

AULA 1 – GERENCIAMENTO E PLANEJAMENTO DE PROJETOS DE SOFTWARE

OBJETIVO DA AULA

Conhecer os processos de gerenciamento e planejamento da elaboração de um software, apresentando boas práticas e ferramentas que podem auxiliar nesse processo.

APRESENTAÇÃO

Esta aula tem o objetivo de apresentar a conceituação de projetos de software, assim como abordar o processo de gerenciamento e validação na construção e entrega de um produto.

Existem diversas ferramentas que podem ajudar no processo de gerenciamento do projeto, elas vão ajudar ao gerente de projeto a organizar, visualizar e controlar o projeto, assim como permitirá o compartilhamento de atividades entre membros da equipe.

O processo de gerenciamento é fundamental para a garantia do controle da condução do projeto, bem como tem o objetivo de apoiar e garantir entregas com qualidade.

1. O GERENCIAMENTO DO PROJETO

Chegamos ao momento em que tudo aquilo que definimos a partir da fase de elicitação de requisitos finalmente é executado, ganhando corpo e, logicamente, precisará ser gerenciado.

Conforme o *Guia PMBOK*, produzido pelo *Project Management Institute*, projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único.

Dessa forma, podemos dizer que o Projeto de Software é uma tarefa específica que possui começo, meio e fim, ou seja, não é considerado um projeto de software uma atividade que faz parte de uma rotina fixa de uma organização.

O projeto de software interliga os requisitos elicitados com a sua implementação. Gerenciar um projeto de software é uma das etapas fundamentais no ciclo de desenvolvimento de software.

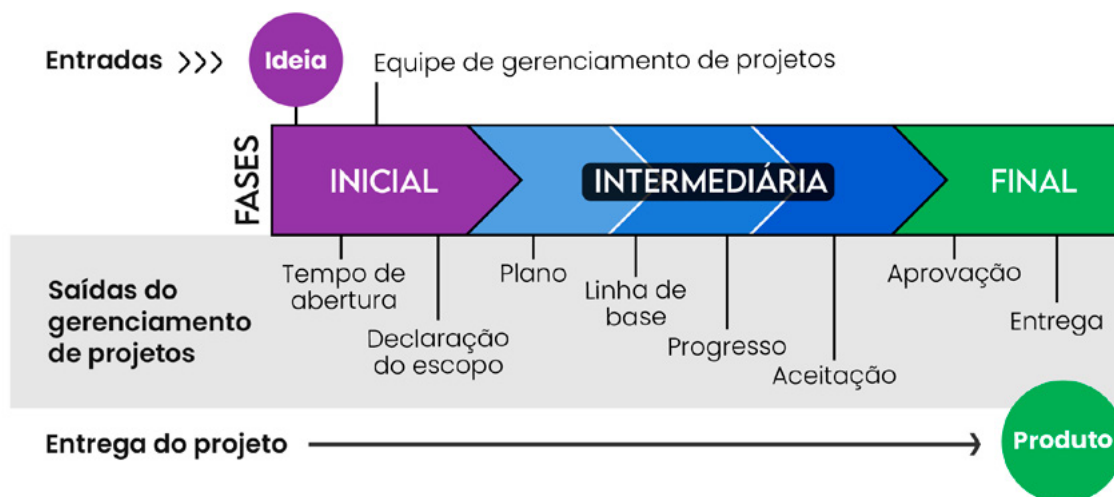
O gerenciamento de projetos de software não é tão diferente do processo de gerenciamento de outros projetos, pois possuem as seguintes características:

- É único;
- Possui um objetivo claro;
- Possui entradas;
- Possui saídas;

- Possui início e fim;
- Possui cronogramas com prazos que precisam ser respeitados;
- Possui limitação de recursos.

A Figura 1 mostra um exemplo de ciclo de vida de projetos de software.

FIGURA 1 | **Exemplo de Ciclo de Vida do Projeto de Software**

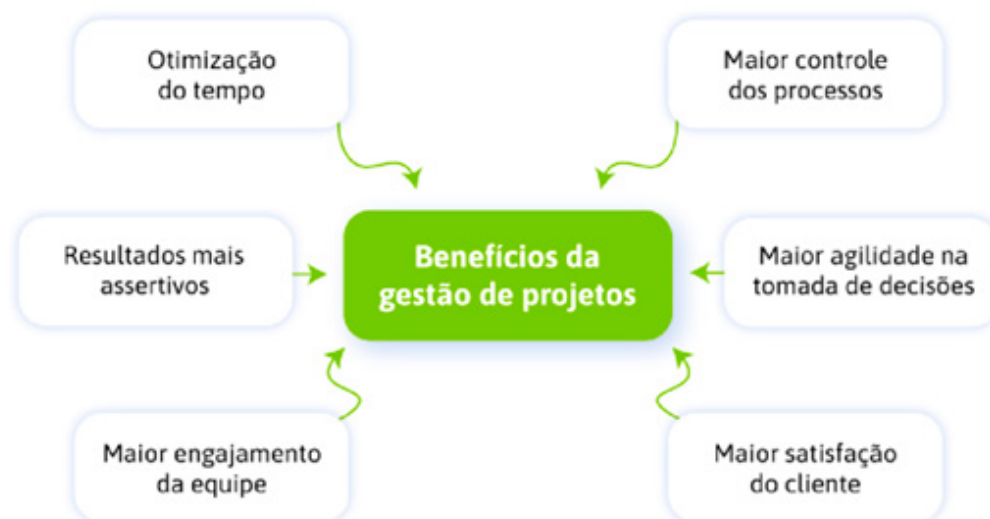


Fonte: Carvalho (2018, p.39)

O gerenciamento de projeto de software traz uma série de benefícios para a equipe, para a organização e, por tabela, para o cliente. Podemos citar imediatamente a questão da redução de custos e desperdícios de recursos materiais e humanos, assim como a mitigação dos riscos.

A Figura 2 mostra outros benefícios importantes no processo de gerenciamento de projetos de software:

FIGURA 2 | **Benefícios no Gerenciamento de Projetos**



Assim, precisamos de recursos humanos qualificados para atuar como gerente de projetos, garantindo o sucesso do produto final que será entregue ao cliente. Esse gestor precisará, ao longo de seu trabalho, coordenar as seguintes tarefas:

- Controlar escopo;
- Controlar prazos;
- Controlar custos;
- Monitorar indicadores;
- Tomar decisões;
- Distribuir tarefas à equipe.

Na área de TI é muito comum que haja um rodízio na função de gerente de projeto, ou seja, para cada novo projeto alguém da equipe é promovido para a função. Essa prática visa aumentar a qualidade do time de desenvolvimento, uma vez que todos terão a oportunidade de gerenciar um projeto ao menos uma vez.

O ciclo de vida do projeto pode ser dividido em duas características principais, definidas como Ciclo de Vida Preditivo e Ciclo de Vida Adaptativo.

No Ciclo de Vida Preditivo precisamos definir e detalhar o escopo de todo o projeto logo no início, o que permite a sua total organização desde o começo.

O Ciclo de Vida Adaptativo possui uma realidade mais próxima dos projetos da área de Tecnologia da Informação, pois ele vai sendo detalhada à medida que o projeto vai sendo construído, ou seja, aprofundado a cada iteração.

2. ÁREAS DE CONHECIMENTO DO PMBOK

As áreas de conhecimento têm uma forte abordagem pelo guia norte-americano chamado PMBOK (*Project Management of Knowledge*). Este guia é o mais conhecido e praticado mundialmente na área de gerenciamento de projetos, ele agrupa em 10 áreas, os principais aspectos envolvidos em um projeto, que interagem e o seu gerenciamento. Essas áreas são: escopo, cronograma, custos, qualidade, recursos, comunicação, riscos, aquisições, partes interessadas e integração de projetos.

Vamos conhecer brevemente o objetivo de cada área:

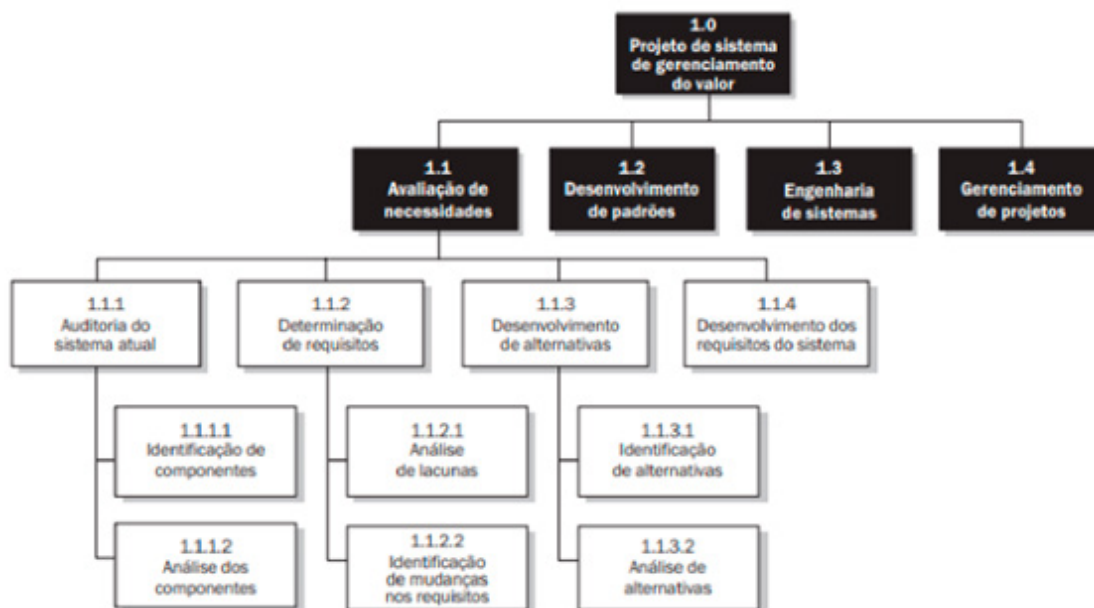
- Gerenciamento de Escopo – Tratar da descrição formal do que precisa ser feito e entregue no projeto;
- Gerenciamento de Cronograma – Cuidar do monitoramento e controle dos prazos relacionados a cada etapa e ao projeto como um todo;

- Gerenciamento de Custos – Cuidar da questão financeira, para que o projeto não ultrapasse o orçamento que foi definido inicialmente;
- Gerenciamento de Qualidade – Garantir que um controle de qualidade seja implementado para que o projeto alcance um nível aceitável de qualidade;
- Gerenciamento de Recursos – Gerenciar os recursos necessários para a condução do projeto, como recursos humanos, materiais, equipamentos e infraestrutura necessária;
- Gerenciamento de Comunicação – Assegurar que a informação do projeto seja distribuída, coletada, armazenada e gerenciada.
- Gerenciamento de Riscos – Mitigar os riscos do projeto tomando medidas que levem a identificação, análise e planejamento de respostas para os pontos de vulnerabilidade do projeto;
- Gerenciamento de Aquisições – Auxiliar no processo de tomada de decisão de compra e tratar do relacionamento com os fornecedores, estabelecer normas para comprar produtos e contratar serviços externos à equipe do projeto;
- Gerenciamento de Partes Interessadas – Identificar todas as pessoas e entidades que podem afetar o projeto ou serem afetadas por ele;
- Gerenciamento de Integração de Projetos – Reunir as demais áreas do projeto para atingir um objetivo comum, coordenando todas as práticas e demais áreas.

Para cada uma das áreas de conhecimento no gerenciamento de projetos temos ferramentas que apoiam o processo e ajudam o gerente a conduzir o projeto com qualidade.

No Gerenciamento de Escopo temos os processos de Inspeção, que incluem atividades de medição, exame e validação aplicadas para avaliar se os requisitos e critérios para aceitação do produto; e também os EAPs (Estrutura Analítica do Projeto), ferramentas que ajudam a dividir e visualiza o escopo do projeto sob forma de diagramas. A Figura 3 mostra um exemplo de EAPs.

FIGURA 3 | Divisão do Projeto em EAPs



Fonte: <https://www.uuax.com.br/2019/02/modelo-de-eap/>. Acesso em: 05 dez. 2022

O conteúdo deste livro eletrônico é licenciado para GRAN FACULDADE, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

O Gerenciamento de Cronograma também possui uma vasta quantidade de ferramentas que ajudam no controle dos prazos. Normalmente elas são expressas em gráficos e diagramas:

- Caminho Crítico (CPM) – Auxilia a encontrar o caminho crítico que de certa forma corresponde ao prazo total projetado do projeto;
- Gráfico de PERT – Trata da programação, coordenação e controle dos processos a partir do tempo;
- Gráfico de Gantt – permite visualizar e acompanhar as tarefas a serem executadas ao longo do projeto em uma perspectiva do tempo.

No Gerenciamento de Custo podemos usar planilhas e dashboards que apoie a visualização dos gastos e orçamento. Há também o Gráfico de PERT para Custos que é uma variação das técnicas PERT/COM, permitindo ao gerente de projetos reduzir custos, avaliar os impactos dos custos diretos e indiretos a partir da perspectiva do tempo.

No Gerenciamento de Qualidade existe o Diagrama de Afinidade, que é usado para gerenciar ideias que podem ser úteis para o estabelecimento da garantia da qualidade no projeto.

Para o Gerenciamento de Riscos existe uma infinidade de ferramentas que apoiam o processo. Uma delas é a Matriz de Impacto e Probabilidade, usada para avaliar os impactos dos riscos sobre cada um dos processos existentes nas áreas de conhecimento, bem como avaliar sua probabilidade de ocorrência. A Figura 4 mostra um exemplo da ferramenta.

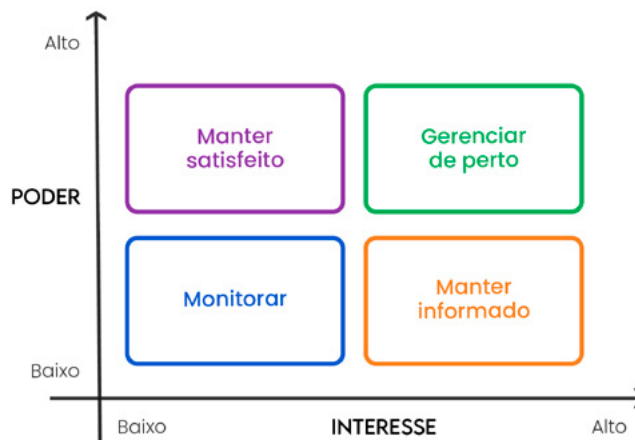
FIGURA 4 | Matriz de Impacto e Probabilidade

| | | | | |
|---------|-------|---------------|-------|-------|
| Impacto | Alto | Média | Alta | Alta |
| | Médio | Baixa | Média | Alta |
| | Baixo | Baixa | Baixa | Média |
| | | Baixo | Médio | Alto |
| | | Probabilidade | | |

Fonte: <https://doo.com.br/matriz-de-risco-probabilidade-x-impacto>. Acesso em: 05 dez. 2022.

No Gerenciamento de Partes Interessadas podemos utilizar a Matriz de Poder x Informação, que permite classificar os stakeholders em uma matriz para que facilite o entendimento sobre o fluxo de informações que devem transitar entre os stakeholders. A Figura 5 mostra um exemplo dessa matriz.

FIGURA 5 | **Matriz de Poder x Informação**



Fonte: <https://www.elirodrigues.com/2016/02/18/como-fazer-matriz-de-poder-x-interesse-das-partes-interessadas-stakeholders/>.

Acesso em: 05 dez. 2022.

