UNIVERSIDADE DE BRASÍLIA

FACULDADE UnB - CAMPUS GAMA

MÉTODOS DE DESENVOLVIMENTO DE SOFTWARE

Documento de Resumo sobre Termos

Utilizados na Disciplina

Termos tratados neste documento:

# Documentos de Projeto de Software

* Documento de Visão
* Documento de Arquitetura
* TAP (Termo de Abertura do Projeto)

# Scrum - Papéis/Habilidades Ágeis

* Scrum Master
* Product Owner
* Sprints, Sprint Planning, reuniões de Sprint, Review

# Documento de Visão

Relativamente autoexplicativo; define o escopo e o propósito de um programa, exposto claramente para ambos o usuário e desenvolvedor. É importante denotar a solução proposta, assim como os recursos utilizados na produção do mesmo.

Segue uma estrutura de tópicos de acordo com o [site da IBM](https://www.ibm.com/docs/pt-br/elm/6.0.5?topic=requirements-vision-document):

1. **Introdução**

2. **Posicionando** – Qual problema o programa resolve, qual a posição exclusiva que preenche no mercado;

3. **Descrição da Parte Interessada e do Usuário** - Demográfica de mercado, todas partes interessadas, todos os tipos de usuário e seu ambiente;

4. **Visão Geral do Produto** – O que o programa oferece de único, benefícios e suas suposições (tem de estar em X sistema operacional);

5. **Recursos do Produto** – Quais as capacidades do programa e o que ele faz (importante descrever em nível geral);

6. **Restrições** - Requisitos operacionais, regulamentares;

7. **Faixas de Qualidade** - Até que ponto o programa é garantido de ser executado com qualidade;

8. **Precedência e Prioridade** – dos recursos;

9. **Outros Requisitos do Produto** - Padrões para qual o programa está conforme, requisitos de sistema, de desempenho e ambientais;

10. **Requisitos de Documentação** - Arquivo README, link para ajuda online, guia de instalação;

11. **Apêndice** - Deve conter atributo, benefícios, esforço, risco, estabilidade, liberação de destino, design e motivo.

# Documento de Arquitetura

Retirado diretamente [deste link da UFPE](https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/workproducts/rup_software_architecture_document_C367485C.html):

“O documento de arquitetura de software fornece uma visão geral de arquitetura abrangente do sistema de software. Serve como um meio de comunicação entre o **arquiteto de** **software** e outros **membros de equipe de projeto**, com relação a decisões arquiteturalmente significativas tomadas sobre o projeto.”

É importante denotar que esse documento sugere uma divisão de equipe entre um arquiteto de software, designer de teste, gerenciador de teste, designer geral, entre outros membros - não sei se precisamos seguir essa divisão, mas acho que é de interesse pra podermos nos organizar melhor, dependendo da situação.

O documento de arquitetura em si é constituído com o objetivo de cumprir diversas tarefas:

* Definir a arquitetura do programa e restringir as técnicas arquiteturais a serem utilizadas no sistema;
* Sintetizar pelo menos uma solução (que pode ser simplesmente conceitual) que atenda aos requisitos críticos de arquitetura;
* Definir abordagem do teste, que é identificar as técnicas que serão utilizadas no teste, descrever o funcionamento e definir uma arquitetura para poder automatizar os testes;
* Definir os requisitos necessários para esse ambiente de teste, ou como a abordagem de teste será restringida pelo software;
* Descrever a arquitetura em tempo de execução, incluindo processos simultâneos, as classes que estarão ativas e suas instâncias e seus relacionamentos com o sistema operacional e

encadeamento;

* Identificar motivadores de teste, que encorajará iterações constantes sobre o programa;
* Entre outros.

NOTA: Esse é um resumo altamente truncado, não sei o nível que precisaremos entrar a fundo no documento de arquitetura, mas recomendo acessar o link da UFPE.

Segue também um [link de um documento de arquitetura post-mortem feito por alunos de MDS](https://fga-eps-mds.github.io/2020.2-CheeryUP/#/./wiki/Documents/Documento_de_Arquitetura) para usarmos de base.

De acordo com o link acima, é de interesse adicionarmos uma tabela contendo significado de siglas, qual modelo arquitetural será seguido e tabelas contendo classes, variáveis e outros dados relevantes de se mostrar.

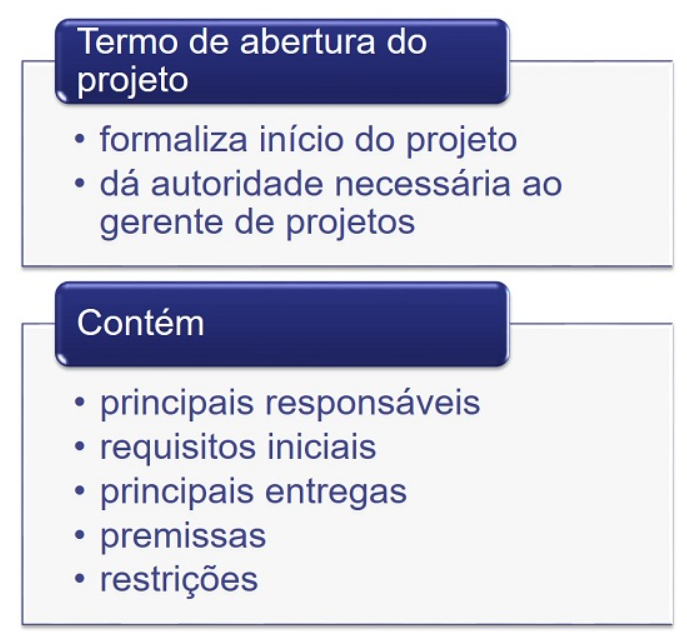
# TAP (Termo de Abertura do Projeto)

Como o nome sugere, este termo é um documento que marca oficialmente o começo de um projeto – ele também sugere uma organização de equipe similar ao Documento de Arquitetura.

[Link dos](https://escritoriodeprojetos.com.br/termo-de-abertura-do-projeto) conceitos;

De acordo com a professora, nós vamos abordar esse documento mais pra depois no processo, talvez pra segunda parte da matéria? Mas o importante agora é focarmos nos treinamentos de Git e podermos separar bem quem vai fazer o quê no método de Scrum.

De qualquer jeito, segue uma quebrada dos conceitos que envolvem TAP:



Idealmente, montaríamos o TAP com ambos o Documento de Visão e o Documento de Arquitetura já prontos, e em contato com o nosso Product Owner (em caso de Scrum).

Quando lançado, ele vai dar autoridade ao gerente de começar a alocar os recursos de organização certinho – essencialmente, é o documento que marca a abertura do projeto.

De acordo com o link da página anterior, seguem os pontos necessários para executar:

* Gerente de projetos designado e nível de autoridade atribuída;
* Requisitos que satisfazem as necessidades do cliente, do patrocinador e de outras partes interessadas;
* Necessidades de negócios, descrição de alto nível do projeto ou requisitos do produto para o qual o projeto é realizado;
* Objetivo ou justificativa do projeto;
* Cronograma de marcos sumarizado (o que vamos fazer nas sprints);
* Influência das partes interessadas;
* Organizações funcionais e sua participação;
* Premissas organizacionais, ambientais e externas;
* Restrições organizacionais, ambientais e externas;
* Caso de negócios justificando o projeto, incluindo o retorno sobre o investimento;
* Orçamento sumarizado.