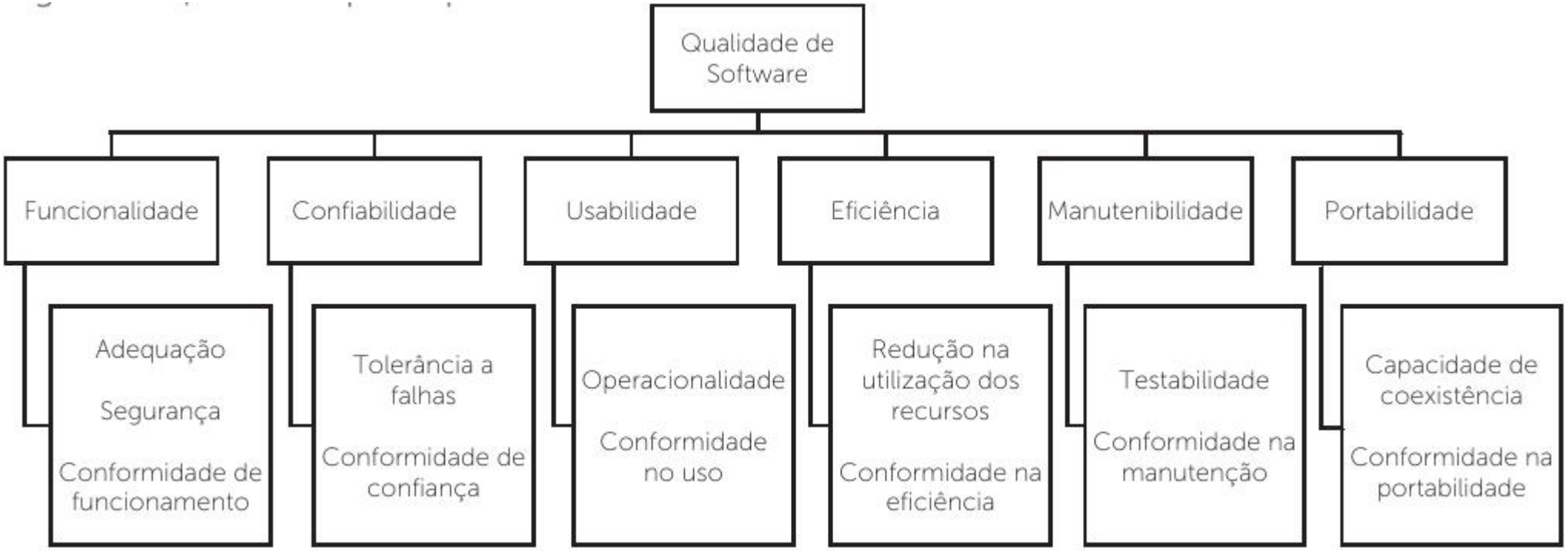
4.1 Qualidade de software

4.2 Melhorias de processo de desenvolvimento de software

Gerenciamento da qualidade do projeto

- Se aplica a todos os projetos, independente da natureza do produto



Conceitos e impactos dos atributos

- Funcionalidade
  - É descrita todas as funções que um determinado desenvolvimento deve efetuar, conforme estabelecido no levantamento de requisitos. Sendo divididos em:
    - Adequação: observa se as funcionalidades do software estão adequadas às necessidades do usuário
    - Acurácia: descreve a capacidade de precisão nas saídas fornecidas pelo software
    - Interoperabilidade: mede a capacidade de coexistência com outros sistemas, dentro de um mesmo domínio
    - Segurança: deve garantir o nível de acesso e permissões dos usuários e garantir que pessoas mal-intencionadas não acessem as informações do sistema

- Confiabilidade: é descrita a capacidade de tolerância a falhas, ou seja, o sistema tem que garantir que o desempenho do sistema se manterá dentro do esperado no projeto. Está subdividida em:
  - Maturidade: demonstra a capacidade de estar e permanecer livre de falhas (utiliza-se a palavra em língua estrangeira “bug”, para se referir a falhas).
  - Tolerância a falhas: descreve a capacidade do sistema, mesmo após a ocorrência de uma falha (independente da causa), permanecer em funcionamento.
  - Recuperabilidade: mede o tempo de recuperação do sistema após a ocorrência de falha.

