

# Engenharia de Software

## Processo de Software

UPF - Universidade de Passo Fundo

Engenharia de Software

Prof. Jeangrei Veiga

[jeangrei@upf.br](mailto:jeangrei@upf.br)



# Engenharia de Software

Uma disciplina que oferece Métodos,  
Técnicas e Ferramentas para desenvolver  
e manter softwares com alta qualidade  
para a resolução de problemas  
(Anneliese Mayrhauser 1990)



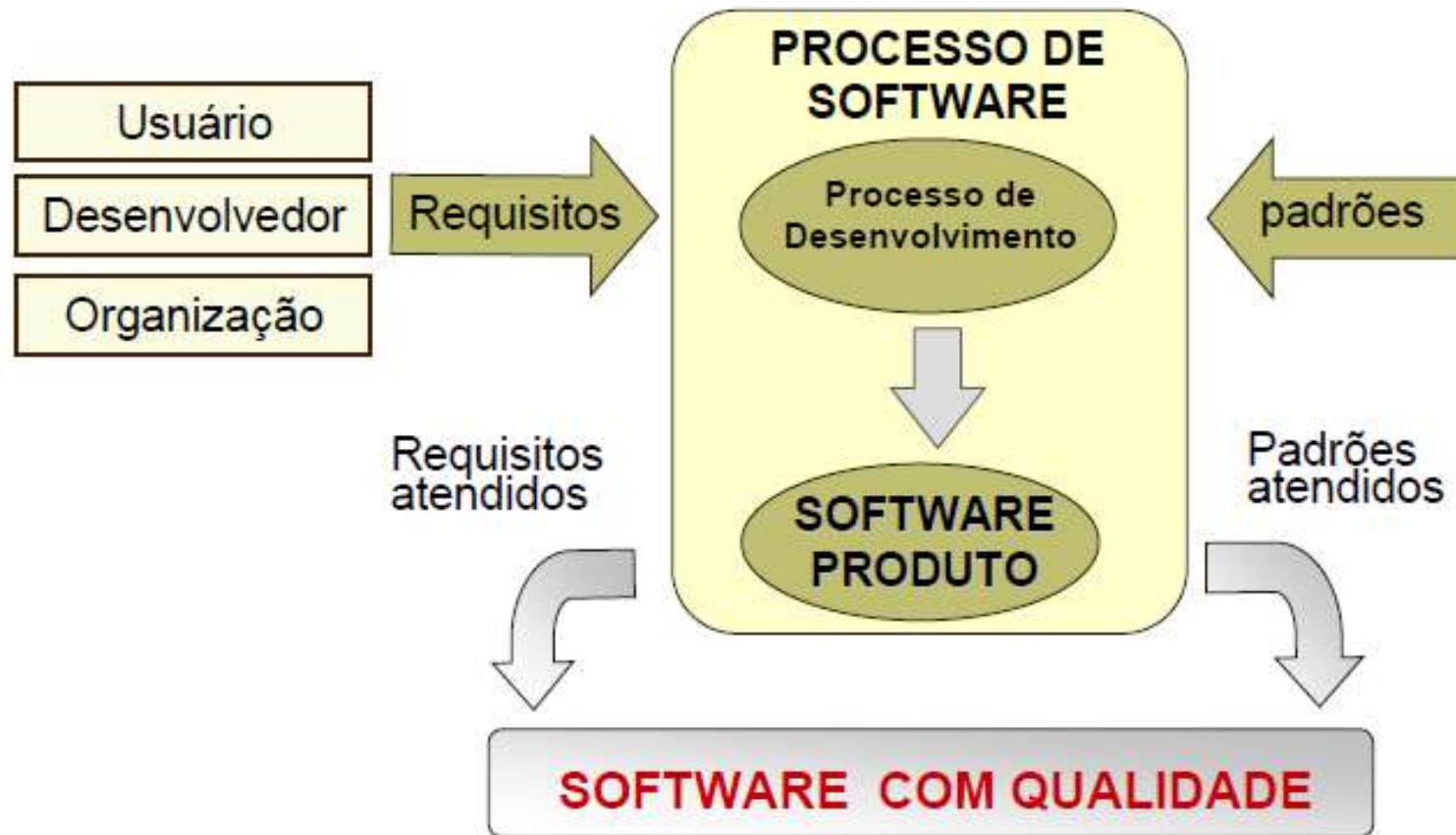
# Engenharia de Software

**ENGENHARIA DE SOFTWARE:** compreende um conjunto de *etapas* que envolve MÉTODOS, FERRAMENTAS e PROCEDIMENTOS.

➡ Essas etapas são citadas como **CICLOS DE VIDA** ou **MODELOS DE PROCESSO DE SOFTWARE**

➡ Uma estratégia de desenvolvimento que englobe processos, métodos e ferramentas, e as fases de desenvolvimento

# Visão Profissional de Qualidade



# Engenharia de Software

A engenharia de software é uma tecnologia em camadas. Como mostra a figura:



*Figura: Camadas da Engenharia de Software, Fonte: Pressmam*



# O PROCESSO

# O Processo

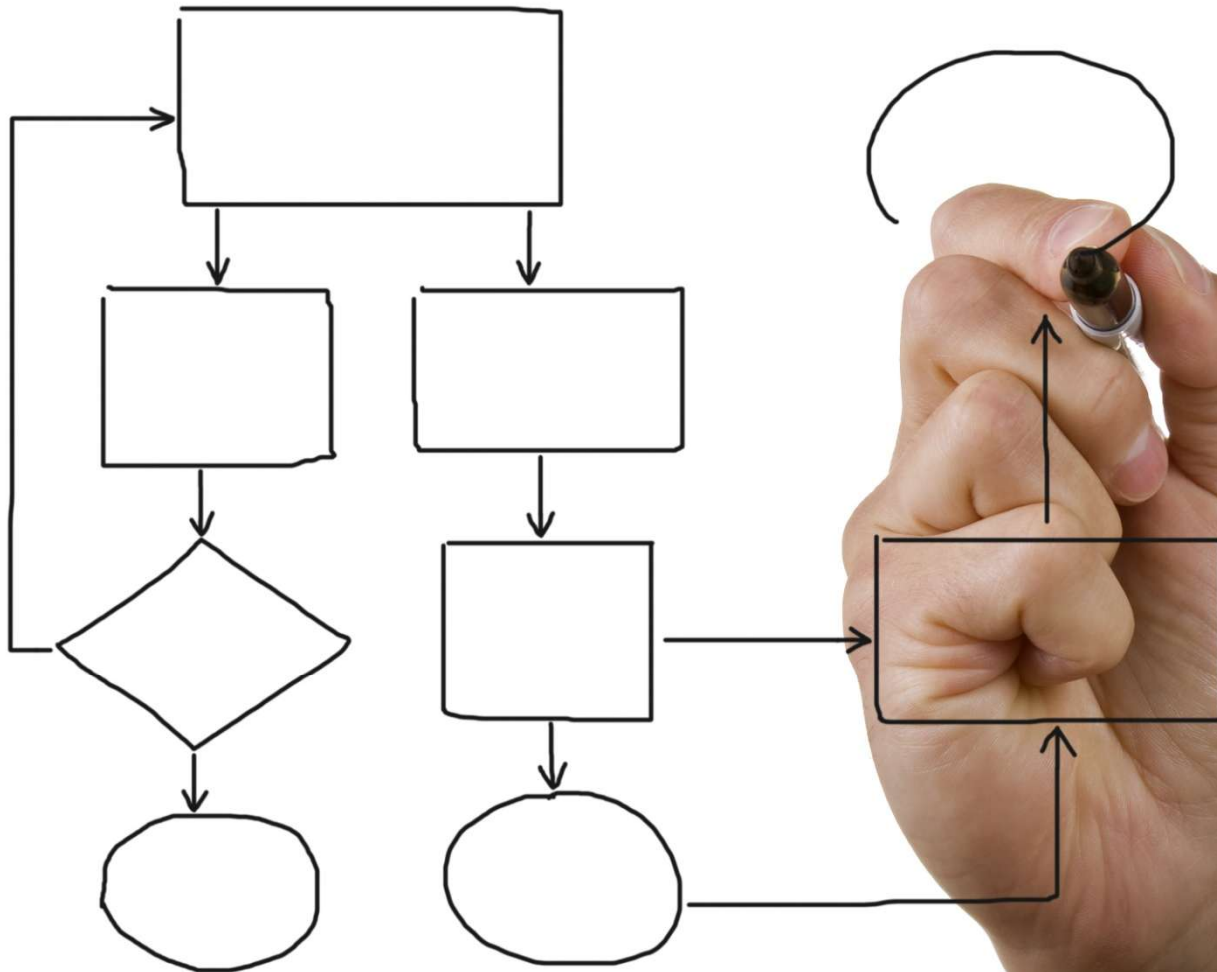


O processo define um arcabouço que deve ser estabelecido para a efetiva utilização da tecnologia de engenharia de software.

Os processos de software formam a base para o controle gerencial dos projetos de software e estabelecem o contexto no qual os métodos técnicos são aplicados, os produtos de trabalho são produzidos, os marcos são estabelecidos, a qualidade é assegurada e as modificações são adequadamente geridas.



# O Processo







# Processo de Software

## O que é?

Quando você cria softwares é importante percorrer uma série de passos presivíveis. O roteiro que você segue é chamado de processo de software.

## Quem faz?

Os engenheiros de software e seus gerentes adaptam um processo a suas necessidades e depois o seguem. Além disso, o pessoal que solicitou o software tem um papel a desempenhar no processo de defini-lo, construí-lo e testá-lo.

<https://www.youtube.com/watch?v=QPir8jTMLdI>



# Processo de Software

## Por que é importante?

Porque fornece estabilidade, controle e organização para uma atividade que pode, se deixada sem controle, tornar-se bastante caótica.

## Quais são os passos?

O processo de software a ser adotado depende do software que está em construção. Um processo poderia ser apropriado à criação de softwares para um sistema de aviônica de uma aeronave, enquanto um processo inteiramente diferente seria indicado para a criação de um site.