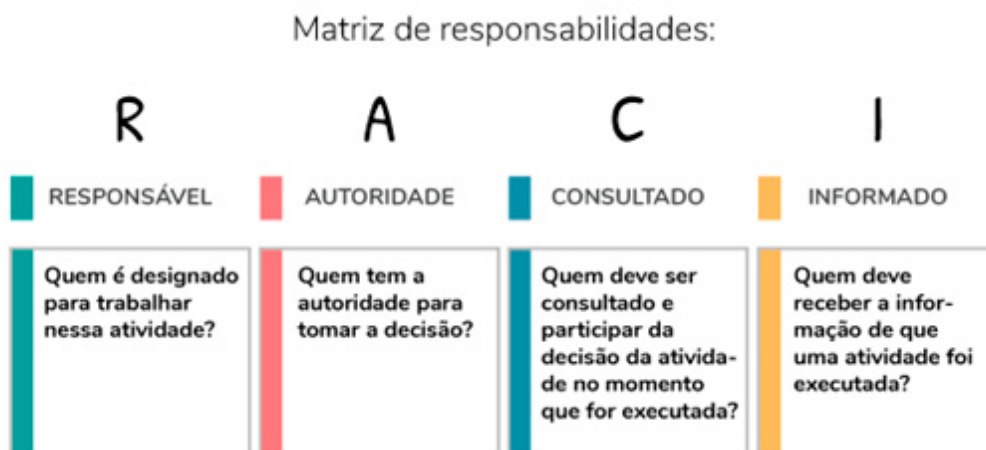
LINK

Saiba como fazer uma Matriz de Poder e Informação. Disponível em: <https://professorluizroberto.com/4-stakeholders-tabelas-e-matrizes/>. Acesso em: 05 dez. 2022.



No Gerenciamento de Recursos pode ser usada a Matriz RACI, que serve para atribuir responsabilidades às pessoas quanto às tarefas a serem desenvolvidas. A Figura 6 mostra um exemplo da Matriz.

FIGURA 6 | **Exemplo da Matriz RACI**



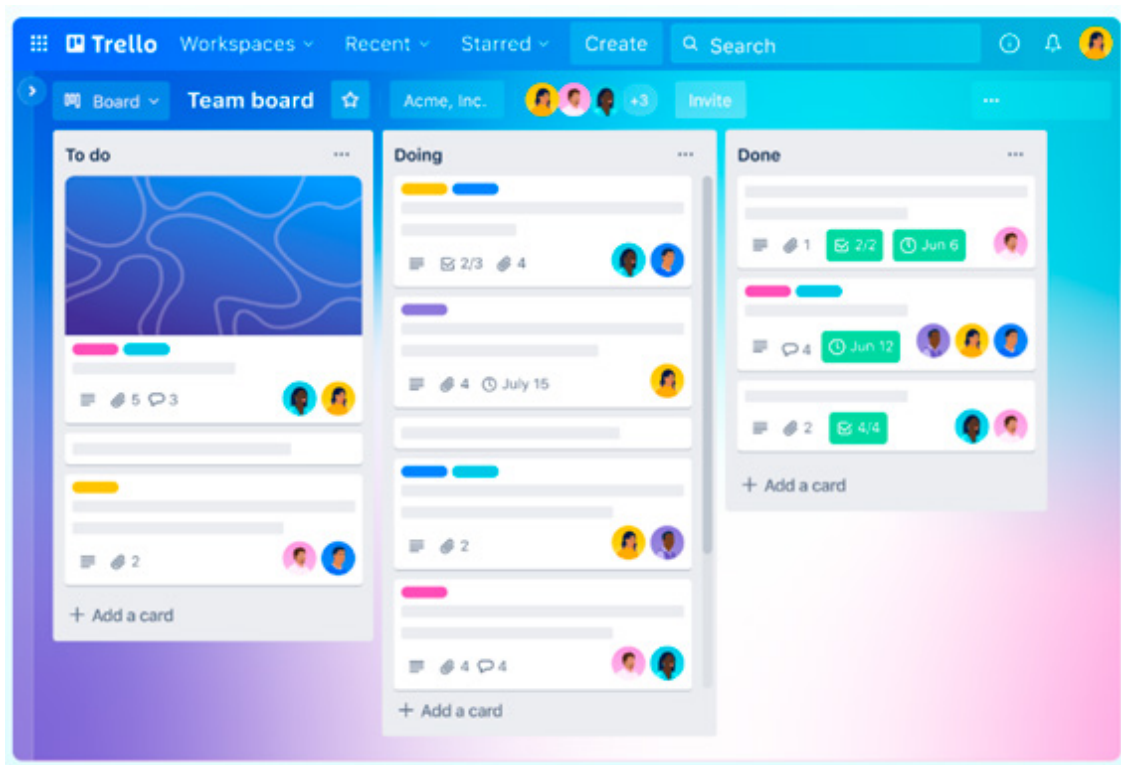
3. FERRAMENTAS DE GERENCIAMENTO DE PROJETO

Para apoiar todo o processo de gerenciamento, existe uma variedade grande de ferramentas no mercado que tornam a vida do gerente muito mais confortável.

- Dentre as mais conhecidas, podemos citar:
- Microsoft Project;
- Trello;
- Primavera;
- Microsoft Planner;
- Monday;
- Jira.

Dentre elas, o Trello acaba sendo uma opção muito viável, já que possui uma versão gratuita bastante completa, que permite que você possa gerenciar o seu projeto trabalhando em equipe.

FIGURA 7 | **O Ambiente do Trello**



Fonte: trello.com. Acesso em: 05 dez. 2022.

CONSIDERAÇÕES FINAIS

Nesta aula vimos quais são as particularidades que devem ser tratadas ao longo do gerenciamento de um projeto de software. Definimos também o projeto de software, considerando as características que são comuns aos projetos de qualquer área e natureza.

O conteúdo desta obra é propriedade intelectual da Gran Faculdade. É proibida a reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

Observamos que existem modelos como o PMBOK que definem uma série de etapas e atividades que devem ser seguidas para o gerenciamento do projeto sob todas as perspectivas.

Abordamos alguns dos principais gráficos e matrizes que apoiam ao gerenciamento de projetos, citando também alguns dos principais softwares disponíveis no mercado para apoiar o gerenciamento de projetos em geral, incluindo os de software.

MATERIAIS COMPLEMENTARES

Link: *Gerenciamento de Projeto de Software*. Disponível em: <https://www.devmedia.com.br/gestao-de-projetos-de-software/9143>. Acesso em: 05 dez. 2022.

Vídeo: *Melhores Ferramentas para Gestão de Projetos*. Disponível em: <https://www.youtube.com/watch?v=gVKrUwqqzCc>. Acesso em: 05 dez. 2022.

Vídeo: *Conceitos de Gerenciamento de Projetos*. Disponível em: <https://www.youtube.com/watch?v=9mCQORwPY-A>. Acesso em: 05 dez. 2022.

Vídeo: *O que é Gráfico de Gantt?* Disponível em: <https://www.youtube.com/watch?v=Al-1GixoGGgw>. Acesso em: 05 dez. 2022.

REFERÊNCIAS

MAXIMIANO, A.C.A, VERONEZE, F. *Gestão de Projetos*. Preditiva, Ágil e Estratégica, 6ª ed. Barueri: Atlas, 2022.

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison – Wesley, 2019.

AULA 2 – TESTES DE SOFTWARE

OBJETIVO DA AULA

Conhecer a conceituação de Testes de Software, abordando seus tipos, características e aplicabilidade.

APRESENTAÇÃO

Os testes de software são necessários para garantir que os requisitos estão funcionando da forma como foi planejado.

Existem vários tipos de testes, e estes abrangem diversos tipos de características, envolvendo questões como funcionalidade e usabilidade.

Nesta aula você aprenderá um pouco sobre essa fase tão importante no processo de construção e entrega de softwares.

1. CONCEITOS DE TESTE DE SOFTWARE

Testes de Software constitui-se como uma disciplina da Engenharia de Software que objetiva verificar se o produto funciona da maneira como foi estabelecido inicialmente.

O ofício de testar Software é tão importante que teve um incrível crescimento ao longo dos anos, tanto que hoje temos uma função específica para quem atua nesta área, que é a de Analista de Testes.

Muito embora esse conceito aparente simples, há muita confusão em relação ao que é realizado na fase de testes de software. Para início de conversa, é importante saber que testar software não significa apenas descobrir erros no processo de execução, mas também validar outros aspectos, como a validação dos requisitos, a análise de adequação no tempo de resposta, e também questões de usabilidade, pois o software precisa ser fácil de usar.

Quando um software é verificado e validado, busca-se identificar se há erros nos requisitos do produto, a validação diz respeito se o software está correto em termos de funcionamento, ou seja, se os resultados produzidos são os esperados.

Vale destacar que a realização do teste busca a identificação de erros, mas não exatamente a ausência de erros, pois um teste não tem como identificar a falta de um requisito, por exemplo.

2. CICLO DE VIDA

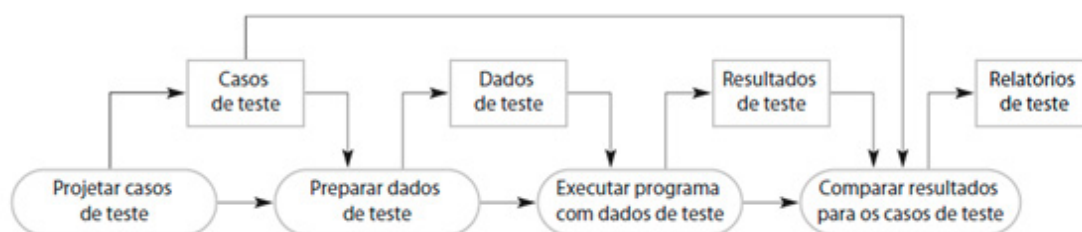
Para realizar um teste de software é importante que haja um planejamento e que um ciclo de vida seja implementado e respeitado. Embora haja outras abordagens, podemos definir um ciclo de vida de software da seguinte forma.

- 1) Análise de Requisitos;
- 2) Planejamento de Testes;
- 3) Integração do Caso de Teste;
- 4) Configuração do Ambiente de Teste;
- 5) Implementação;
- 6) Encerramento.

Na análise de requisitos os profissionais identificam todos os tipos de testes que devem ser executados, há a comunicação com as partes interessadas no projeto, são definidas prioridades e o foco das validações. Também é analisada a necessidade de automatizar os testes.

A Figura 1 mostra um modelo completo de processo de software.

FIGURA 1 | **Modelo de Processo de Software**



Fonte: Sommerville (2011, p. 147).

Na fase de planejamento são elaborados os planos de teste, são definidas ferramentas para a validação, estimados o tempo de trabalho e o custo aproximado para o projeto. Nessa etapa também são definidas as funções e responsabilidades de cada profissional.

Na integração os casos de teste e scripts são elaborados, analisados e aplicados, para que posteriormente os dados sejam avaliados.

Na configuração do ambiente de teste é necessário verificar a arquitetura utilizada, preparar e configurar o ambiente, ou seja, requisitos de hardware e software.

Na implementação os profissionais de teste vão realizar as devidas validações considerando o que foi estabelecido nos casos de teste. Nessa fase recomenda-se a documentação dos resultados dos testes, com o registro dos erros, e deve-se reportar os problemas para a equipe de desenvolvimento. Após as devidas correções as validações devem ser executadas novamente.

Por fim, no encerramento os resultados obtidos são discutidos, para reduzir falhas e custos, possibilitando também a otimização dos processos e cumprimento dos objetivos do negócio. Outra boa prática é a elaboração de um relatório de qualidade abordando todo o processo dos testes, incluindo os detalhes do projeto.

Toda a atividade no processo de testes de software vai procurar analisar o software sob algum ponto de vista técnico. Além dos requisitos funcionais, os testes de software também vão avaliar os requisitos não funcionais como:

- Operabilidade – funcionamento do sistema;
- Observabilidade – visualização dos resultados e coerência na geração das saídas, sinalizando erros;
- Controlabilidade – capacidade de automação e controle do processo;
- Decomposabilidade – capacidade de os módulos de software serem testados independentemente;
- Simplicidade – o quão fácil e intuitivo é o software;
- Estabilidade – indicador de alterações durante os testes;
- Compreensão – o quão clara é a documentação do componente.

3. TÉCNICAS DE TESTES

Os testes podem ser classificados sob diversos pontos de vista, com base em suas técnicas.

Nesse quesito temos as seguintes técnicas:

- Técnica Estrutural (Caixa Branca);
- Testes Funcionais (Caixa Branca);
- Técnica Baseada em Erros.

3.1. TÉCNICA ESTRUTURAIS (CAIXA BRANCA)

As técnicas estruturais, também conhecidas como Teste da Caixa Branca, é a técnica aplicada para descobrir falhas de construção interna das partes do software, ou seja, são verificadas estruturas condicionais e de repetições, se elas são desnecessárias ou foram mal especificadas.

Essa abordagem visa verificar se os caminhos no código foram executados ao menos uma vez. Quanto às estruturas lógicas, são apuradas todas as decisões de verdadeiro ou falso. As estruturas de repetição também são averiguadas para garantir que todos os caminhos possíveis tenham sido percorridos.

3.2. TÉCNICA FUNCIONAL (CAIXA PRETA)

Essa técnica também é conhecida como Teste da Caixa Preta, e utilizada para a validação dos requisitos sem que sejam observadas a elaboração dos códigos. Neste teste são submetidas as entradas e observado o resultado das saídas, que devem corresponder com o que foi planejado.

3.3. TÉCNICA BASEADAS EM ERROS

Essa técnica tem o objetivo buscar erros cometidos durante o desenvolvimento de sistemas. É uma técnica mais imediatista adotada pelos programadores durante o período de desenvolvimento. Ainda assim, ela deve estar contida na elaboração do plano completo de testes, considerando a teoria de que o próprio autor sempre busca testar o correto.

4. TIPOS DE TESTES

Há diversos tipos de testes utilizados dentro do processo de testagem de software. A seguir destacamos os tipos mais utilizados, com o objetivo de cada um:

- Teste de funcionalidade – busca de erros funcionais, ou seja, que tem relação direta com o funcionamento da funcionalidade testada;
- Teste de interface – verifica a usabilidade e a intuitividade na interface do software;
- Teste de conteúdo de informação – verifica as estruturas de dados definidas, criadas e persistidas no banco de dados;
- Teste de desempenho – verifica o comportamento do sistema relacionado ao desempenho quando colocado em uso;
- Teste de carga – também conhecido como teste de stress, verifica o comportamento do sistema diante de situações anormais de recursos em quantidade, frequência ou volumetria;
- Teste de regressão – consiste em repetir testes já realizados a partir de ajustes efetivados, assim quando o software tem alguma configuração mudada, deve-se garantir que essa alteração funcionará e não impactará em nenhuma outra funcionalidade que já esteja rodando;
- Teste de aceitação – este é o teste realizado pelo usuário final como treinamento, validação de todos os requisitos e obtenção da aprovação final para a obtenção da homologação do software;
- Teste de recuperação – realizado para verificar se a recuperação é executada adequadamente após uma falha operacional;
- Teste de segurança – testa questões relacionadas à segurança do sistema quanto aos dados e às invasões.

Existem ainda dois tipos de testes muito comuns, que dizem respeito ao local em que os testes são realizados:

- Teste alfa – é realizado pelo usuário nas instalações do desenvolvedor, para que qualquer ajuste percebido como necessário;
- Teste beta – realizado pelo usuário em seu próprio ambiente, este teste acontece quando o produto já está pronto para o uso, mas ainda requer um monitoramento do usuário, que **relata os comportamentos fora do esperado à equipe de desenvolvedores**

5. NÍVEL DO TESTE

Os testes também podem ser classificados pelo seu nível, isso é, eles podem ser classificados conforme os seguintes níveis:

- Componente ou unidade;
- Integração;
- Validação;
- Sistema.

Os testes ao nível de Componente ou Unidade são focados em pequenas partes do software, segundo o que foi definido na codificação do software. Este tipo de teste serve para validar pequenas partes do software com base em suas entradas possíveis e saídas esperadas.

Os testes de integração têm por objetivo unir os diversos módulos do sistema e testá-los em conjunto, são realizados assim que são finalizados os testes de cada uma das partes.

Os testes de validação, como o próprio nome sugere, validam os requisitos definidos pelo cliente. Este teste ocorre após o sistema ter sido completado e terem sido realizados os outros tipos de testes.

Nos testes de sistema são realizados os testes de segurança e recuperação, o que visa garantir a segurança do sistema e a proteção dos dados nele armazenados.

