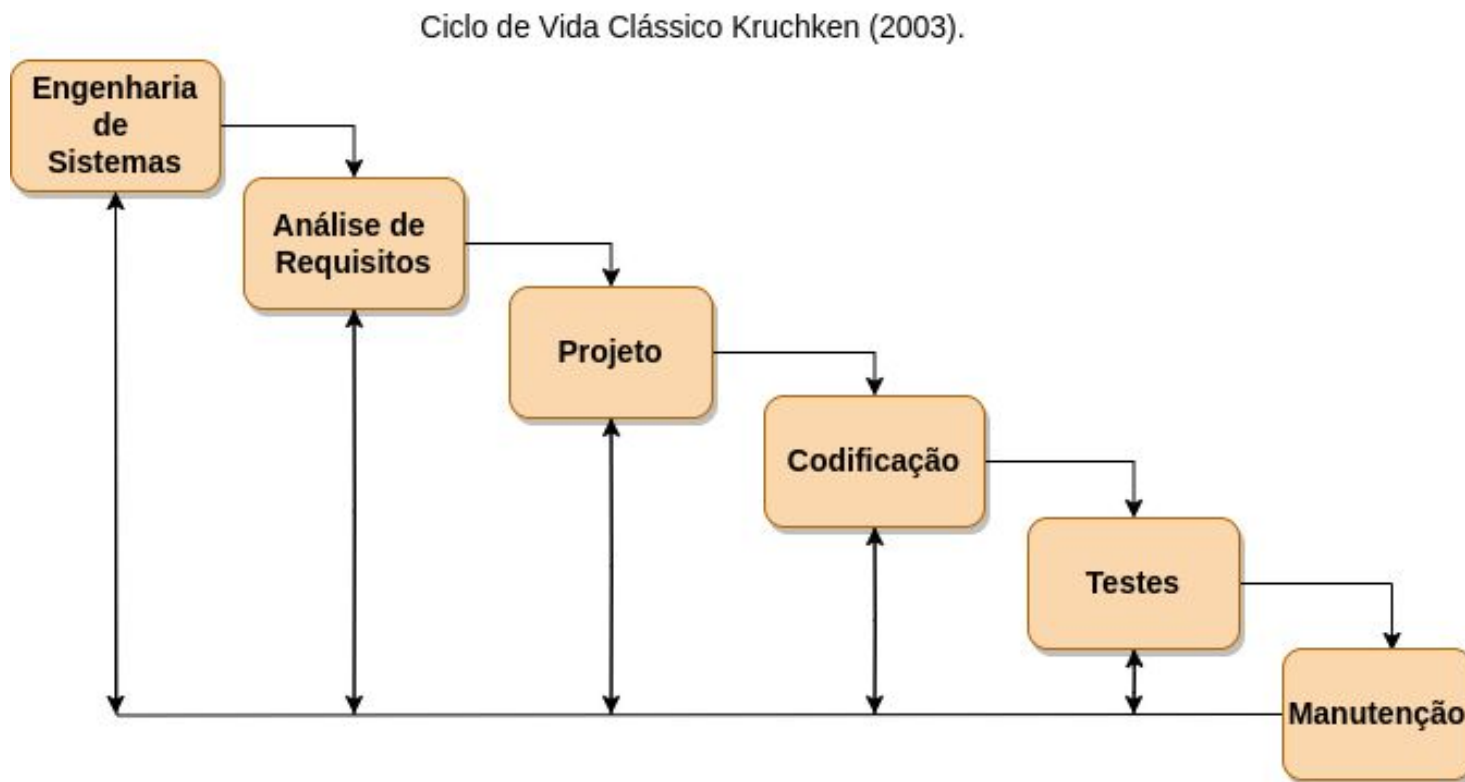
Segundo proposta de Eduardo Bezerra, o modelo do ciclo de vida clássico da engenharia de software é dividido em seis atividades, conforme a figura a seguir.



Fonte: SILVA, Eliseu Lemes, 2023.