# ISO/IEC 12207

- A **ISO/IEC 12207** é a norma **ISO/IEC** que define **processo de desenvolvimento de software**. A norma internacional *ISO/IEC 12207* tem como objetivo principal estabelecer uma estrutura comum para os processos de ciclo de vida de **software** visando ajudar as **organizações** a compreenderem todos os **componentes** presentes na aquisição e fornecimento de software e, assim, conseguirem firmar **contratos** e executarem projetos de forma **mais eficaz**.
- Um **processo** é uma seqüência de passos realizados para um determinado propósito [*IEEE, 1990*]; o **processo de software** envolve *métodos, técnicas, ferramentas e pessoas*. Um processo pode ser descrito de duas formas: por **propósito ou resultado** e **por atividade**.



# ISO/IEC 12207

- Para isso deve adotar as **premissas**:
  - ❖ Que **procedimentos e métodos** serão usados para a execução das atividades;
  - ❖ Que **ferramentas e equipamentos** suportarão a realização das atividades, de forma a simplificar e *automatizar o trabalho*;
  - ❖ Qual o **perfil adequado** de quem irá executar as atividades e qual o **treinamento requerido** nos *procedimentos, métodos, ferramentas* para que se possam realizar as atividades de forma adequada;
  - ❖ Quais as **métricas de processo** que poderão ser empregadas para que a execução do processo possa ter a *qualidade avaliada*.