Para fins de resumo, seguem abaixo os testes abordados nesta aula e suas classificações por nível.

Testes de Componentes ou Unidade:

- Teste de Funcionalidade;
- Teste de Interface;
- Teste de Conteúdo da Informação;
- Teste de Desempenho;
- Teste de Carga.

Testes de Integração:

- Teste de Regressão.

Testes de Validação:

- Teste Alfa;

- Teste Beta

Testes de Sistema:

- Teste de Desempenho;
- Teste de Carga;
- Teste de Recuperação;
- Teste de Segurança.

Embora tenhamos abordado muitos testes e suas classificações nesta aula, cabe dizer que esse estudo não esgota os muitos tipos de testes que existem, a literatura está repleta de outras formas de se testar software.

CONSIDERAÇÕES FINAIS

Nesta aula você viu a importância da fase de teste de software como disciplina de Engenharia de Software. Esta fase resultará na aprovação do software produzido, autorizando assim a implantação do mesmo.

Testar software não significa apenas encontrar erros, mas validar requisitos e verificar questões como segurança, desempenho e usabilidade, questões não-funcionais fundamentais em um produto.

Abordamos o ciclo de vida do teste de software, apresentando o que deve ser feito antes e durante o processo de testagem de software.

Também foram discutidas, amplamente, as classificações dos testes, que podem ser de acordo com técnicas, tipos e níveis, apresentando os objetivos destas classificações e dos próprios testes.

MATERIAIS COMPLEMENTARES

Link: *Técnicas e Fundamentos de Testes de Software?* Disponível em: <https://www.devmedia.com.br/guia/tecnicas-e-fundamentos-de-testes-de-software/34403>. Acesso em: 05 dez. 2022.

Vídeo: *Entenda os 7 princípios do Teste de Software que todo engenheiro de software deve saber.* Disponível em: <https://www.youtube.com/watch?v=FCMYM01Wflc>. Acesso em: 05 dez. 2022.

Vídeo: *Níveis e Técnicas de Teste. Entenda unidade, integração, sistema, regressão, funcional e estrutural.* Disponível em: <https://www.youtube.com/watch?v=5V012tGZot4>. Acesso em: 05 dez. 2022.

REFERÊNCIAS

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison – Wesley, 2019.

O conteúdo desta obra é eletrônico e não pode ser reproduzido ou transmitido por qualquer meio eletrônico, a sua reprodução, distribuição ou divulgação, sujeitando-se aos infratores à responsabilização civil e criminal.

AULA 3 – GERENCIAMENTO DE QUALIDADE E MELHORIA DOS PROCESSOS

OBJETIVO DA AULA

Conhecer os conceitos de qualidade de software, e como a organização dos processos e implementação de *frameworks* de qualidade pode representar um diferencial para as fábricas de software.

APRESENTAÇÃO

Nesta aula vamos discutir um pouco um assunto bastante importante que é o da Qualidade de Software. Esta área é responsável por dar foco no processo de garantia de qualidade na etapa de produção e entrega de sistemas computacionais.

A qualidade no processo pode ser adquirida através da implantação de *frameworks* como o CMMI ou MPS.BR, que se preocupam em estabelecer princípios e boas práticas para apoiar todo o processo.

Também abordaremos a qualidade do ponto de vista de suas 4 subcategorias: planejamento, garantia, controle, e melhoria da qualidade de software.

1. CONCEITO DE QUALIDADE

Já entendemos que a aplicação de processos de software, gestão de requisitos e políticas de testes promove a qualidade no processo de desenvolvimento de software como um todo.

Mas precisamos entender o conceito de qualidade e saber, de fato, o que significa um software de qualidade.

De acordo com o Wikipedia:

Qualidade é o grau de utilidade esperado ou adquirido de qualquer coisa, verificável através da forma e dos elementos constitutivos do mesmo e pelo resultado do seu uso. A palavra “qualidade” tem um conceito subjetivo que está relacionada com as percepções, necessidades e resultados em cada indivíduo. Diversos fatores, como a cultura, modelos mentais, tipo de produto ou serviço prestado, necessidades e expectativas influenciam diretamente a percepção da qualidade.

Esta definição aborda uma importante questão, o grau de utilidade, ou seja, realizar aquilo que foi combinado com o cliente. Dessa forma, e pensando agora em termos computacionais, a IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) define qualidade de Software como: “grau de conformidade de um sistema, componente ou processo com os respectivos requisitos”, ou “grau de conformidade de um sistema, componente ou processo com as necessidades e expectativas de clientes ou usuários”.

Estas definições esclarecem que o sistema precisa estar de acordo com os requisitos elicitados, e com as expectativas do cliente. Podemos acrescentar a esta definição, que se espera que, para ter qualidade, o Software também precisa corresponder com o orçamento e o prazos estipulados ao cliente.

A preocupação com a qualidade de software começou a ser discutida a partir da “crise do software”, datada dos anos 1970, quando passaram a aplicar métodos, técnicas e ferramentas da Engenharia tradicional para resolver problemas de falhas e de processos de software.

2. PROCESSOS DE GERÊNCIA DE QUALIDADE DE SOFTWARE

Os processos de qualidade de software podem ser divididos em quatro subcategorias:

- Planejamento da Qualidade de Software;
- Garantia da Qualidade de Software;
- Controle da Qualidade de Software;
- Melhoria da Qualidade de Software.

No Planejamento da Qualidade, precisamos selecionar os padrões e procedimentos apropriados para o projeto em questão.

No processo de Garantia de Qualidade, deve-se selecionar *frameworks* que sejam aderentes aos processos e padrões da empresa.

Em controle, os processos de qualidade implementados são verificados para garantir se eles estão sendo seguidos e os padrões implementados.

Por fim, o processo de Melhoria da Qualidade prevê pontos em que a qualidade pode ser continuamente aprimorada e, no caso dos *frameworks*, deve-se cumprir as etapas para se chegar ao próximo nível.

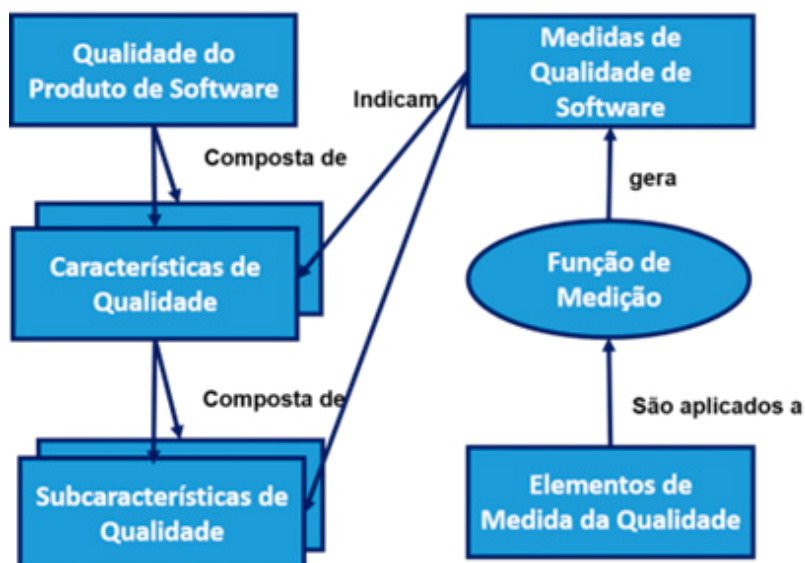
3. MEDIÇÃO

Existe a velha (mas não obsoleta) frase de William Edwards Deming que diz “o que não pode ser medido não pode ser gerenciado”. Dessa forma, todo o processo de qualidade de software vai precisar de medidas que possam mensurar processos para que a qualidade possa ser, de fato, medida, comparada e avaliada.

A norma ISO/IEC 25020:2009 apresenta um modelo de referência para a medição de qualidade de produto de software, decompondo-o em características e subcaracterísticas de qualidade, as quais podem ser avaliadas por medidas de qualidade de software. Tais medidas são geradas por funções de medição, que derivam de elementos da medida de qualidade. A

Figura 1 mostra este modelo.

FIGURA 1 | **Modelo de Referência para Medição da Qualidade do Produto**



Fonte: <https://upload.wikimedia.org/wikipedia/commons/thumb/f/f5/ModRefMedQuaProSof.png/1024px-ModRefMed-QuaProSof.png>. Acesso em: 06 dez. 2022.

4. MODELOS DE QUALIDADE DE SOFTWARE

Para apoiar os processos de qualidade de software, existem alguns *frameworks* que orientam a implantação de processos de verificação e controle dentro do processo de desenvolvimento de software em curso em fábricas de desenvolvimento de serviços de tecnologia.

Modelos de Processos de Software:

- CMMI;
- MPS.BR.

Padrões Internacionais de Processo de Software:

- ISO 12207.

Padrões Internacionais de Qualidade de Software:

- ISO 9126;
- ISO 15504;
- ISO/IEC 25010 (SQuaRe).

Padrões de certificação de Qualidade de Software:

- SCAMPI para o CMMI;
- Fórum de credenciamento e controle FCC para o MPS.br.

A Tabela 1 mostra as principais normas brasileiras e internacionais na área de software:

TABELA 1 | **Principais normas**

| Norma | Definição |
|-----------------|--|
| ISO 9126 | Características da qualidade de produtos de software. |
| NBR 13596 | Versão brasileira da ISO 9126. |
| ISO 14598 | Guias para a avaliação de produtos de software, baseados na utilização prática da norma ISO 9126. |
| ISO 12119 | Características de qualidade de pacotes de software (software de prateleira, vendido como um produto embalado). |
| IEEE P1061 | <i>Standard for Software Quality Metrics Methodology</i> . Norma que trata das metodologias para padronização da qualidade de software, incluindo algumas abordagens de medição. |
| ISO 12207 | <i>Software Life Cycle Process</i> . Norma para a qualidade do processo de desenvolvimento de software. |
| NBR ISO 9001 | Sistemas de qualidade – Modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo). |
| NBR ISO 9000-3 | Gestão de qualidade e garantia de qualidade. Aplicação da norma ISO 9000 para o processo de desenvolvimento de software. |
| NBR ISO 10011 | Auditoria de Sistemas de Qualidade (processo). |
| CMMI | <i>Capability Maturity Model Integration</i> . Modelo da SEI (Instituto de Engenharia de Software do Departamento de Defesa dos USA) para avaliação da qualidade do processo de desenvolvimento de software. Não é uma norma ISO, mas é muito bem aceita no mercado. |
| SPICE ISO 15504 | Projeto da ISO/IEC para avaliação do processo de desenvolvimento de software. Ainda não é uma norma oficial ISO, mas o processo está em andamento. |

Fonte: <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>. Acesso em: 06 dez. 2022.

Nas próximas unidades vamos conhecer, de maneira mais ampla, os *frameworks* de processo de software CMMI e MPS.BR.

CONSIDERAÇÕES FINAIS

Nesta aula falamos sobre a importância da qualidade implementada como uma importante disciplina na área de desenvolvimento de software.

Tratamos do aspecto de qualidade do ponto de vista de suas 4 subcategorias: planejamento, garantia, controle, e melhoria da qualidade de software.

Abordamos os *frameworks* e algumas normas fundamentais para o apoio ao processo de qualidade no desenvolvimento de software. Esses *frameworks* funcionam como um ponto de apoio importante para que as empresas possam garantir a implementação de qualidade na produção de software.

MATERIAIS COMPLEMENTARES

Link: *Qualidade de Software – Engenharia de Software*. Disponível em: <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>. Acesso em: 06 dez. 2022.

Vídeo: *O que é Qualidade de Software e porque você precisa ter?* Disponível em: <https://www.youtube.com/watch?v=TmH7Jeb89wY>. Acesso em: 07 dez. 2022.

REFERÊNCIAS

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison – Wesley, 2019.

AULA 4 – FRAMEWORK CMMI

OBJETIVO DA AULA

Conhecer o *Framework* CMMI, que significa *Capability Maturity Model Integration* ou Modelo Integrado de Maturidade em Capacitação. Em tender que este modelo é responsável por apresentar práticas relacionadas à maturidade nos processos de produção de Software.

APRESENTAÇÃO

Após entender os conceitos de qualidade e a importância de se produzir software com excelência, vamos aprender um pouco sobre um dos seus principais *frameworks*: o CMMI.

A sigla CMMI é a forma mais comum de nos referirmos ao *framework*, mas a sigla significa *Capability Maturity Model Integration* ou Modelo Integrado de Maturidade em Capacitação.

Nesta aula vamos aprender sobre o significado deste *framework*, além de conhecer o seu histórico, e entender o seu funcionamento, considerando os seus níveis definidos de maturidade.

1. O CMMI

O Modelo Integrado de Maturidade, ou CMMI (*Capability Maturity Model Integration*) é um *framework* que apoia as fábricas de software no desenvolvimento de sistemas, guiando os processos a partir de um conjunto de boas práticas que visam a melhoria do processo como um todo.

A ideia por trás do CMMI é organizar as boas práticas, isto é, práticas comprovadamente efetivas, em uma estrutura que ajuda a organização a estabelecer metas e prioridades para melhoria dos processos, a partir do fornecimento de um guia para a implementação destas melhorias.

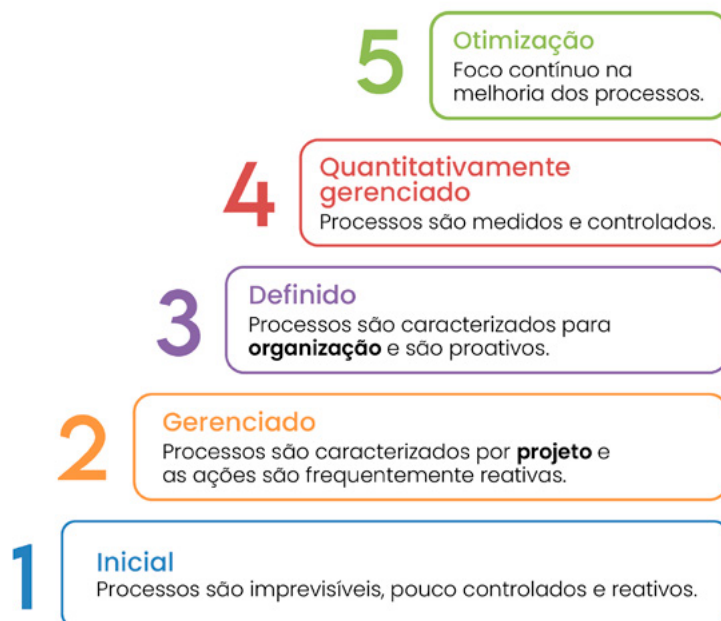
Atualmente o CMMI encontra-se na versão 2.0, que foi lançada em 2018 e substituiu a versão 1.3.

2. A ESTRUTURA DO CMMI

A versão do CMMI 2.0 apresenta 6 níveis de estágio, enquanto a anterior, dividia o *framework* em 5 níveis: Inicial, Gerenciado, Definido, Quantitativamente Gerenciado e Otimização, conforme mostra a Figura 1.

Livro Eletrônico

FIGURA 1 | Os 5 Níveis do CMMI 1.3



Fonte: Disponível em: http://www.isdbrasil.com.br/artigos/artigo_migracao.php. Acesso em: 07 dez. 2022.

Na versão 2.0 a nova divisão ganhou o nível 0, chamado de Incompleto, na Abordagem Contínua, conforme mostra a Figura 2.

Nesta abordagem o foco está em aumentar a capacitação da organização em algumas áreas, de modo que permita a organização a caminhar em direção à melhoria em áreas mais necessárias.