Ciclo de vida - Kruchten (2003)

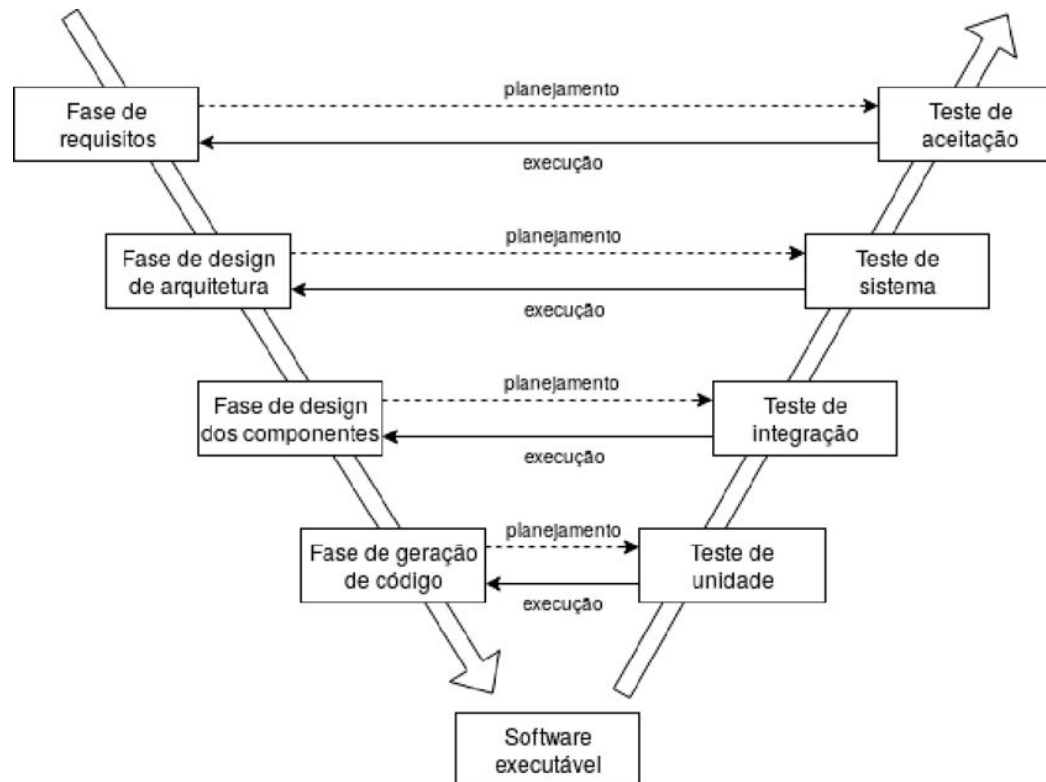
A proposta de Bezerra baseou-se na proposta de Kruchten (2003), basicamente são as mesmas fases que podem se desdobrar em repetições.



Fonte: SILVA, Eliseu Lemes, 2023.

Ciclo de Vida em V

Uma variação do modelo cascata, destacada por Pressman (2011) é o modelo V. Segundo o autor, à medida que a equipe de software desce em direção ao lado esquerdo do V, os requisitos básicos do problema são refinados em representações progressivamente cada vez mais detalhadas e técnicas do problema e de sua solução.



Vantagens & Desvantagens (Clássico)

Vantagens:

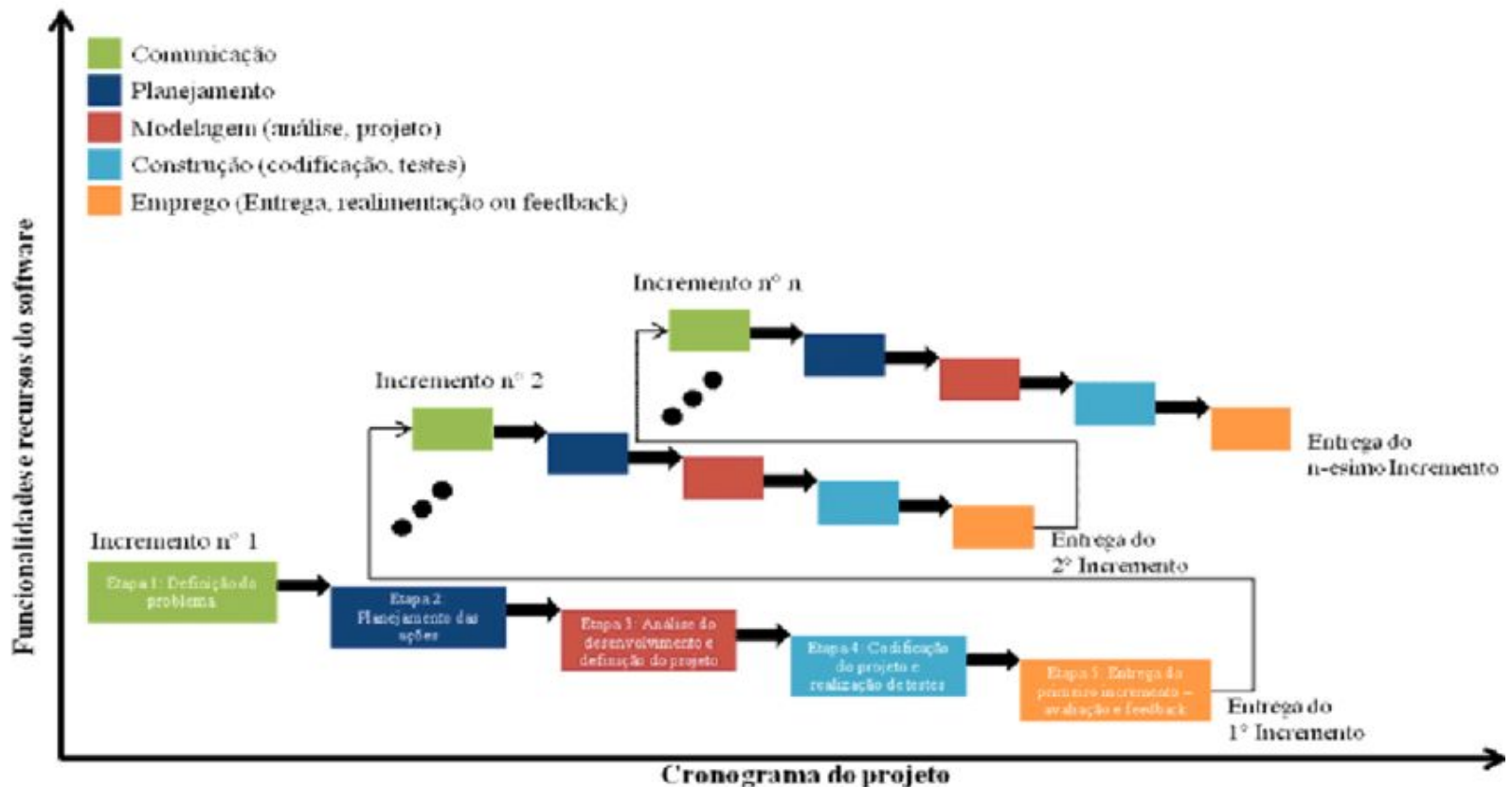
- Simplicidade na aplicação do processo
- Facilidade na gerência do processo de criação

Desvantagens:

- Projetos reais raramente seguem este fluxo sequencial.
- Dificuldades no entendimento de todas as necessidades dos usuários;
- Demora em apresentar resultados aos clientes.

Modelo Incremental

Este modelo foi proposto como uma alternativa ao modelo em cascata, aplicando-o iterativamente, tendo como objetivo a elaboração de um produto operacional a cada incremento. Os primeiros incrementos são versões simplificadas do produto final, mas oferecem capacidades que servem ao usuário, além de servir como uma plataforma de avaliação.



(Modelo Incremental)

VANTAGEM :