- Usabilidade: neste quesito como o usuário conseguirá utilizar o software. Além da capacidade de operação, o desenvolvimento deve ter um layout atraente. Sendo subdividido em:
  - Inteligibilidade: compreende a forma com que o usuário entenderá e identificará as funções no sistema
  - Operacionalidade: descreve a facilidade de operação do usuário, mesmo na ocorrência de falhas
  - Atratividade: demonstra como o layout pode tornar o sistema intuitivo para o usuário, auxiliando na compreensão e na operação das funcionalidades

- Eficiência: é medido o consumo dos recursos no menor tempo de execução (conforme o projeto e a capacidade física), sendo dividido em:
  - Tempo: afere o tempo resposta das funcionalidades do desenvolvimento
  - Utilização dos recursos: são medidos os recursos consumidos para realizar as tarefas do sistema e a capacidade de não comprometer o desempenho de outros sistemas rodando em paralelo

- Manutenibilidade: são definidas a capacidade de modificação, melhorias, correção de falhas, entre outra qualquer necessidade de se alterar o código do desenvolvimento. Está subdividida em:
  - Analisabilidade: descreve a capacidade de identificar as falhas do sistema
  - Modificabilidade: identifica como o sistema deve se comportar ao sofrer alterações, independente dos motivos ou necessidades
  - Testabilidade: descreve a capacidade de testar o sistema, após este sofrer uma modificação ou falha

- Portabilidade: mede como o sistema pode ser transferido, adaptado em outros ambientes e infraestruturas. Devendo ser levado em consideração as características, como capacidade de hardware, acesso a recursos de comunicação e idioma. Está dividida em:
  - Coexistência: permite identificar como o software convive com outras aplicações, sem que ocorra conflitos ou falha de ambos os sistemas
  - Adaptabilidade: representa a forma que o sistema vai poder ser utilizado em diferentes sistemas operacionais, banco de dados e hardware

- Cada projeto possui características próprias, portanto, não deve ser utilizada “receita de bolo” em gerenciamento de projetos, ou seja, não significa que determinada ação que funcionou em um projeto vai ocorrer em outro

- Todos os atributos relacionados à gestão da qualidade de software requerem uma abordagem no projeto

- No entanto, cada projeto pode necessitar de maior atenção, em um ou mais atributos da qualidade, devido às características e às necessidades de cada projeto.

- O custo da qualidade se refere ao custo total de todos os esforços relacionados à qualidade