FIGURA 2 | Esquema de Níveis do CMM 2.0



Fonte: Do Autor (Adaptado de The Process Group. Disponível em: <https://processgroup.com/improving-capability-and-performance-with-cmmi-v2-0-what-has-changed/>)

O conteúdo deste livro eletrônico é licenciado para GRAN - 06303020092; vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

Vamos agora descrever as características e requisitos para cada um dos 6 níveis do CMMI:

- **Nível 0 – Incompleto:** Este é o primeiro nível do CMMI, ele consiste na execução parcial ou na não execução de práticas específicas, não atingindo um objetivo específico, tornando a área de processo incompleta;
- **Nível 1 – Executado:** Este nível contempla a execução de uma área de processo, assim como o alcance de seus objetivos específicos e genéricos;
- **Nível 2 – Gerenciado:** Aqui já se tem a execução de uma área de processo em conformidade com a política da organização. Além disso, já temos planejamento, monitoramento, controle e revisão dos processos;
- **Nível 3 – Definido:** Há o gerenciamento de uma área de processo, e neste nível os processos se encontram de acordo a um conjunto de processos padrões da organização;
- **Nível 4 – Quantitativamente Gerenciado:** Neste nível há a definição de uma área de processo, que está controlada, e que utiliza estatísticas e outras técnicas quantitativas, que utilizam como critério de gerenciamento dos processos os resultados deste controle;
- **Nível 5 – Otimizado:** Neste último nível há o gerenciamento quantitativo de uma área de processo, esta área é modificada e adaptada para alcance dos objetivos de negócio da organização, com foco em melhorias contínuas em aspectos tecnológicos incrementais e inovadores.

A estrutura do CMMI 2.0 possui a classificação dividida por categorias, que, por sua vez, possui desdobramentos como indica a estrutura a seguir:

Categorias:

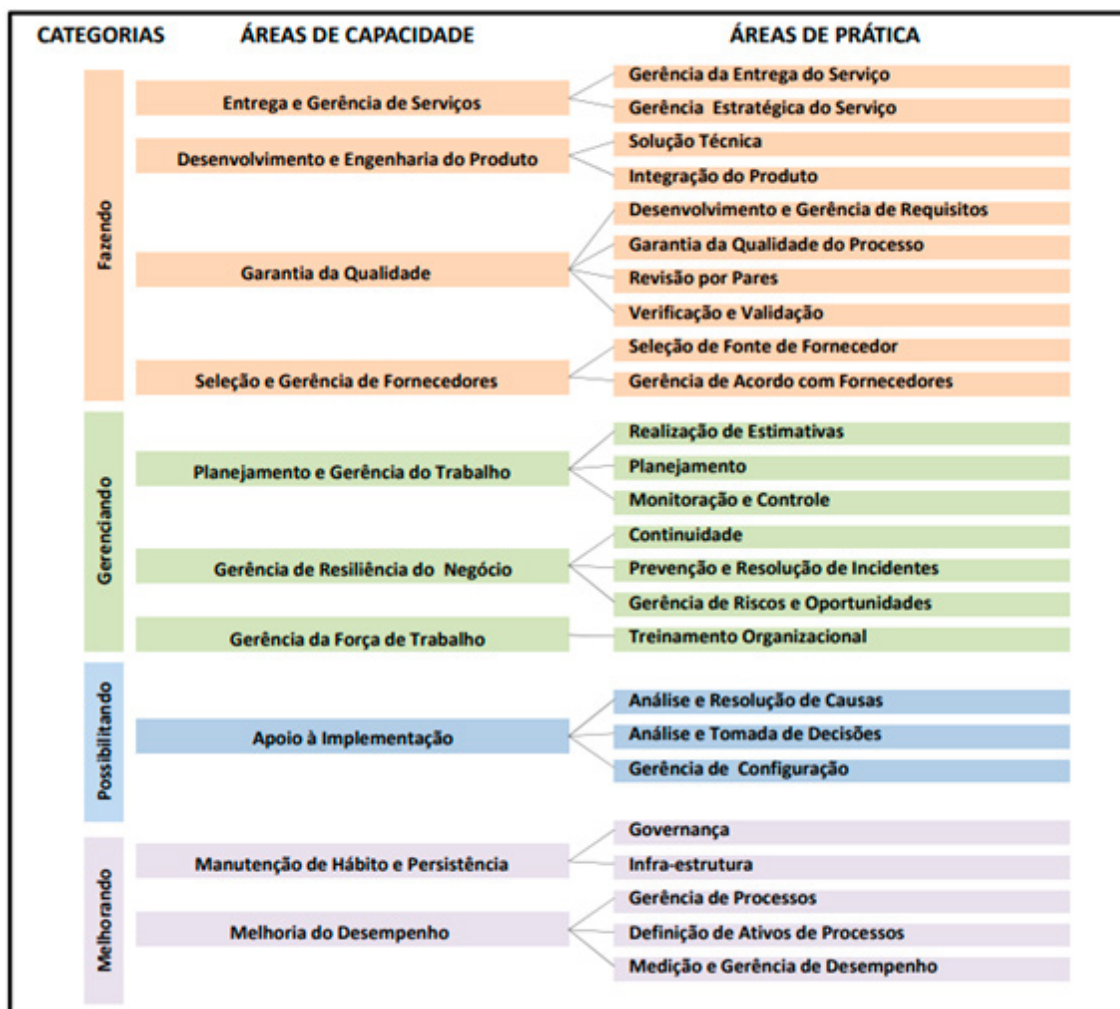
- Áreas de Capacidade.
 - Áreas de Prática.
 - Práticas

Por sua vez, a Categoria é dividida em quatro partes:

- **Fazendo (*doing*):** contém as áreas de capacidade, responsáveis por produzir e entregar soluções de qualidade;
- **Gerenciando (*managing*):** contém as áreas de capacidade, responsáveis por planejar e gerenciar a implementação das soluções;
- **Disponibilizando (*enabling*):** contém as áreas de prática, responsáveis por apoiar a implementação e as entregas;
- **Melhorando (*improving*):** contém as áreas de capacidade, responsáveis por manter e melhorar o desempenho dos processos

A Figura 3 mostra como ficou a estrutura do CMMI, tendo por base a organização da estrutura.

FIGURA 3 | **Estrutura do CMMI**



Fonte: <https://eres2020.github.io/images/palA.pdf>. Acesso em: 08 dez. 2022.

Quando falamos em qualidade e maturidade nos processos de software, a implantação do CMMI é bastante recomendável.

Normalmente esse processo se dá para grandes fábricas de software. Isso porque a implementação dos diversos níveis não é uma tarefa fácil, assim como a mudança de um nível para o outro exige bastante esforço.

Também há a questão financeira, que acaba sendo crítica, assim como a mudança de cultura, que é indispensável. Isso acaba sendo um estágio difícil de lidar, pois os desenvolvedores vão precisar se adaptar à nova realidade em que a empresa passa a gerir os seus processos de produção de software.

Existem diversas consultorias especializadas que acabam orientando e apoiando as organizações na obtenção da certificação CMMI, e também assessorando as mesmas a continuar sempre o processo de melhoria e maturidade.

CONSIDERAÇÕES FINAIS

Nesta aula conhecemos um pouco sobre o *framework* CMMI. O CMMI é um modelo de maturidade e capacidade que apoia os processos de desenvolvimento de software.

Vimos que esse modelo cuida dos processos de desenvolvimento de software e que cada processo representa um conjunto de atividades cujo objetivo é atingir uma meta previamente estipulada. Capacidade e Maturidade são questões importantes quando pensamos em qualidade do processo de desenvolvimento de Software.

É importante também perceber que o CMMI não impõe como fazer, mas indica o que deve ser feito. Outra característica marcante é o fato de o *framework* ser compatível com qualquer processo de desenvolvimento.

Nesta aula também aprendemos sobre a versão atual do CMMI, que mudou um pouco a sua arquitetura de níveis de estrutura.

MATERIAIS COMPLEMENTARES

Link: *CMMI 2.0: entenda o que é e quais as mudanças dessa versão*. Disponível em: <https://promovesolucoes.com/cmmi-2-0-entenda-o-que-e-e-quais-as-mudancas-dessa-versao/>. Acesso em: 08 dez. 2022.

Link: *CMMI 2.0 O que mudou?* Disponível em: <https://blog.grancursosonline.com.br/cmmi-2-0/>. Acesso em: 08 dez. 2022.

REFERÊNCIAS

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison – Wesley, 2019.

ROCHA, A. R. *Compreendendo as Novas Perspectivas da Melhoria de Processos As Novas Versões do CMMI e do MPS-SW*. Disponível em: <https://eres2020.github.io/images/palA.pdf>. Acesso em: 08 dez. 2022.

AULA 5 – MPS.BR

OBJETIVO DA AULA

Conhecer o *framework* brasileiro MPS.BR, voltado para o apoio às melhorias da capacidade de desenvolvimento de software.

APRESENTAÇÃO

Esta nossa última aula tem o objetivo apresentar o MPS.BR (Melhoria do Processo de Software Brasileiro), que é um programa da Softex que possui apoio do Ministério da Ciência, Tecnologia e Inovações (MCTI).

Assim como o CMMI, o MPS.BR também é um *framework* que visa apoiar o processo de maturidade e capacidade no desenvolvimento de software. Vamos ver as diferenças entre os dois modelos e conhecer a estrutura do modelo brasileiro.

Ao final da unidade, vamos observar um pequeno comparativo entre os dois *frameworks* mais conhecidos em termos de qualidade de software.

1. MPS.BR

Em dezembro de 2003, foi criado o MPS.BR, Melhoria do Processo de Software Brasileiro, programa da Softex cujo objetivo é melhorar a capacidade de desenvolvimento de software, serviços e as práticas de gestão de RH na indústria de Tecnologia da Informação.

Conforme o site oficial, a Softex é uma Organização Social Civil de Interesse Público (OSCIP) que atua há 25 anos em prol do fomento da Transformação Digital Brasileira, criando, promovendo e executando iniciativas no âmbito nacional e internacional nas áreas de tecnologia e inovação.

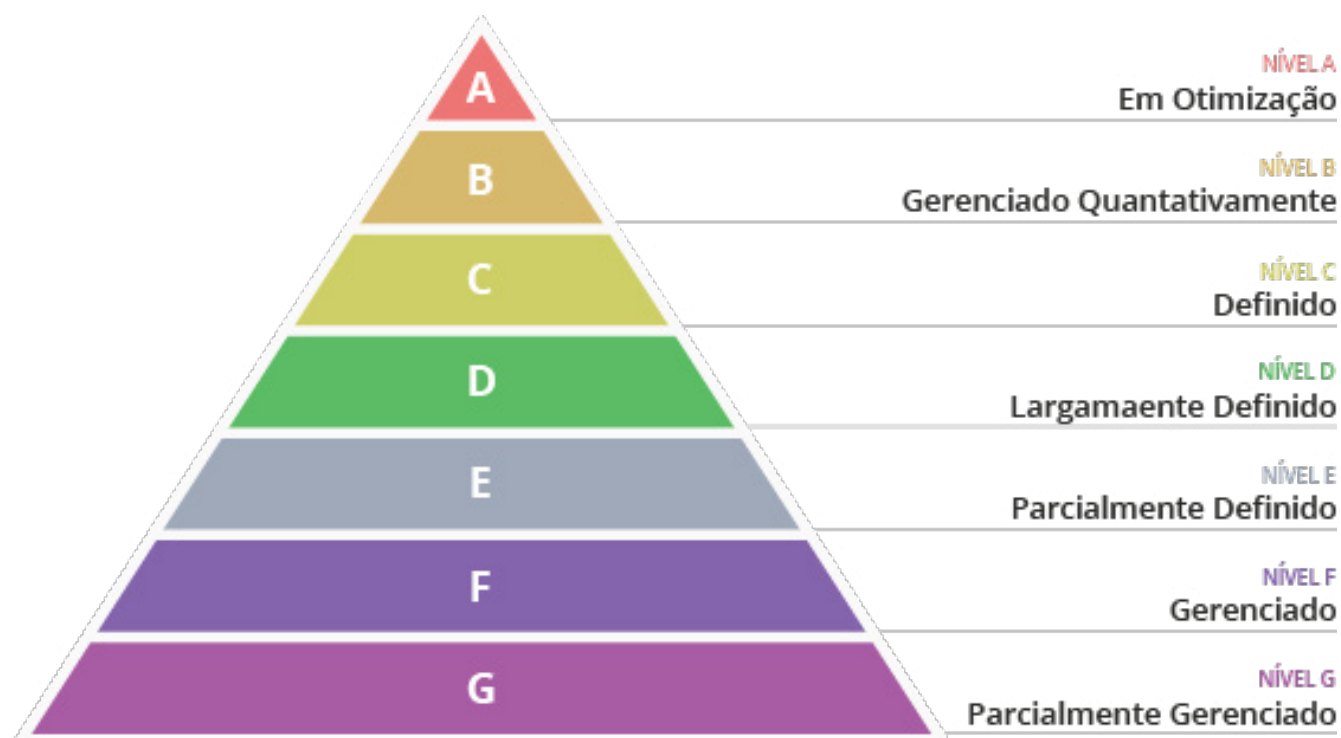
Conheça o site oficial do MPS.BR <https://softex.br/mpsbr/>. Acesso em: 08 dez. 2022.

LINK

2. ESTRUTURA DO MPS.BR

O MPS.BR é organizado segundo o que pode ser visto na Figura 1. Os modelos do MPS.BR dividem as suas práticas em processos organizados em 7 níveis. Os níveis servem como um guia para o estabelecimento da melhoria contínua.

FIGURA 1 | **Níveis de Maturidade do MPS.BR (MR-MPS-SW)**



Fonte: <https://promovesolucoes.com/mps-br/>. Acesso em: 08 dez. 2022.

Já os processos são divididos da seguinte forma:

19 Processos no MR-MPS-SW:

- modelo de referência associado à melhoria de processo de Software.

22 processos no MR-MPS-SV:

- modelo de referência associado à melhoria de processo de Serviços.

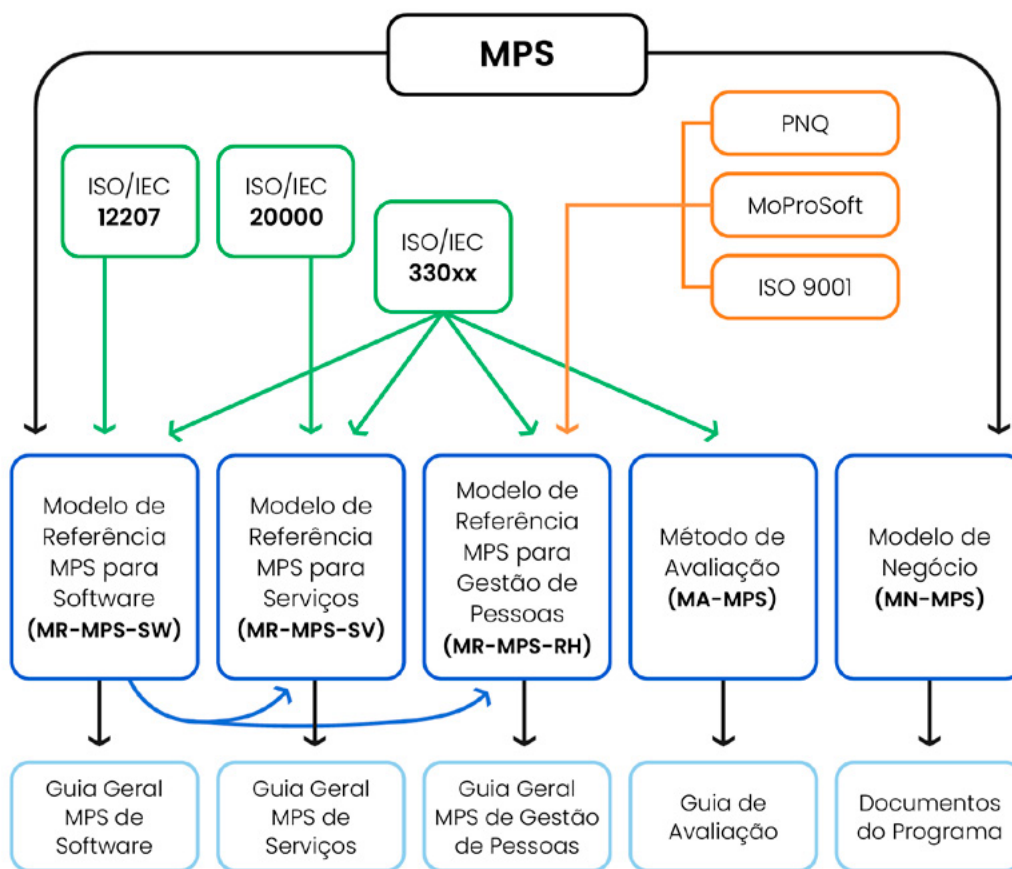
18 processos no MR-MPS-RH:

- modelo de referência associado à melhoria de processo de Gestão de Pessoas.