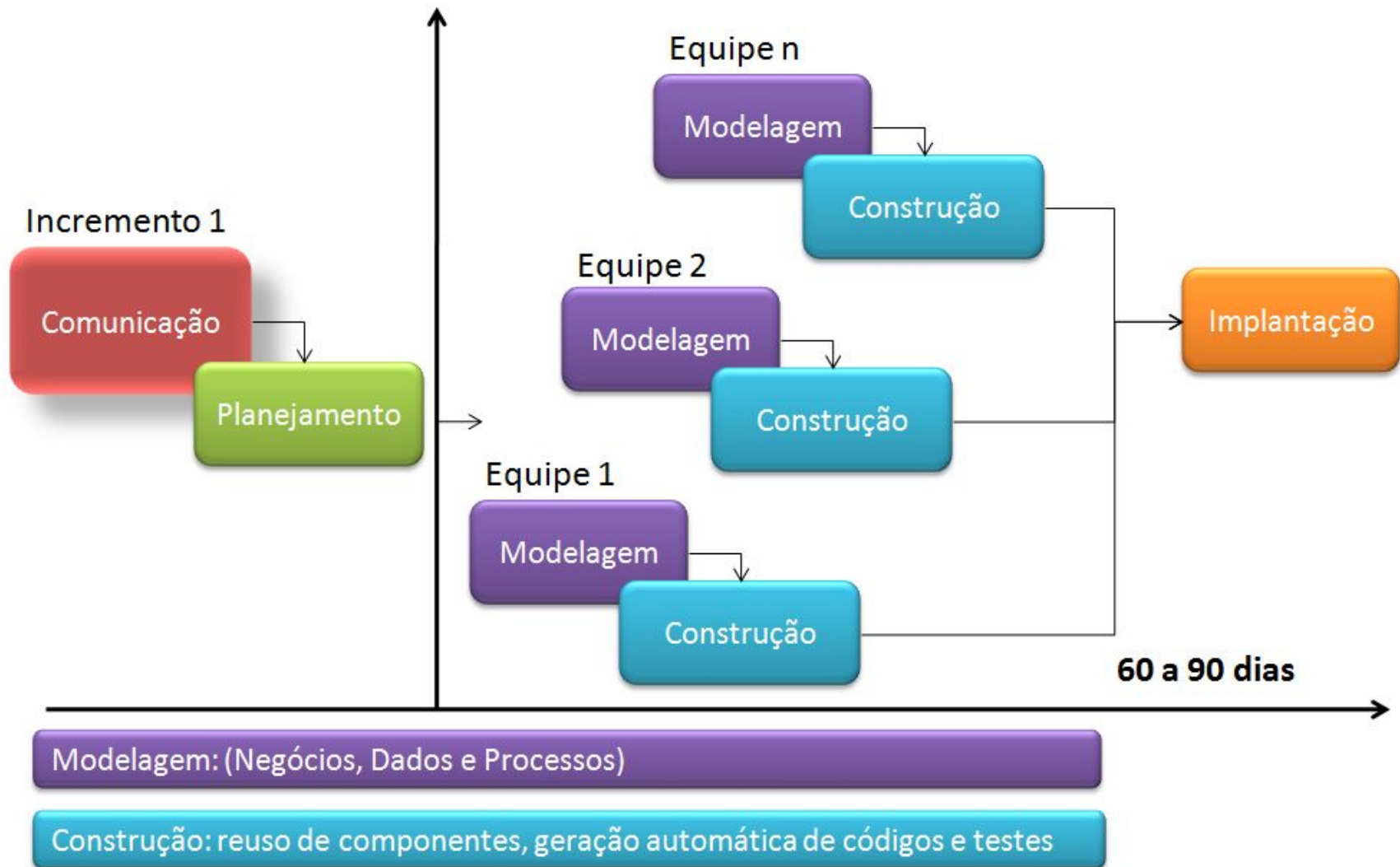
- Este modelo possui como vantagem o fato de apresentar constantemente novas versões aos usuários.
- Pode ser aplicado também quando não houver mão-de-obra disponível para uma implementação completa
- Pode ser aplicado quando for necessário gerenciar riscos técnicos.

Modelo RAD (Rapid Application Development)

Desenvolvimento Rápido de Aplicação **Trata-se de um modelo de processo de software incremental** que enfatiza um ciclo de desenvolvimento curto. É uma adaptação “de alta velocidade” do modelo em cascata, no qual o desenvolvimento rápido é conseguido com o uso de uma abordagem de construção baseada em componentes.