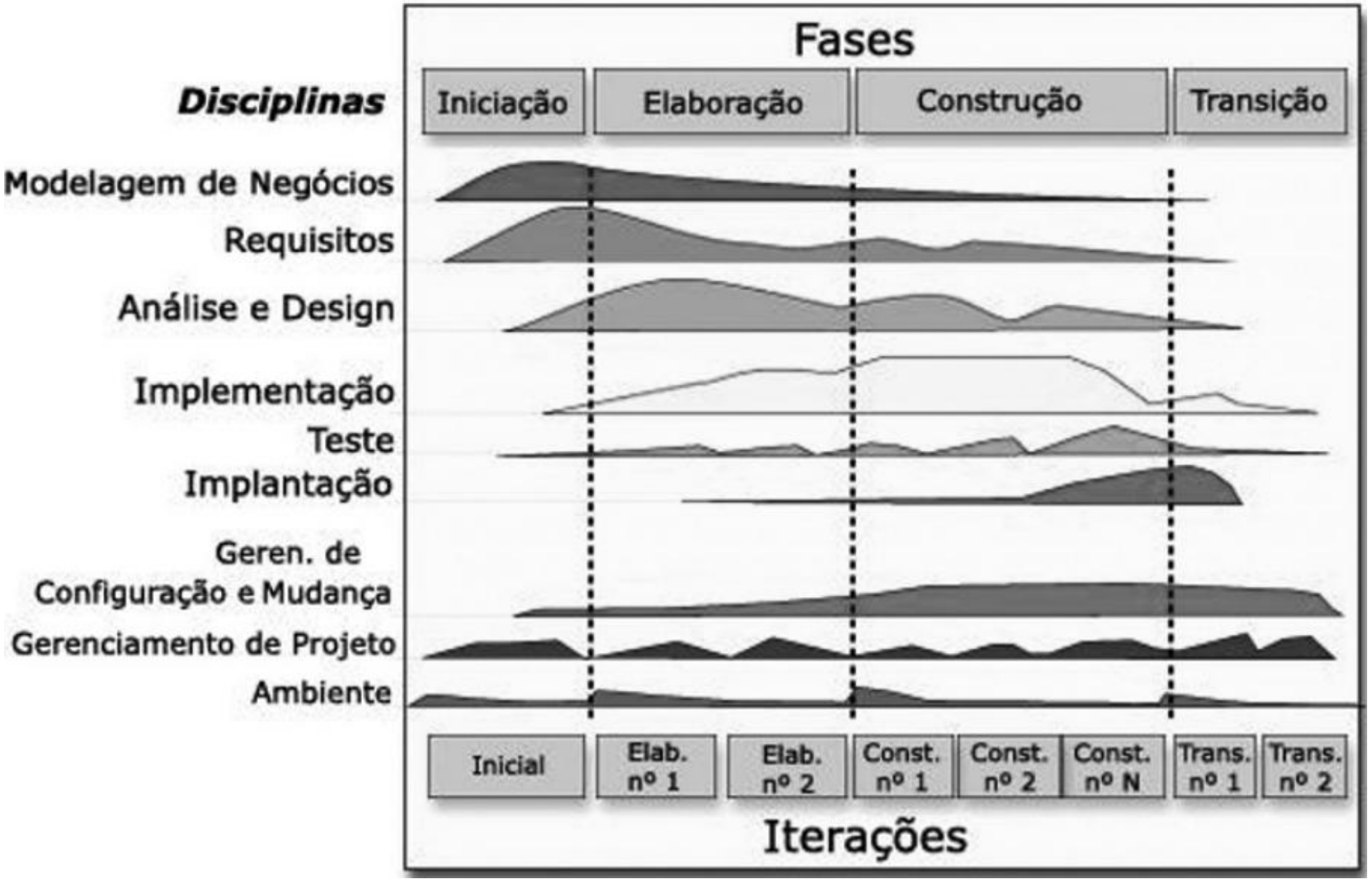
- Nesse contexto, vale a pena lembrar os custos relacionados à recall que as empresas são obrigadas por lei a realizar, além da devolução de lotes de produtos com defeitos ou falhas

RUP (*Rational Unified Process*)

- Ferramenta para auxiliar na condução e monitoramento das atividades
- A utilização de metodologias faz com que exista uma padronização nos desenvolvimentos de softwares, sendo possível estruturar e desenvolver um formato para os processos
- A arquitetura RUP foi baseada no UML (Unified Modeling Language) e estabelecida como padrão nos desenvolvimentos orientado a objetos

- RUP tem como objetivo garantir que as estruturas dos processos possam ser adaptadas para uma produção do software com alta qualidade e que atenda às necessidades das organizações

- O seu desenvolvimento foi apoiado nas áreas de conhecimento e iteração das disciplinas do PMBOK