Esses 3 modelos visam melhorar a capacidade de desenvolvimento de software e também os serviços e as práticas de gestão de RH na indústria de produtos de TI.

A Figura 2 mostra como os componentes do programa podem ser definidos.

FIGURA 2 | Componentes do Programa MPS.BR



Fonte: <https://blog.grancursosonline.com.br/mps-br-2020/>. Acesso em: 08 dez. 2022.

O MR-MPS-SW e MR-MPS-SV podem ser usados integradamente por empresas prestadoras de serviços de desenvolvimento de software e sistemas.

O MR-MPS-RH pode ser usado por empresas na evolução da gestão das pessoas.

3. NÍVEIS DE MATURIDADE DO MR-MPS-SW

Diferentemente do CMMI, no MPS.BR os níveis são classificados por letras, tendo 7 níveis, 1 a menos que o CMMI. Além disso, no CMMI o nível máximo se dá pelo maior número, enquanto no MPS.BR, o nível máximo é a menor letra.

- G – Parcialmente Gerenciado.
- F – Gerenciado.
- E – Parcialmente Definido.
- D – Definido.
- C – Totalmente Definido.
- B – Gerenciado Quantitativamente.
- A – Em otimização.

Vamos agora entender os requisitos para que uma organização se encaixe em cada nível do MPS-BR.

3.1. NÍVEL G – PARCIALMENTE GERENCIADO

Este é o primeiro nível na escala, aqui a execução do processo é gerenciada, ou seja, um processo de capacidade é capaz de produzir resultados definidos, e de executar um processo de maneira planejada e monitorada.

Os processos presentes neste nível são “Gerência de Projetos” e “Engenharia de Requisitos”.

3.2. NÍVEL F – GERENCIADO

No nível Gerenciado a execução do processo e os produtos de trabalhos são gerenciados. O processo deve conseguir realizar a verificação objetiva de que o processo é seguido, assim como assegurar que produtos de trabalho são também avaliados objetivamente em relação ao processo e padrões, deve-se registrar os resultados dessas avaliações, comunicar e resolver as não conformidades.

Neste nível, além dos 2 processos anteriores, é composto pelos processos de “Gerência de Configuração”, “Medição”, “Aquisição”, “Gerência de Processos” e “Gerência Organizacional”.

3.3 NÍVEL E – PARCIALMENTE DEFINIDO

Aqui o processo é definido e deve ter a capacidade de identificar oportunidades de melhoria nos processos durante as atividades de garantia da qualidade, assim como deve estar disponível para a organização de informações relacionadas ao processo ou ativos de processo.

Além dos processos dos níveis anteriores, no Nível E temos também os processos de “Gerência de Recursos Humanos”, “Gerência de Processos” e “Gerência Organizacional”, esses dois últimos em versões evoluídas em relação às versões anteriores.

3.4. NÍVEL D – DEFINIDO

Este nível em particular deve estar em consonância com os níveis anteriores. Os processos são os anteriores, além das evoluções dos processos de “Gerência de Projetos” e “Engenharia de Requisitos”, e mais os processos de “Projeto e Construção do Produto”, “Integração do Produto” e “Verificação e Validação”.

3.5. NÍVEL C – TOTALMENTE DEFINIDO

Atende as capacidades dos processos anteriores. Quanto aos seus processos, ele é composto pelas evoluções dos processos de “Medição”, “Aquisição” e “Gerência Operacional”, e acrescido do processo de “Gerência de Decisões”.

3.6. NÍVEL B – GERENCIADO QUANTITATIVAMENTE

Neste nível o processo é tido como previsível, ele deve ter a capacidade de utilizar técnicas estatísticas e demais técnicas quantitativas para determinar ou prever o alcance de objetivos de qualidade e de desempenho dos processos.

Quanto aos processos, ele é composto pela evolução dos processos de “Gerência de Projetos”, “Medição”, “Aquisição”, “Gerência de Processos” e “Gerência Organizacional”.

3.7. NÍVEL A – EM OTIMIZAÇÃO

No nível mais alto da escala, o processo é melhorado continuamente e deve ter a capacidade de selecionar e implementar propostas de melhoria usando resultados da realização de análises estatísticas e quantitativa sobre efeitos esperados das propostas de melhoria para o alcance dos objetivos de negócio, qualidade e desempenho do processo.

É composto pela evolução dos processos de “Gerência de Projetos” e “Medição”, além de todos os processos anteriores.

4. BENEFÍCIOS

A implantação desse processo nas empresas pode trazer diversos benefícios para as organizações e seus colaboradores.

Dentre eles, destaco os seguintes:

- Validação da qualidade técnica da empresa;
- Redução do tempo de desenvolvimento de software;
- Otimização dos processos internos;
- Ganho de credibilidade junto aos clientes;
- Melhoria na gestão de projetos;
- Diferencial competitivo;
- Melhoria contínua.

5 MPS.BR X CMMI

Após analisarmos dois dos *frameworks* mais conhecidos de qualidade, vamos a um breve comparativo entre os modelos na Tabela 1:

O conteúdo aqui divulgado constitui-se em mera opinião do autor, não podendo ser considerado uma recomendação ou uma garantia de resultados. É vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

TABELA 1 | Comparativo entre os frameworks

| MPS.BR | CMMI |
|---|--|
| O MPS.BR é mais conhecido nacionalmente e na América Latina. | O Modelo de Qualidade CMMI é reconhecido internacionalmente |
| No MPS.BR o custo da certificação é mais acessível | O modelo CMMI envolve um grande custo na Avaliação e Certificação do Modelo |
| No MPS.BR as avaliações são bienais | No CMMI é necessário investir tempo, geralmente para se chegar aos níveis de maturidade mais altos |
| MPS.BR é um modelo criado em função das médias e pequenas empresas | O CMMI tem foco global voltado para empresas de maior porte |
| MPS.BR possui sete níveis de maturidade, onde a implantação é mais gradual | O CMMI possui seis níveis de maturidade por estágio |
| O MPS.BR é aceito como maturidade para licitações | O CMMI também é aceito como maturidade para licitações |
| O MPS.BR não torna as empresas competitivas internacionalmente | O CMMI torna as empresas competitivas internacionalmente. |
| No MPS.BR pode acontecer contrato cooperado em grupo de empresas que queiram a certificação | O CMMI não utiliza contrato conjunto de Empresas |
| Implementação mais simples | Implementação mais complexa |
| Desenvolvido em 2003 | Desenvolvido em 1992 |

Fonte: Adaptado de CMMI e MPS.BR: Um Estudo Comparativo. Disponível em: <https://docplayer.com.br/71029499-Cmmi-e-mps-br-um-estudo-comparativo-cmmi-and-mps-br-a-comparative-study.html>. Acesso em: 08 dez. 2022.

Por fim, cabe dizer que ambos os *frameworks* são importantes e plenamente reconhecidos no mercado e também aceitos em licitações.

As certificações são importantes e, muitas vezes, um pré-requisito para a participação em licitações governamentais. Assim, as empresas que desejam estarem aptas à participação nesses processos e com isso aumentar o seu leque de possibilidades, devem pensar em adotar esses modelos.

O MPS-BR é uma importante alternativa ao CMMI em organizações de médio e pequeno porte, principalmente com a sua nova versão lançada em 2020, que tornou o processo mais simples.

CONSIDERAÇÕES FINAIS

Nesta aula abordamos o *framework* MPS-BR (Melhoria de Processo do Software Brasileiro), que é uma alternativa nacional ao modelo CMMI.

Conhecemos a sua estrutura, dividida em 7 níveis, explorando um pouco cada um dos níveis e seus processos.

O processo avaliativo do *framework* visa ajudar as empresas na melhoria dos processos, no desempenho dos negócios, e na alavancagem da inovação, tornando-as mais produtivas, preparando a empresa para alcançar o tão sonhado diferencial competitivo.

Entendemos o quanto é importante a adoção de um *framework* de qualidade para a organização e a promoção da maturidade e capacidade dos processos de uma empresa, bem como que o processo pode ser um requisito fundamental para empresa que deseja participar de uma licitação.

Líder em avaliações no Brasil, o *framework* MPS.BR é um programa da Softex que apoia a inserção da Cultura da Qualidade, em organizações de qualquer perfil e tamanho.

MATERIAIS COMPLEMENTARES

Vídeo: *O que é MPS.BR – Modelo de qualidade de processo brasileiro*. Disponível em: <https://www.youtube.com/watch?v=SUTecO8YQIs>. Acesso em: 08 dez. 2022.

Vídeo: *Comparativo do CMMI versus o MPS.BR – Resumo das principais diferenças!* Disponível em: <https://www.youtube.com/watch?v=AZXFeJ92JvI>. Acesso em: 08 dez. 2022.

Link: *MPS.BR 2020*. Disponível em: <https://blog.grancursosonline.com.br/mps-br-2020/>. Acesso em: 08 dez. 2022.

REFERÊNCIAS

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison – Wesley, 2019.

**INTRODUÇÃO À ENGENHARIA DE
SOFTWARE | UNIDADE 4**

Aula 2 | Testes de Software

PROFESSOR(A): JOSÉ REGINALDO

Introdução

NESTA AULA VAMOS APRESENTAR A CONCEITUAÇÃO DE TESTES DE SOFTWARE, SEUS TIPOS, CARACTERÍSTICAS E APLICABILIDADE.

OBJETIVOS DA AULA

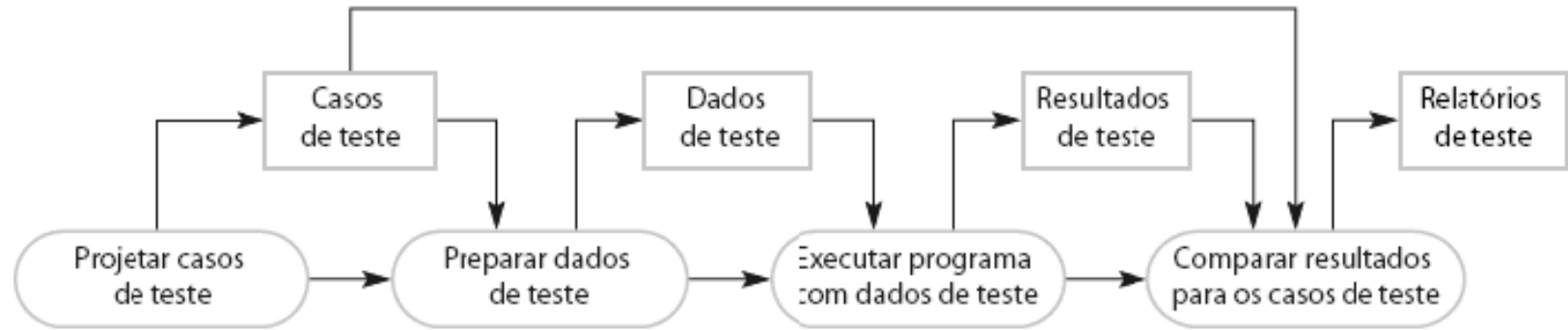
→ ABORDAR OS TESTES DE SOFTWARE NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.

O que é um Teste de Software?

Processo sistemático e planejamento que tem por finalidade única a identificação de erros.

[BARTIÉ, 2002]

Processo de Teste de Software



Fonte: SOMMERVILLE(2011)

Estratégias de Testes

- Caixa Branca
- Caixa Preta

Teste Caixa Branca

- Identifica defeito nas estruturas internas do sistema.
- Validação de algoritmos no código fonte.
- Requer conhecimento da tecnologia.
- Mais eficiente e mais difícil de implementar.

Teste Caixa Preta

- Valida se o sistema produz os resultados esperados.
- Requisitos decompostos em casos de testes
- Não requer conhecimento da tecnologia.
- Mais simples de implantar.

Tipos de Testes

- Funcionalidade
- Usabilidade
- Desempenho
- Volume
- Carga
- Segurança
- Recuperação
- Contingência
- Instalação
- Configuração

Teste de Funcionalidade

- Simula os cenários de negócio.
- Conformidade dos requisitos funcionais.
- Exemplos:
 - Validar precondições e pós-condições.
 - Validar fluxo principal.
 - Validar fluxos alternativos.

Teste de Usabilidade

- Simula as condições de utilização do software.
- Foco em navegabilidade, acessibilidade, padrão visual, clareza de mensagens, etc.
- Exemplos:
 - Avaliar a facilidade de navegação.
 - Avaliar mensagens de alerta.
 - Avaliar existência de ajuda.

Teste de Desempenho

- Avalia desempenho em situações de pico.
- Compara tempo de resposta com limites especificados.
- Exemplos:
 - Vários usuários acessando o mesmo dado.
 - Vários usuários processando a mesma transação.

Teste de Volume

- Simula condições extremas de utilização.
- Aumento contínuo de processamento.
- Determina os limites máximos do sistema.
- Exemplos:
 - Aumentar sucessivamente o volume de transações.
 - Aumentar sucessivamente o volume de consultas.
 - Aumentar sucessivamente o volume de dados.

Teste de Carga

- Simula condições atípicas de utilização.
- Variações sucessivas de processamento que chegam a ultrapassar os limites do sistema.
- Exemplos:
 - Aumentar e reduzir o volume de transações.
 - Aumentar e reduzir o tráfego de rede.
 - Aumentar número de usuários simultâneos.

Teste de Segurança

- Detecta falhas de segurança no software.
- Comprometimento da confiabilidade, integridade e disponibilidade.
- Exemplos:
 - Validar os perfis de acesso.
 - Acessar dados com um perfil inadequado.
 - Identificar pontos vulneráveis no sistema.

Teste de Recuperação

- Avalia comportamento em situações anormais.
- Testa a tolerância a falhas e a robustez do sistema.
- Exemplos:
 - Simular falha de acesso à rede.
 - Simular falha de processamento.
 - Simular defeito de um dispositivo.

Teste de Contingência

- Valida procedimentos de contingência, principalmente em casos de desastre e recuperação.
- Exemplos:
 - Instalação emergencial.
 - Recuperação de backup.
 - Recuperação da perda de conexão.

Teste de Configuração

- Valida configurações de hardware e software.
- Garantia de execução em diversos ambientes.
- Exemplos:
 - Testar em vários sistemas operacionais.
 - Testar em vários navegadores.
 - Testar em vários dispositivos.

Teste de Instalação

- Valida procedimentos de instalação.
- Realizado geralmente pelo próprio usuário.
- Exemplos:
 - Instalar o software a primeira vez.
 - Atualizar o software.
 - Remover o software.

Níveis de Testes

- Unidade
- Integração
- Sistema
- Aceite

Teste de Unidade

- Testa parte do software (ex.: função, classe, componente).
- Tipicamente executado pelo desenvolvedor.
- Pode ser automatizado.

Teste de Integração

- Testa a integração entre partes do software.
- Executado pelo desenvolvedor.
- Pode ser automatizado.

Teste de Sistema

- Testes aplicados no software como um todo.
- Executado pelo testador.
- Pode ser automatizado.

Teste de Aceite

- Testes aplicados no software para validação dos requisitos.
- Executado pelo usuário final (alpha e beta).
- Não é automatizado.

Abordagens de Testes

- Progressivo
- Regressivo

Teste Progressivo

- Testa somente as inovações do produto.
- As funcionalidades já existentes não são testadas.
- Assume que nenhum erro foi introduzido.

Teste Regressivo

- Teste parcial ou total do que não foi modificado.
- Assegura que as alterações não afetaram outras partes do produto.

Encerramento

NESTA AULA ABORDAMOS OS TESTES DE SOFTWARE E A IMPORTÂNCIA DELES PARA A VALIDAÇÃO DO SOFTWARE, SEJA NA FASE DE REQUISITOS SEJA NA FASE DE ENTREGA DO PRODUTO.