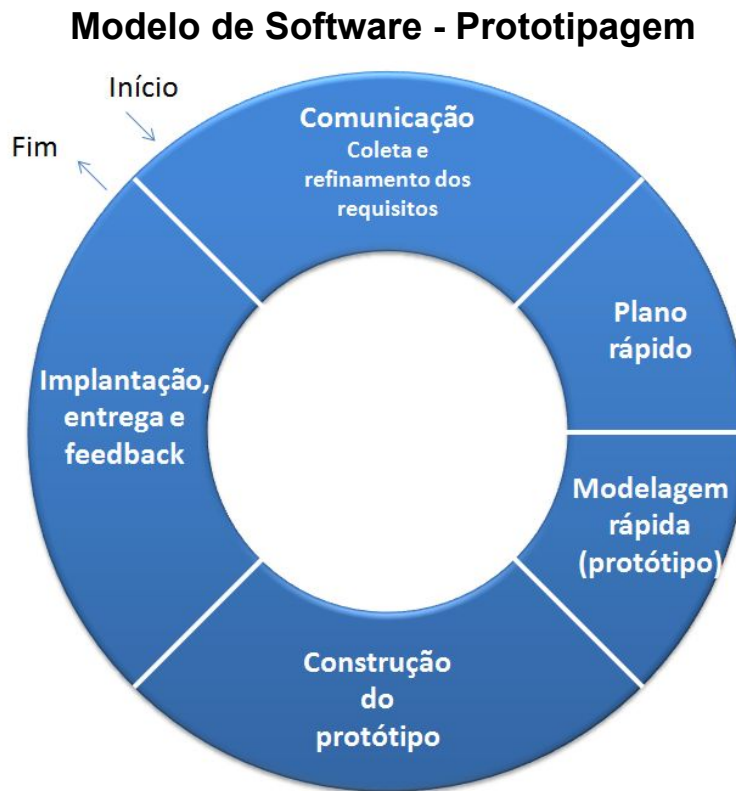
DESVANTAGENS : projetos grandes precisam de muitas equipes; exige comprometimento dos clientes e desenvolvedores; sistemas não modularizados são problemáticos; não é adequado para projetos com grandes riscos técnicos.

Modelo RAD (Rapid Application Development)



Modelo Prototipagem

A prototipagem pode ser empregada em conjunto com outros modelos para auxiliar no entendimento do sistema, em geral ela é mais utilizada como uma técnica para entendimento do sistema. A próxima figura apresenta o esquema deste modelo.



Fonte: <https://julianakolb.wordpress.com>

Prototipagem - Vantagens / Desvantagens

Vantagens:

- Maior participação e comprometimento dos clientes e usuários;
- Os resultados são apresentados mais rapidamente;

Desvantagens:

- Forte dependência das linguagens e ambientes utilizados, bem como da experiência da equipe;
- O cliente tende a considerar o protótipo como versão final, podendo comprometer a qualidade do projeto;
- O desenvolvedor tende a fazer concessões na implementação, a fim de colocar um protótipo em funcionamento rapidamente podendo comprometer o sistema.



(ADS)
Modelo de Dados

Prof. Dr. Eliseu LS

Campo identificador - Chave Primária

Toda entidade que será transformada em tabela ou objeto deve ter um campo identificador, nem que seja um número sequencial, um código, numérico de um documento, etc. Este campo é denominado de **CHAVE PRIMÁRIA**.

| Id_Chave | Nome | Idade |
|----------|-------------------|-------|
| 1 | JOÃO DA SILVA | 20 |
| 2 | ALEXANDRA ALMEIDA | 25 |
| 3 | MARCIO GOMES | 18 |
| 4 | GEORGE OLIVEIRA | 35 |
| 5 | MARIA DA SILVA | 40 |
| 6 | JOSÉ INÁCIO | 45 |
| 7 | | |
| 8 | | |
| 9 | | |

Características da Chave Primária

ÚNICA : Deve ser usado apenas um campo para ser a chave primária, por exemplo, cnpj, cpf, número sequencial. Este campo não pode ser duplicado, cada ocorrência de uma entidade têm a sua chave única.

NÃO NULA : O campo deverá ser preenchido, jamais poderá ser nul, vazio, preenchido por zeros, etc.

MÍNIMA : Deve ser usado um campo com comprimento menor possível, não é recomendado que seja um nome por exemplo, geralmente é um valor número individual que cada ocorrência da entidade têm associado a ela.

ADO 2 - Modelagem de dados

- a) Descreva as fases metodológicas genéricas da engenharia de software para produção de um software;
- b) Descreva o que você entende por fluxo de processo explicando cada um deles segundo.
- c) O que seria um ciclo de vida de um projeto de criação de software, descreve as fases do modelo em cascata segundo Bezerra.
- d) Descreva a principal diferença entre o modelo Incremental e o modelo de prototipação.
- e) Qual é a norma no Brasil que regulamenta e padroniza o processo de criação de software.
- f) Identifique as chaves primárias das entidades identificadas na ADO 1

CENÁRIO 2 - Estacionamento : Identifique as entidades, atributos e chave primária de cada entidade para um sistema que propõe o cadastro de clientes de um estacionamento que deve registrar os dados do cliente, do veículo, do funcionário, horário de entrada, saída e finalmente o valor a pagar.