Fases
Disciplinas
Iniciação
Elaboração
Construção
Transição
Modelagem de Negócios
Requisitos
Análise e Design
Implementação
Teste
Implantação
Geren. de Configuração e Mudança
Gerenciamento de Projeto
Ambiente
Inicial
Elab. nº 1
Elab. nº 2
Const. nº 1
Const. nº 2
Const. nº N
Trans. nº 1
Trans. nº 2
Iterações

RUP - fases

- Iniciação
  - Levantamento de requisitos
  - Detalhamento do escopo do projeto
  - Documentação do acordo entre as partes interessadas
  - Todas as necessidades do projeto devem ficar bem definidas
  - Pode ser desenvolvido um protótipo para aprovação do patrocinador do projeto
- Elaboração
  - Definição das atribuições de cada desenvolvedor do projeto
  - O GP consulta documentações de projetos anteriores para buscar referência de confiabilidade, custos e qualidade

RUP - fases

- Construção
  - Desenvolvimento de fato
  - Também são efetuados os testes e a validação do software
- Transição
  - Última fase do projeto
  - O produto final é entregue
  - São efetuados ajustes quando necessário

RUP

- O grande atrativo da metodologia RUP é fazer com que os ciclos do projeto sejam iterativos e incrementais, conforme o projeto avança em busca do produto final

Quadro 4.1 | Exemplo de diferenças entre PMBOK/RUP

PMBOK	RUP
Iniciação	Iniciação
Planejamento	Elaboração
Execução	Construção
Controle e Monitoramento	
Encerramento	Transição

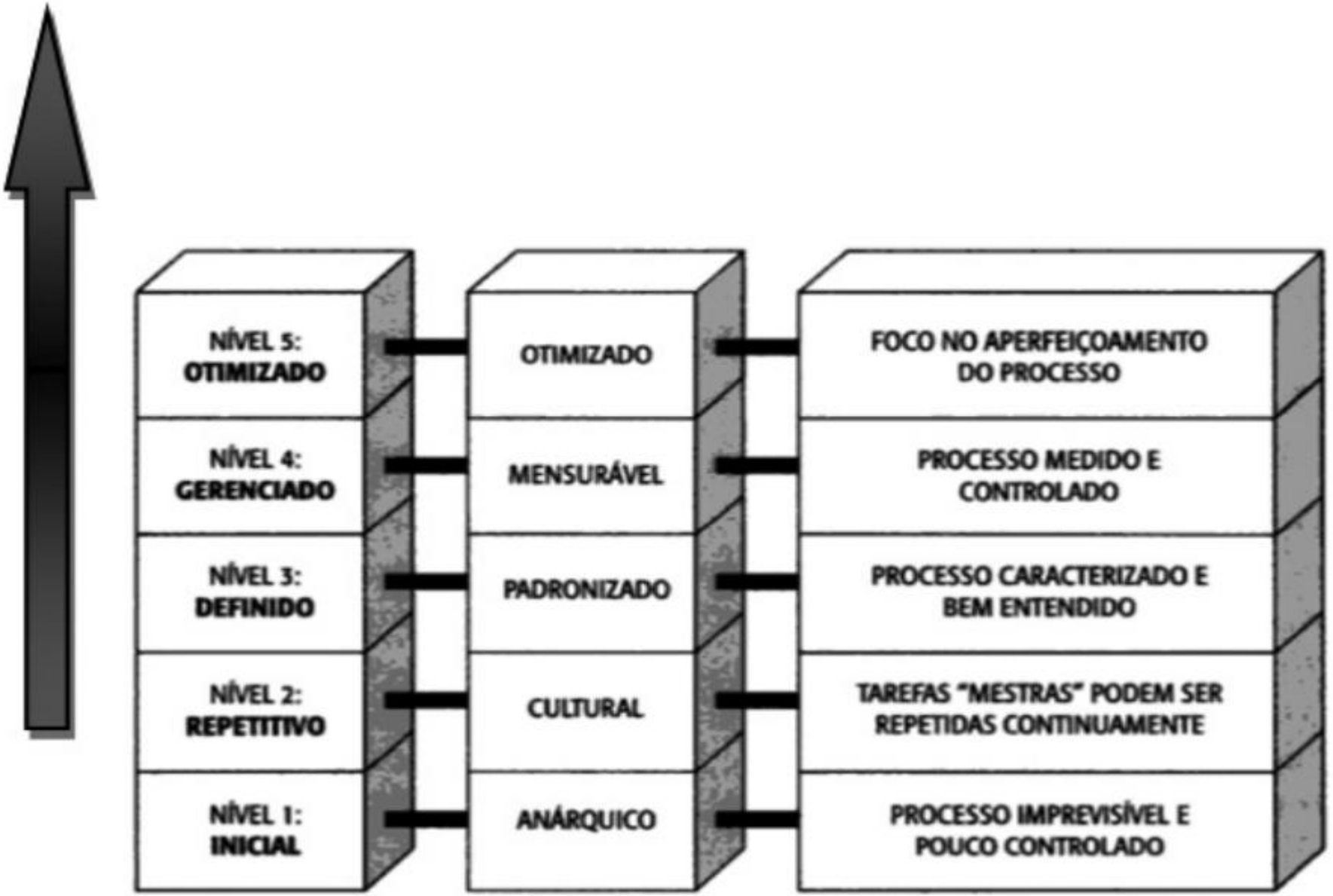
CMM (Modelo de Maturidade em Capacitação)