**INTRODUÇÃO À ENGENHARIA DE
SOFTWARE | UNIDADE 4**

Aula 5 | MPS.BR

PROFESSOR(A): JOSÉ REGINALDO

Introdução

NESTA AULA VAMOS APRESENTAR O MPS.BR, SUAS PRÁTICAS E NÍVEIS DE MATURIDADE.

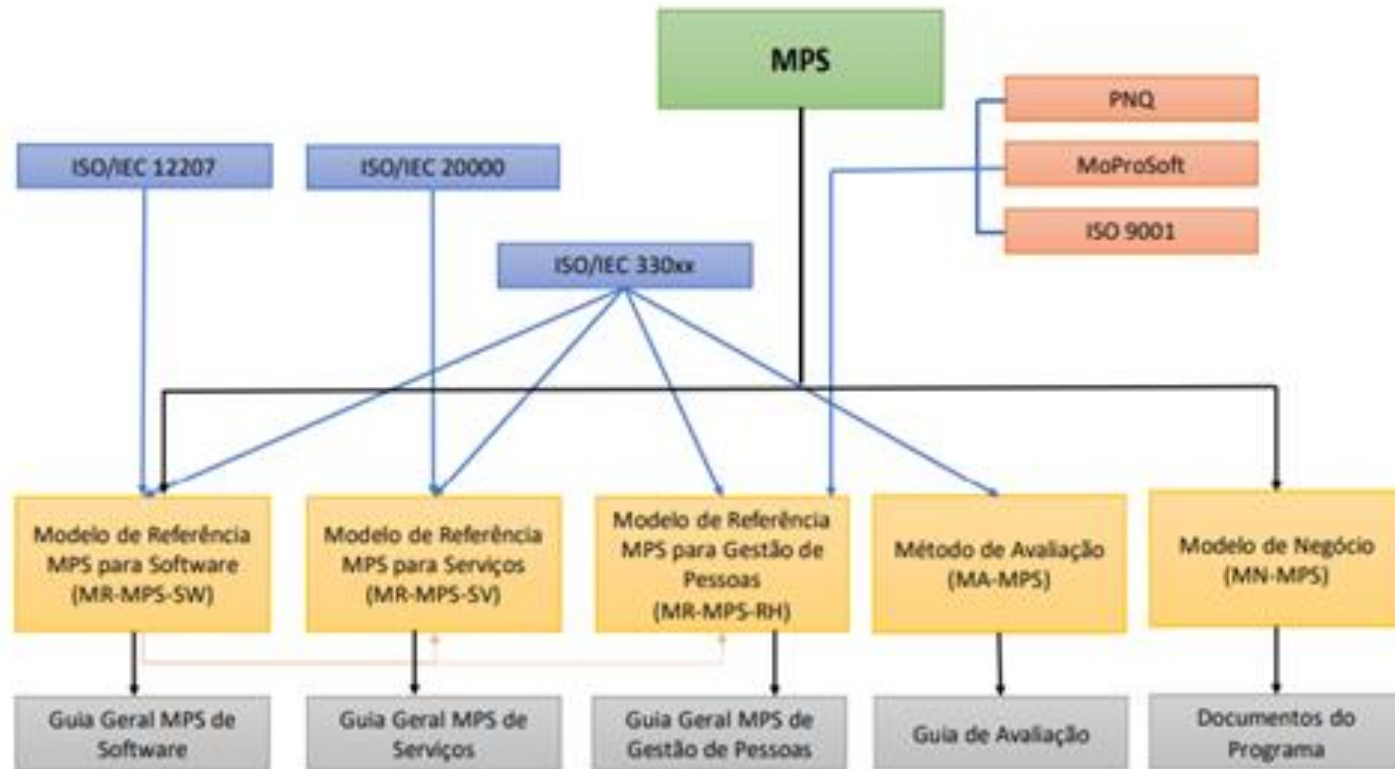
OBJETIVOS DA AULA

- CONHECER O MODELO DE MELHORIA DO PROCESSO DE SOFTWARE BRASILEIRO.

Melhoria do Processo de Software BR

- Modelo criado pela Softex com base no CMMI.
- Tem como objetivo melhorar a capacidade de desenvolvimento de software, serviços e as práticas de gestão de RH nas micros, pequenas e médias empresas de TI.

MPS.BR



Modelos de Referência do MPS.BR

- MR-MPS-SW
- MR-MPS-SV
- MR-MPS-RH

Estrutura do MPS.BR

- Práticas são agrupadas em processos.
- Processos são divididos em dois grupos: Projeto e Organizacional.
- Cada modelo de referência estabelece a quantidade de processos.

Processos de Projeto e Organizacionais

Processos de Projeto

Gerência de Projetos

Gerência de Configuração

Engenharia de Requisitos

Projeto e Construção do Produto

Integração do Produto

Verificação

Validação

Processos Organizacionais

Gerência de Recursos Humanos

Gerência Organizacional

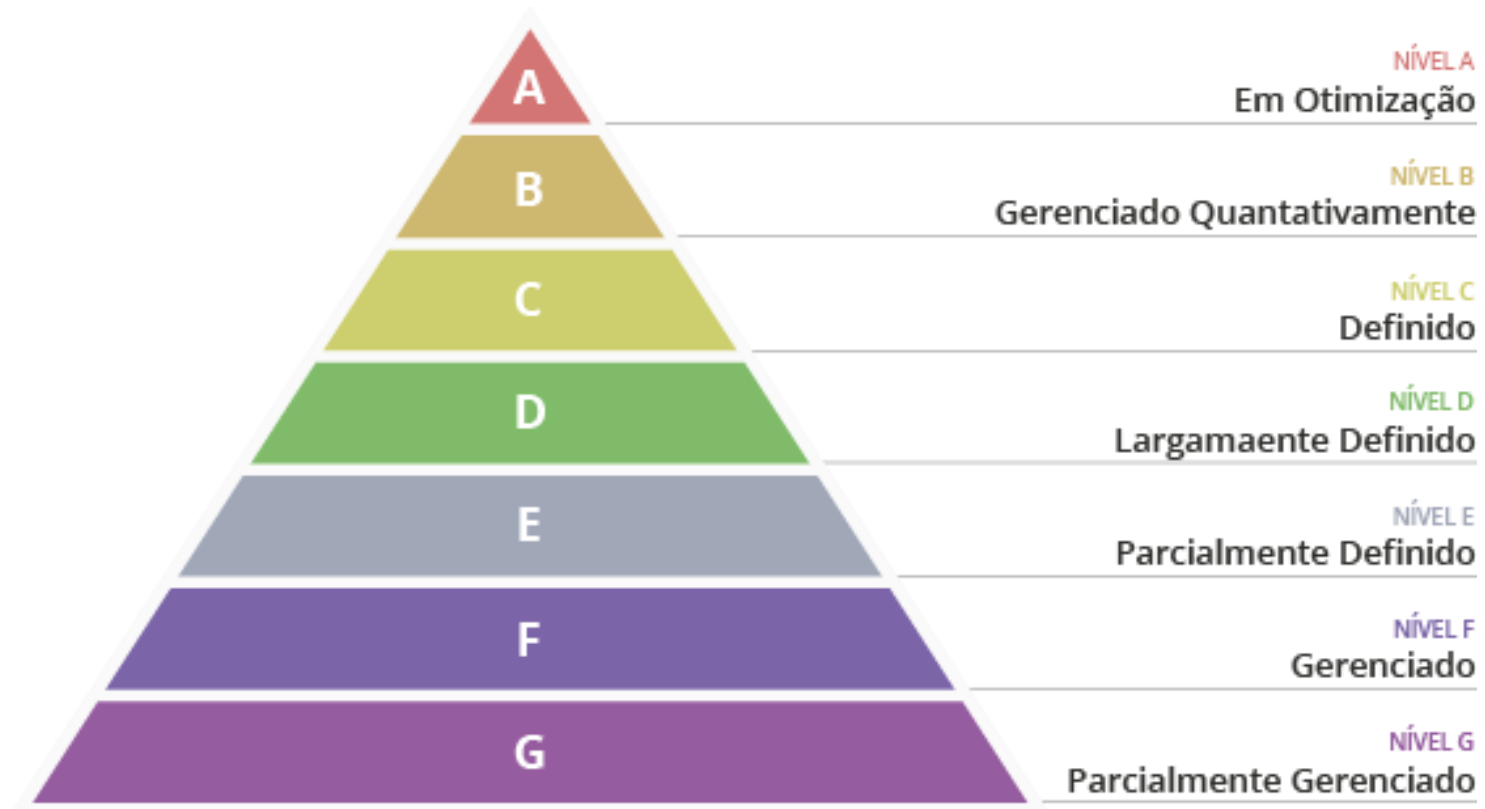
Gerência de Processos

Medição

Aquisição

Gerência de Decisões

Níveis do MPS.BR



Nível G - Parcialmente Gerenciado

- Apenas a execução do processo é gerenciada.
- O processo é executado de maneira planejada e monitorada.
- Possuir os processos Gerência de Projetos e Engenharia de Requisitos.

Nível F - Gerenciado

- A execução do processo e os produtos de trabalho são gerenciados.
- Verifica de forma objetiva se o processo é seguido, e se os produtos de trabalho seguem os padrões definidos.
- Adiciona os processos Gerência de Configuração, Medição, Aquisição, Gerência de Processos e Gerência Organizacional.

Nível E - Parcialmente Definido

- O processo é definido e implementado parcialmente.
- Adiciona os processos Gerência de Recursos Humanos, Gerência de Processos e Gerência Organizacional.

Nível D - Largamente Definido

- O processo é definido e implementado de forma mais ampla ao em relação ao nível anterior.
- Adiciona os processos Projeto e Construção do Produto, Integração do Produto e Verificação e Validação.

Nível C - Definido

- O processo é totalmente definido e implementado.
- Evolução dos processos Medição, Aquisição e Gerência Operacional, e inclusão do processo de Gerência de Decisões.

Nível B - Gerenciado Quantitativamente

- O processo é tido como previsível.
- Utiliza técnicas estatísticas e demais técnicas quantitativas para determinar ou prever o alcance de objetivos de qualidade e de desempenho dos processos.
- Evolução dos processos de Gerência de Projetos, Medição, Aquisição, Gerência de Processos e Gerência Organizacional.

Nível A - Em Otimização

- O processo é melhorado continuamente.
- Seleciona e implementa propostas de melhoria usando resultados da realização de análises estatísticas e quantitativa no nível anterior.
- Evolução dos processos de Gerência de Projetos e Medição.

Benefícios do MPS.BR

- Validação da qualidade técnica da empresa
- Redução do tempo de desenvolvimento de software
- Otimização dos processos internos
- Ganho de credibilidade junto aos clientes
- Melhoria na gestão de projetos e nos processos
- Diferencial competitivo

MPS.BR x CMMI

| MPS.BR | CMMI |
|---|---|
| Criado em função das médias e pequenas empresas | Voltado para empresas de maior porte |
| Conhecido nacionalmente e na América Latina | Reconhecido internacionalmente |
| Custo da certificação mais acessível | Maior custo na avaliação e certificação |
| Implementação mais simples | Implementação mais complexa |
| 7 níveis de maturidade | 6 níveis de maturidade |

Alinhamento dos Níveis de Maturidade

| MPS.BR | CMMI |
|--------|------|
| - | 0 |
| - | 1 |
| G | - |
| F | 2 |
| E | - |
| D | - |
| C | 3 |
| B | 4 |
| A | 5 |

Encerramento

NESTA AULA VIMOS O FRAMEWORK MPS.BR, UM MODELO DE CAPACIDADE E MATURIDADE QUE APOIA NA QUALIDADE DAS PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE DAS PEQUENAS E MÉDIAS EMPRESAS BRASILEIRAS.

**INTRODUÇÃO À ENGENHARIA DE
SOFTWARE | UNIDADE 4**

**Aula 3 | Gerenciamento
da Qualidade e Melhoria
dos Processos**

PROFESSOR(A): JOSÉ REGINALDO

Introdução

NESTA AULA VAMOS APRESENTAR O PROCESSO DE GARANTIDA DA QUALIDADE DE SOTWARE.

OBJETIVOS DA AULA

→ CONHECER OS CONCEITOS DE QUALIDADE DE SOFTWARE E AS NORMAS E MODELOS DE QUALIDADE.

O que é Qualidade de Software?

A totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas.

[ISO/IEC 25010, 2011]

O que é Qualidade de Software?

Conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais.

[PRESSMAN, 2016]

O que é Garantia da Qualidade?

Padrão sistemático e planejado de ações que são exigidas para garantir a qualidade de software.

[PRESSMAN, 2016]

Processo de GQS

- Planejamento da Qualidade
- Garantia da Qualidade
- Controle da Qualidade
- Melhoria da Qualidade

Planejamento da Qualidade

- Identifica quais os padrões de qualidade são importantes para o projeto e determina como eles serão alcançados.

Garantia da Qualidade

- Estrutura, sistematiza e executa as atividades de testes, satisfazendo os padrões de qualidade definidos no processo.

Controle da Qualidade

- Monitora e analisa os resultados do projeto, determinando se ele atende aos padrões de qualidade preestabelecidos.

Melhoria da Qualidade

- Prevê pontos em que a qualidade pode ser continuamente aprimorada.

Por que medir a qualidade?

Você não consegue controlar aquilo que não se pode medir.

[DEMARCO, 1982]

Se você não sabe para onde você quer ir, qualquer caminho você pode seguir. Se você não sabe onde você está, um mapa não vai ajudar.

[PRESSMAN, 1995]

Organismos Normativos

- International Organization for Standardization
- Institute of Electrical and Eletronics Engineers
- International Electrotechnical Commission
- Software Engineering Institute
- Associação Brasileira de Normas Técnicas

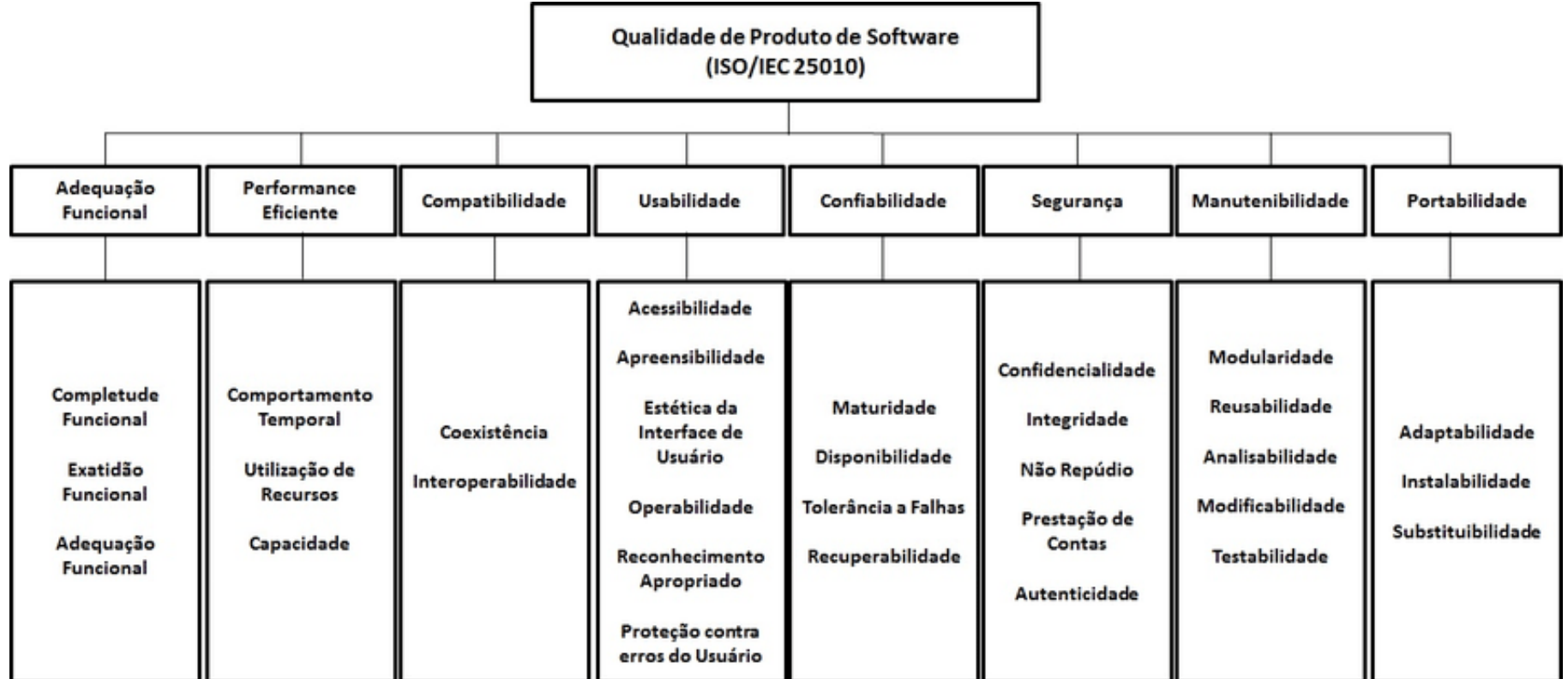
Normas e Modelos de Qualidade

- ISO/IEC 25010
- ISO/IEC 25020
- ISO/IEC 25040
- ISO/IEC 25051
- CMMI
- MPS.BR

ISO/IEC 25010

- Trata da avaliação do software do ponto de vista de suas características de qualidade.
- Aplicável para quem faz aquisição, auditoria, desenvolvimento, suporte e manutenção.