- Tem como conceito um modelo evolutivo de maturidade, em que o projeto inicia sem nenhum controle dos processos, para gradativamente ir aumentando a eficiência, principalmente nos processos tidos como mais críticos no desenvolvimento

- A metodologia apresenta cinco níveis de maturidade, em que cada organização possui uma particularidade no controle dos seus processos

- A partir daí o nível individual de maturidade é encaixado dentro de um nível, e este só consegue evoluir dentro da metodologia, assim que cumpre os requisitos dentro do nível

Figura 4.3 | Metodologia CMM



CMM - níveis

- Nível I (Inicial)
  - Poucos ou nenhum processo estão definidos dentro da organização, as práticas utilizadas pelos desenvolvedores, são esforços individuais
- Nível II (Repetitivo)
  - O objetivo desse nível é tornar os processos corporativos. Permitindo, assim, que as melhores práticas adotadas pela organização, sejam repetidas sistematicamente

- Nível III (Definido)
  - Nesse nível, os processos padronizados no nível II são documentados e integrados a qualquer processo de software existente na organização (se houver um processo definido)
  - O objetivo é auxiliar os gerentes de projetos e desenvolvedores no ganho de produtividade
- Nível IV (Gerenciado)
  - Os processos para desenvolvimento de software são medidos para a garantia da qualidade do software
  - O objetivo é fazer uma medição quantitativa para a garantia dos padrões de qualidade adotados nos níveis mais baixos

- Nível V (Otimizado)
  - Esse nível visa monitorar o desenvolvimento do software, dentro da metodologia adotada
  - O objetivo é a garantia contínua na qualidade dos processos, por meio das métricas de qualidade e produtividade

4.2 Melhorias de processo de desenvolvimento de software

- As melhorias de processo de desenvolvimento de software não podem ser predefinidas para atender as diferentes necessidades das organizações
- A maneira com que as empresas tratam os seus processos de desenvolvimento, pode variar conforme o grau de formalidade, o porte da organização e o tipo de aplicação
- Não existe um formato padrão (template) para a melhoria dos processos, cabendo ao GP adaptar os procedimentos conforme as necessidades do projeto

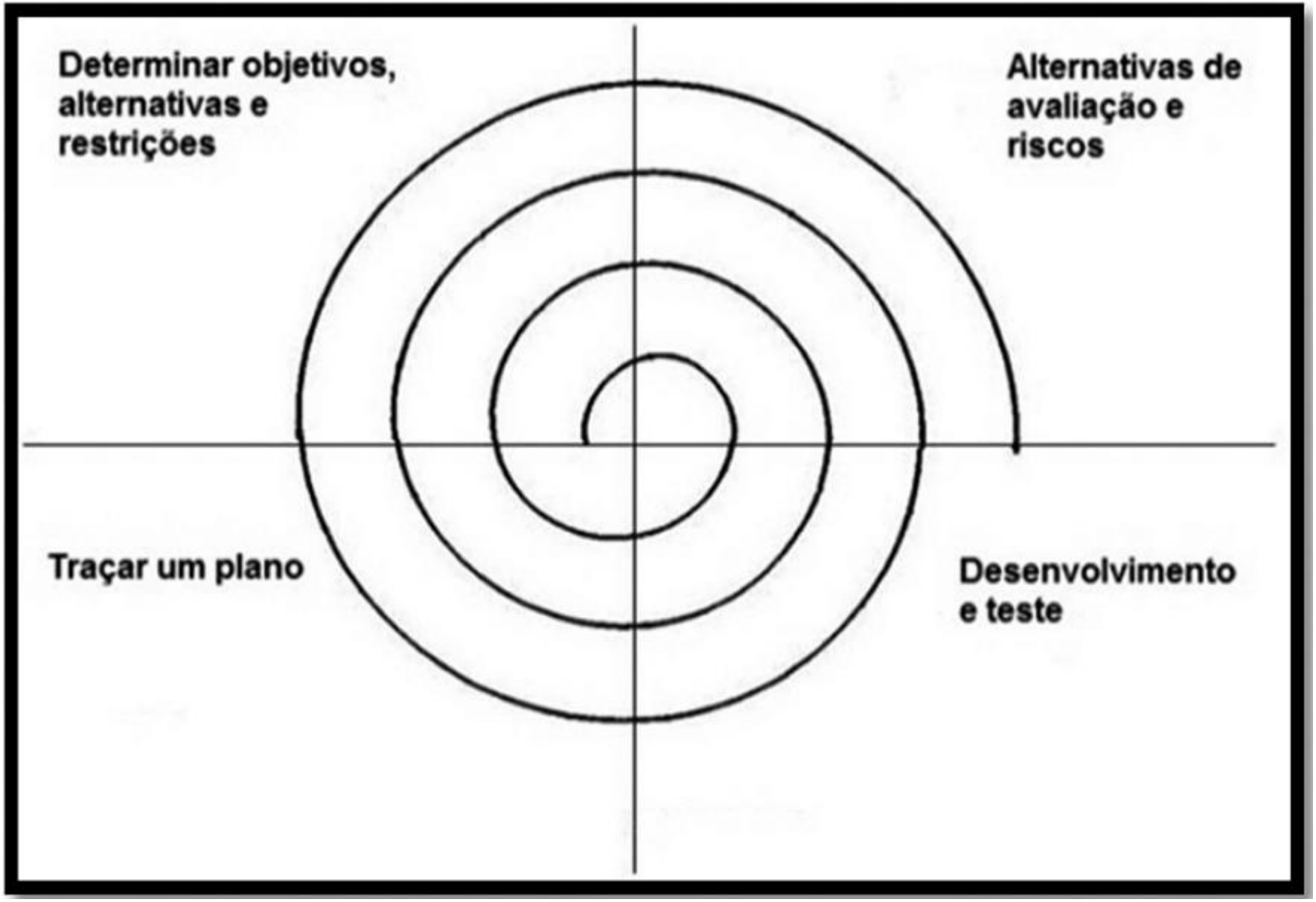
Dois modelos de processos mais utilizados

- Espiral
- Cascata

Espiral

- Dividido em quatro seções:
  - (1) Objetivos, alternativas e restrições
  - (2) Alternativas para avaliação de riscos
  - (3) Desenvolvimento e teste
  - (4) Planejamento