Atributos de Qualidade



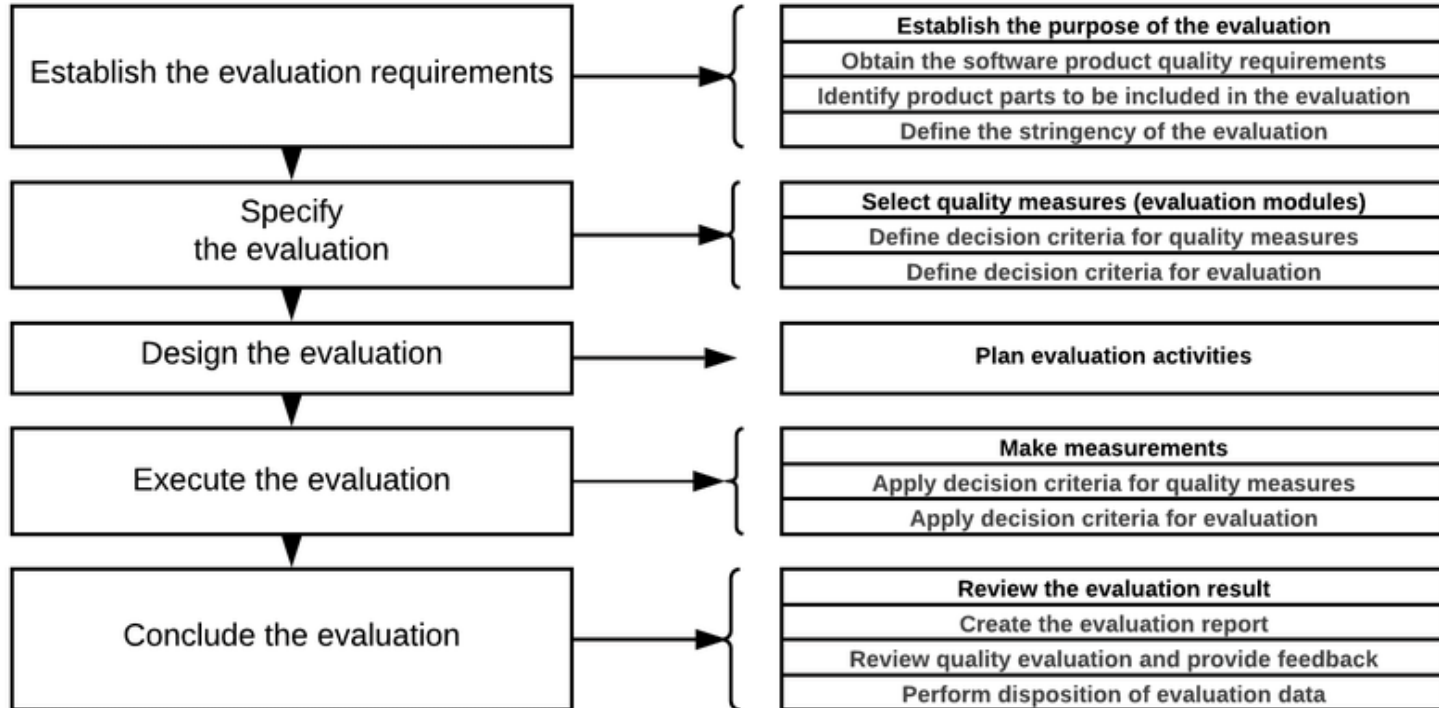
ISO/IEC 25020

- Apresenta um guia de referência para a medição de qualidade do produto de software.
- As medidas são geradas por funções de medição a partir de elementos da medida da qualidade.

ISO/IEC 25040

- Guia para avaliação de produtos de software.
- Determinação da qualidade do produto a ser adquirido.

Visão do Processo de Avaliação



ISO/IEC 25051

- Definição de requisitos de qualidade do software comercial de prateleira e de instruções para testes.
- Aplicável à avaliação de pacotes de software, considerando a forma como são oferecidos e liberados para uso no mercado.

CMMI

- Modelo que descreve orientações para a definição e implantação de processos.
- Possui um conjunto de práticas do que se deve fazer para alcançar a melhoria do processo.

MPS.BR

- Modelo de Melhoria de Processo do Software Brasileiro.
- Semelhante ao CMMI, só que mais barato.
- Apresenta uma estratégia de implementação para micros, pequenas e médias empresas.

Encerramento

NESTA AULA VIMOS A IMPORTÂNCIA DA QUALIDADE DE SOFTWARE NO PROCESSO DE DESENVOLVIMENTO E AS NORMAS E MODELOS QUE SÃO ADOTADOS PELAS EMPRESAS PARA GARANTIR A QUALIDADE NA PRODUÇÃO DE SOFTWARES.

**INTRODUÇÃO À ENGENHARIA DE
SOFTWARE | UNIDADE 4**

**Aula 4 | Framework
CMMI**

PROFESSOR(A): JOSÉ REGINALDO

Introdução

NESTA AULA VAMOS APRESENTAR O CMMI, DESCREVENDO SUAS PRATICAS E NÍVEIS DE CAPACIDADE.

OBJETIVOS DA AULA

- CONHECER O MODELO INTEGRADO DE CAPACIDADE E MATURIDADE.

Capability Maturity Model Integration

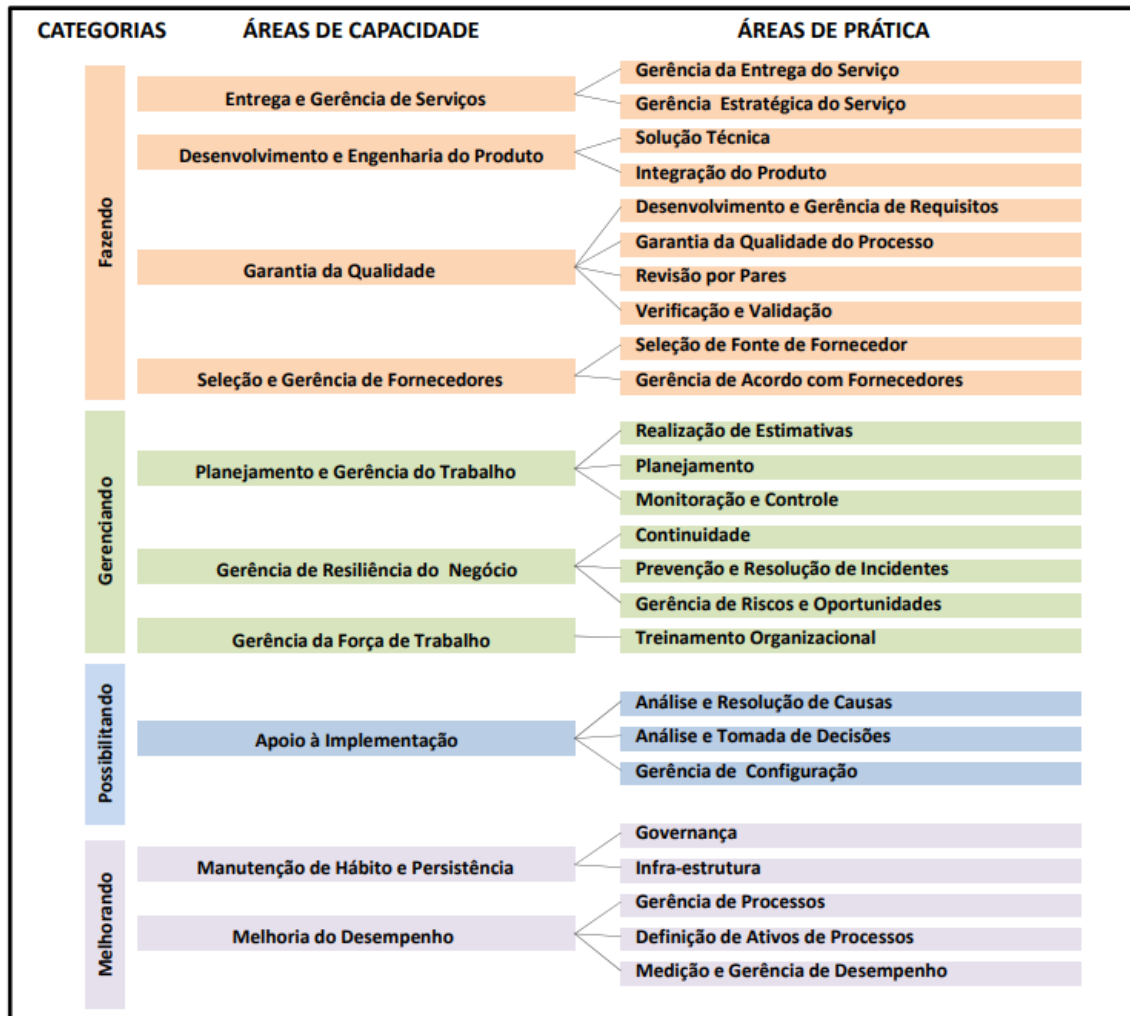
- Modelo criado pelo SEI e mantido pelo CMMI Institute.
- Descreve as boas práticas para melhorar processos e desempenho no desenvolvimento de produtos e serviços.
- Possui um conjunto de práticas que devem ser implementadas para se alcançar um nível de melhoria do processo.

Visões do CMMI

- CMMI Desenvolvimento (CMMI-DEV)
- CMMI Serviços (CMMI-SVC)
- CMMI Gestão de Fornecedor (CMMI-SPM)

CMMI 2.0

- Categorias
 - Áreas de Capacidade
 - Áreas de Prática
 - Práticas



Estrutura do CMMI

- Uma categoria agrupa as áreas de capacidade.
- Uma área de capacidade agrupa as áreas de prática.
- Uma área de prática é um conjunto de práticas.

Categorias

- Fazendo (Doing): contém as áreas de capacidade responsáveis por produzir e entregar soluções de qualidade.
- Gerenciando (Managing): contém as áreas de capacidade responsáveis por planejar e gerenciar a implementação das soluções.

Categorias

- Possibilitando (Enabling): contém a área de capacidade responsável por apoiar a implementação e as entregas.
- Melhorando (Improving): contém as áreas de capacidade responsáveis por manter e melhorar o desempenho dos processos.

Áreas de Capacidade

- Entrega e Gerência de Serviços
 - Gestão da Entrega do Serviço
 - Gestão Estratégica do Serviço
- Desenvolvimento e Engenharia do Produto
 - Solução Técnica
 - Integração do Produto

Áreas de Capacidade

- Garantia da Qualidade
 - Desenvolvimento e Gerência de Requisitos
 - Garantia da Qualidade do Processo
 - Revisão por Pares
 - Verificação e Validação
- Seleção de Gerência de Fornecedores
 - Seleção de Fonte do Fornecedor
 - Gerência de Acordo com Fornecedores

Áreas de Capacidade

- Planejamento e Gerência do Trabalho
 - Realização de Estimativas
 - Planejamento
 - Monitoramento e Controle
- Gerência de Resiliência do Negócio
 - Continuidade
 - Prevenção e Resolução de Incidentes
 - Gestão de Riscos e Oportunidades

Áreas de Capacidade

- Gerência da Força Trabalho
 - Treinamento Organizacional
- Apoio à Implementação
 - Análise e Resolução de Causas
 - Análise e Tomada de Decisões
 - Gerência da Configuração

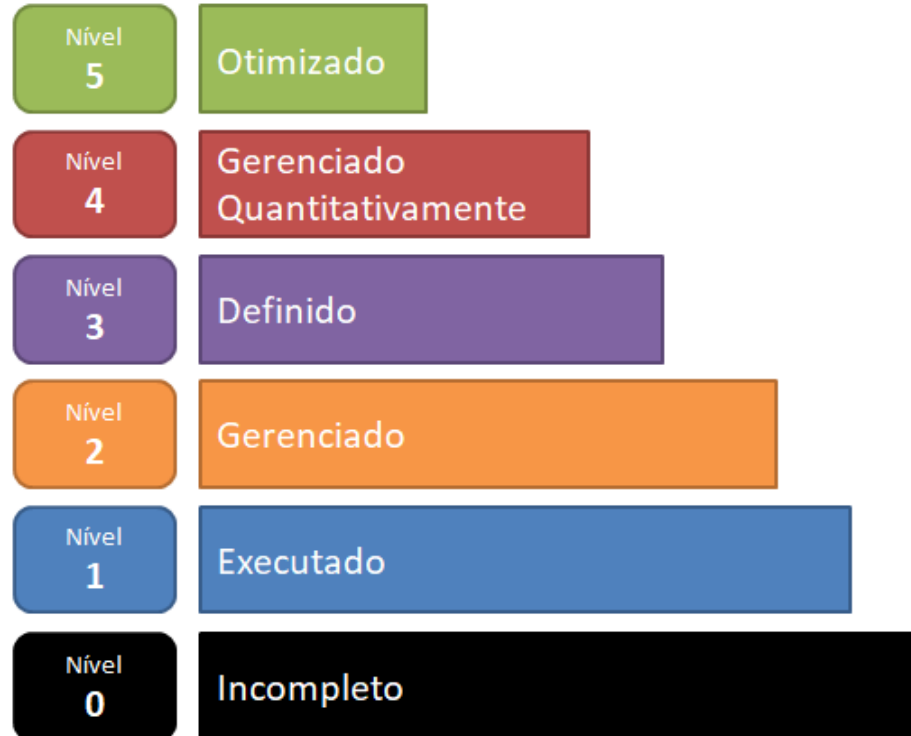
Áreas de Capacidade

- Manutenção de Hábito e Persistência
 - Governança
 - Infraestrutura
- Melhoria do Desempenho
 - Gerência de Processos
 - Definição de Ativos de Processo
 - Medição e Gerência de Desempenho

Capacidade x Maturidade

- Capacidade é o conjunto de conhecimentos, habilidades e proficiências da organização. As práticas são agrupadas em níveis de capacidade compondo um guia de melhoria.
- Maturidade é o estágio ou nível organizacional que uma empresa pode alcançar. O nível de maturidade indica que as práticas estão no nível de capacidade correspondente.

Níveis do CMMI



Nível 0 - Incompleto

- Processo ad-hoc ou não conhecido.
- Trabalho executado de forma aleatória, podendo ou não ser concluído.

Nível 1 - Executado

- Processos são improvisados e caóticos.
- Trabalho realizado de forma reativa.
- Atraso nas entregas e gastos excessivos são comuns.