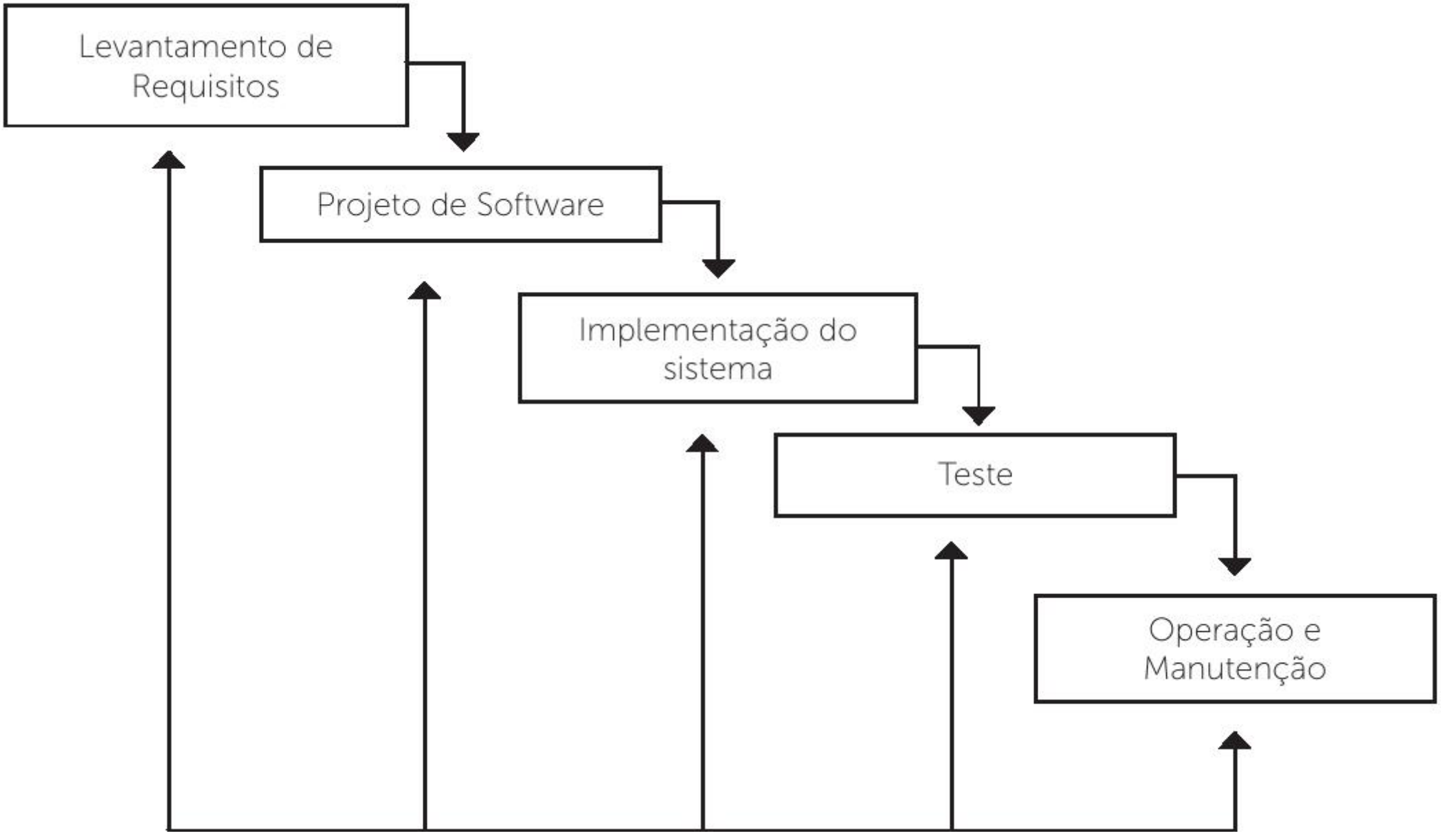
Espiral


Determinar objetivos, alternativas e restrições
Alternativas de avaliação e riscos
Traçar um plano
Desenvolvimento e teste

Espiral

- (1)) são determinados os objetivos, as funcionalidades que o desenvolvimento deve possuir, qual deve ser o desempenho ao se utilizar a aplicação
- (2) os riscos são avaliados, por meio de análise das informações obtidas na primeira seção, prototipação do software e simulações (quando o projeto possibilita)
- (3) ocorre o desenvolvimento do software de fato
- (4) é efetuado o planejamento para a próxima etapa do desenvolvimento, e assim sucessivamente

Cascata


Levantamento de Requisitos
Projeto de Software
Implementação do sistema
Teste
Operação e Manutenção

MPS.BR (Melhoria de Processos do Software Brasileiro)

- Utilização indicada para pequenas e médias empresas, devido ao seu baixo custo de implantação
- O modelo possui sete níveis de maturidade

Nível	Nível de Maturidade	Processos
G	Parcialmente gerenciado	Gerência de requisitos.
		Gerência de projetos.
F	Gerenciado	Aquisição.
		Gerência de configuração.
		Gerência de portfólio de projetos.
		Garantia de qualidade.
E	Parcialmente definido	Treinamento.
		Avaliação e melhoria de processos.
		Definição de processos.
		Adaptação do processo para gerência de projetos.
D	Largamente definido	Desenvolvimento de requisitos.
		Solução técnica.
		Integração do produto.
		Instalação do produto.
		Liberação do produto.
		Validação.
		Verificação.
C	Definido	Análise de decisão e resolução.
		Gerência de riscos.
B	Gerenciado quantitativamente	Desempenho do processo.
		Gerência quantitativa do produto.
A	Otimização	Inovação e implantação.
		Análise de causas e resolução.

PDCA

- Ferramenta para controle de qualidade do desenvolvimento
  - PLAN (Planejar)
    - Determinar as entradas e as saídas, as formas e os métodos, como os processos serão executados e mapeados às necessidades de melhorias
  - DO (Fazer)
    - Execução de todo o planejamento efetuado na fase anterior
    - Documentar a forma como serão executadas as atividades do projeto
  - CHECK (Verificar)
    - Análise do desenvolvimento
    - Métricas utilizadas podem variar conforme o projeto
  - ACT (Agir)
    - Caso tudo tenha ficado dentro dos padrões, o processo é documentado
    - Em caso de falhas nas métricas analisadas, são feitas as correções necessárias

Melhoria contínua

- Conjunto de ações que buscam aperfeiçoar os processos de desenvolvimento, para garantir a qualidade dos projetos
  - Além de utilizar processos que atendam às necessidades, deve-se melhorá-los continuamente
- Deve-se utilizar metodologias de desenvolvimento, a fim de se garantir que as tarefas sejam bem definidas, passem por avaliações e medições, e os processos sejam estruturados
- Ex.:
  - Melhoria contínua do produto
  - Melhoria contínua do software

Melhoria contínua do produto

- O software já está desenvolvido e em uso
- Manutenção
  - Determinar as correções das falhas e identificar por meio de documentação a manutenibilidade do desenvolvimento
- Refatoração
  - Descrever os processos envolvidos para se modificar um código de software
- Software Analytics
  - Refere-se à análise de desempenho e atendimento às necessidades

Melhoria contínua do software

- Processos relacionados ao desenvolvimento da aplicação
- Reuniões
  - Encontros periódicos a fim de se verificar os requisitos do desenvolvimento para efetuar as correções necessárias
- Lições aprendidas
  - Experiências positivas e negativas vivenciadas em outros projetos devem servir de base para consulta e referência
- Medições
  - Aferições realizadas, podendo variar a funcionalidade medida ou a métrica, conforme o projeto