Nível 2 - Gerenciado

- Processos gerenciados a nível de projeto.
- Os projetos são planejados, realizados, medidos e controlados.
- Maior facilidade em controlar os custos e os prazos dos projetos para entregas de produtos e serviços.

Nível 3 - Definido

- Processos são proativos, bem caracterizados e compreendidos, sendo descritos por padrões estabelecidos e melhorados ao longo do tempo.
- Documentações são padronizadas e as práticas são uniformes.

Nível 4 - Gerenciado Quantitativamente

- Processos são medidos e controlados.
- Qualidade e Desempenho dos processos são compreendidos em termos estatísticos.
- A organização consegue melhorar a qualidade do seu produto/serviço e ter mais controle sobre o andamento dos processos.

Nível 5 - Otimizado

- Processos são estáveis e flexíveis.
- São melhorados continuamente com base na compreensão quantitativa de suas causas comuns de variação.
- Busca alcançar processos que sejam capazes de responder às mudanças e oportunidades.

Controle de aula

Data: 21/12/2022

Horário agendada: 08:30

Horário entrada: 08:30

Horário saída: *(sem informação)*

Professor: JOSE REGINALDO DE SOUSA MENDES JUNIOR

Disciplina: Engenharia de Software

Matriz: Graduação - Análise e desenvolvimento de sistemas

Estúdio: 4 PKS

Observação: *(sem observação)*

| Bloco | Nome da aula | Código | Observação |
|-------|---|-------------|---|
| 01 | U4A1 - Gerenciamento e Planejamento de Projetos de Software | 221221(4)01 | <i>(sem observação)</i> |
| 02 | U4A2 - Teste de Software | 221221(4)02 | <i>(sem observação)</i> |
| 03 | U4A3 - Gerenciamento da Qualidade e Melhoria dos Processos | 221221(4)03 | Última fala: "em relação a desenvolvimento de software, certo?" |
| 04 | U4A4 - Framework CMMI | 221221(4)04 | <i>(sem observação)</i> |

Encerramento

NESTA AULA VIMOS O FRAMEWORK CMMI, UM MODELO DE CAPACIDADE E MATURIDADE QUE APOIA OS PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE NO QUE TANGE A MELHORIA DA QUALIDADE DO PROCESSO.

**INTRODUÇÃO À ENGENHARIA DE
SOFTWARE | UNIDADE 4**

**Aula 1 | Gerenciamento e
Planejamento de
Projetos de Software**

PROFESSOR(A): JOSÉ REGINALDO

Introdução

NESTA AULA VAMOS ABORDAR OS PROCESSOS DE GERENCIAMENTO E PLANEJAMENTO DA ELABORAÇÃO DE UM SOFTWARE.

OBJETIVOS DA AULA

- APRESENTAR A CONCEITUAÇÃO DE PROJETOS DE SOFTWARE E AS BOAS PRÁTICAS DE GERENCIAMENTO DE PROJETOS.

O que é um Projeto?

Um esforço temporário, empreendido para criar um produto, serviço ou resultado único.

[PMI, 2008]

Missão Impossível

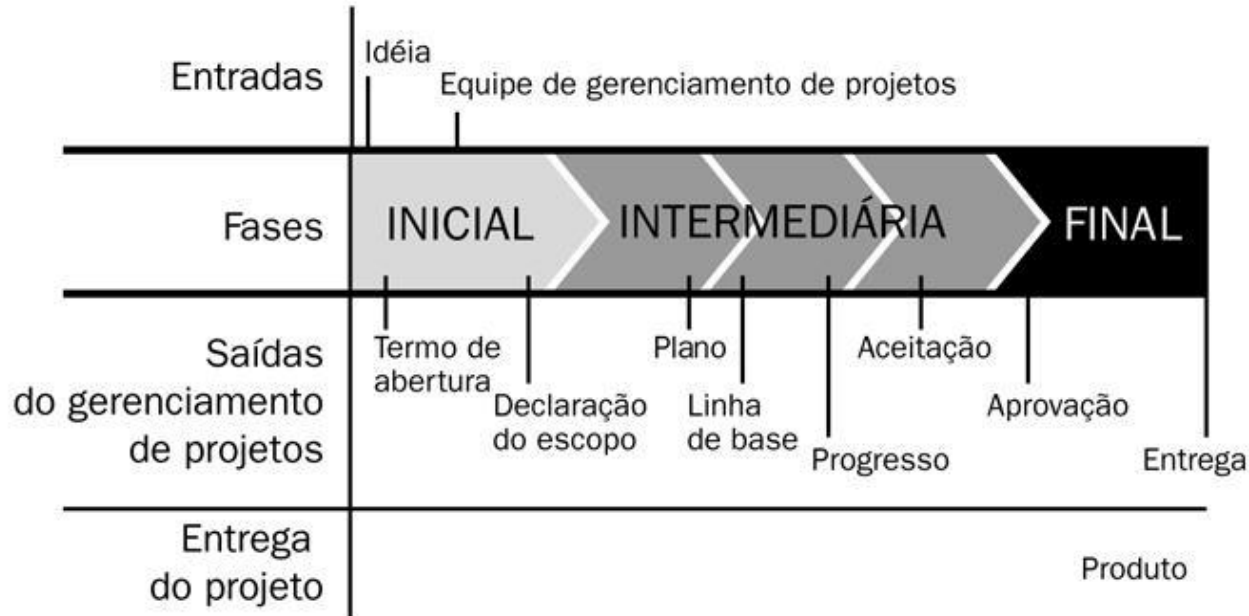


O que é Gerenciamento de Projeto?

Aplicação de conhecimentos, habilidades, ferramentas e técnicas adequadas à condução das atividades do projeto, de forma a atingir seus requisitos.

[PMI, 2008]

Ciclo de Vida do Projeto de Software



Benefícios da Gestão de Projetos

- Maior controle dos processos
- Maior agilidade na tomada de decisões
- Resultados mais assertivos
- Maior engajamento da equipe
- Otimização do tempo
- Maior satisfação do cliente

O Gerente de Projeto

- Controla escopo, prazos e custos do projeto.
- Distribui tarefas à equipe.
- Monitora indicadores e toma decisões.

PMBOK

- Um guia das melhores práticas de gerenciamento de projetos publicado pelo Project Management Institute (PMI).
- Pode ser aplicado a todos os tipos de projetos, independentemente do nicho, da dimensão, do pessoal envolvido, dos prazos e orçamentos.

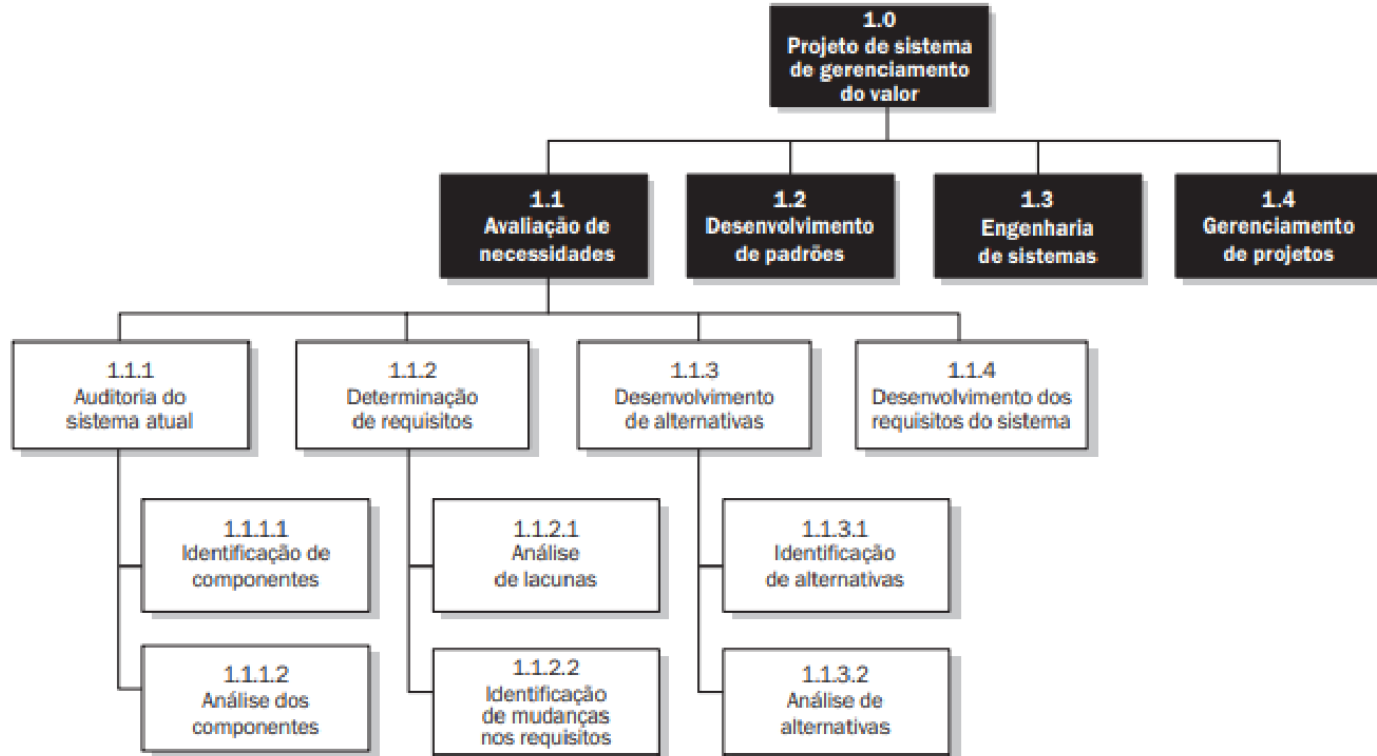
Áreas de Conhecimento do PMBOK

- Gerenciamento de Escopo
- Gerenciamento de Cronograma
- Gerenciamento de Custos
- Gerenciamento de Qualidade
- Gerenciamento de Recursos
- Gerenciamento de Comunicação
- Gerenciamento de Riscos
- Gerenciamento de Aquisições
- Gerenciamento de Stakeholders
- Gerenciamento de Integração

Gerenciamento de Escopo

- Trata da descrição formal do que precisa ser feito e entregue no projeto.
- O escopo é visualizado na forma de EAP (Estrutura Analítica do Projeto).

Estrutura Analítica do Projeto



Gerenciamento de Cronograma

- Cuida do monitoramento e controle dos prazos relacionados a cada etapa e ao projeto como um todo.
- O Gráfico de Gantt permite visualizar e acompanhar as tarefas a serem executadas ao longo do projeto em uma perspectiva do tempo.

Gerenciamento de Custos

- Cuida da questão financeira, para que o projeto não ultrapasse o orçamento que foi definido inicialmente.
- Usa planilhas e dashboards que apoiam a visualização dos gastos e orçamento.

Gerenciamento de Qualidade

- Garante que um controle de qualidade seja implementado para que o projeto alcance um nível aceitável de qualidade.

Gerenciamento de Recursos

- Gerencia os recursos necessários para a condução do projeto, como recursos humanos, materiais, equipamentos e infraestrutura necessária.
- A Matriz de Responsabilidade serve para atribuir responsabilidades às pessoas quanto às tarefas a serem desenvolvidas.

Matriz de Responsabilidades

R

A

C

I

RESPONSÁVEL

AUTORIDADE

CONSULTADO

INFORMADO

Quem é designado para trabalhar nessa atividade?

Quem tem a autoridade para tomar a decisão?

Quem deve ser consultado e participar da decisão da atividade no momento que for executada?

Quem deve receber a informação de que uma atividade foi executada?

Gerenciamento de Comunicação

- Assegura que a informação do projeto seja distribuída, coletada, armazenada e gerenciada.

Gerenciamento de Riscos

- Trata os riscos do projeto tomando medidas que levem a identificação, análise e planejamento de respostas para os pontos de vulnerabilidade do projeto.
- A Matriz de Impacto e Probabilidade é usada para avaliar os impactos dos riscos sobre cada um dos processos existentes nas áreas de conhecimento, bem como avaliar sua probabilidade de ocorrência.

Matriz de Impacto e Probabilidade

| | | | | |
|---------------|-------|-------|-------|-------|
| Impacto | Alto | Média | Alta | Alta |
| | Médio | Baixa | Média | Alta |
| | Baixo | Baixa | Baixa | Média |
| | | Baixo | Médio | Alto |
| Probabilidade | | | | |

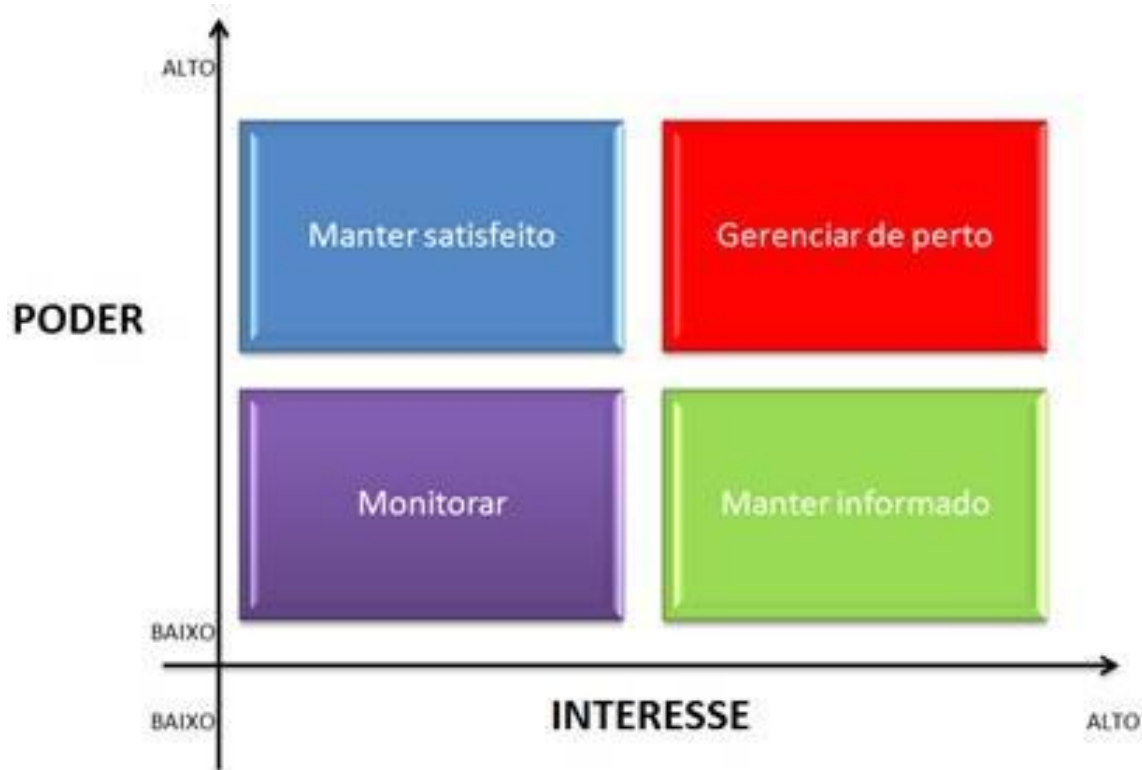
Gerenciamento de Aquisições

- Auxilia no processo de tomada de decisão de compra.
- Tratar do relacionamento com os fornecedores.
- Estabelece normas para comprar produtos e contratar serviços externos à equipe do projeto.

Gerenciamento de Stakeholders

- Identifica todas as pessoas e entidades que podem afetar o projeto ou serem afetadas por ele.
- Utiliza a Matriz de Poder x Interesse, que permite classificar os stakeholders para que torne mais fácil o entendimento sobre o fluxo de informações que devem transitar entre eles.

Matriz Poder x Interesse



Gerenciamento de Integração

- Junta todas as demais áreas do projeto para atingir um objetivo comum, coordenando todas as práticas e demais áreas.

Encerramento

NESTA AULA ABORDAMOS A CONCEITUAÇÃO DE PROJETO DE SOFTWARE, E DISCUTIMOS AS BOAS PRÁTICAS DO MODELO PMBOK QUE DEVEM SER SEGUIDAS PARA O GERENCIAMENTO DE PROJETOS.