# Descrição por Propósito ou Resultado

---

- É utilizada quando não há necessidade de detalhar o processo, apenas indicar o objetivo e o resultado. Essa abordagem poderá ser utilizada na avaliação do processo em relação aos modelos de maturidade de software como, por exemplo, o modelo **CMMI** e o modelo da **ISO/IEC 15504 (SPICE)**.



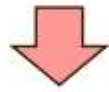
# Descrição por Atividade

---

- É a abordagem mais conhecida e intuitiva. Nela são descritas as atividades com as inter-relações e o algoritmo de execução de cada atividade. As atividades devem atingir o propósito do processo.

# Modelos de Processo de Software

- Uma das **maiores dificuldades** encontradas pelas empresas de software é o **gerenciamento** de seus **processos de software**.



**Modelos de Processo de Software**





# Modelos de Processo de Software

---

- Um **modelo de processo** procura descrever formalmente e de maneira organizada todas as atividades que devem ser seguidas para a obtenção segura de um produto de software.
- É importante escolher um **modelo apropriado** às metas da organização e saber em que **grau** esse modelo será implementado.





# Um Processo de Software Imaturo

---

- **Ad hoc**; processo improvisado por profissionais e gerências.
- **Não é** rigorosamente **seguido** e o cumprimento não é **controlado**.
- Altamente **dependente** dos profissionais atuais.
- **Baixa visão** do progresso e da qualidade.



# Um Processo de Software Imaturo

---

- A **qualidade** do produto pode ficar **comprometida** para que prazos sejam cumpridos.
- **Arriscado** do ponto de vista do uso de **nova tecnologia** e novos padrões.
- Custos de **manutenção** excessivos.
- Qualidade **difícil de prever**.



# Um Processo de Software Maduro

---

- **Coerente** com as linhas de ação, o trabalho é efetivamente concluído.
- Definido, documentado e melhorando constantemente:
  - compreendido
  - utilizado
  - vivo e ativo



# Um Processo de Software Maduro

---

- Com o **apoio** visível da alta administração e outras gerências.
- **Bem controlado** - fidelidade ao processo é objeto de auditoria e de controle.
- São utilizadas medições do produto e do processo.
- Uso disciplinado da tecnologia.

# MODELOS DE MATURIDADE DE PROCESSO







# Maturidade de Processo de Software

---

- A **qualidade do processo de software** pode ser analisada por intermédio do nível de maturidade do processo .
- A **maturidade dos processos de software** de uma organização influencia a sua capacidade de atingir metas de custo, qualidade e cronograma.



# Valor da Melhoria de Processo

---

- A **melhoria do processo** de software oferece um retorno no investimento que pode ser medido - **quando é medido**.
- **Benefícios adicionais** são intangíveis e não podem ser quantificados facilmente.
- O **CMM** é uma ferramenta útil para orientação no processo de melhoria. Porém muito complexa para ser implementada.



# Capability Maturity Model (CMM)

---

- Modelo de Maturidade de Processo de Software que auxilia as empresas a melhorar seus **processos** de software.
- 1986 - Software Engineering Institute (SEI) por solicitação do *DoD* que necessitava de um método de avaliar a capacidade de seus fornecedores de software.
- 1991 - 1ª versão do CMM (Versão 1.0)
- 1993 - Versão 1.1 do CMM (atualmente usada)





# Capability Maturity Model (CMM)

---

- O primeiro **ponto importante** a se observar no modelo CMM:
  - ❖ É um modelo que se refere ao **processo** pelo qual o produto é construído.

# CMMI (Capability Maturity Model Integration)

- O **CMMI** é um modelo de referência que contém práticas (*Genéricas ou Específicas*) necessárias à maturidade em disciplinas específicas (*Engenharia de Sistemas (SE), Engenharia de Software (SE), Desenvolvimento de Processo e Produto Integrado (IPPD), Supplier Sourcing (SS)*). Desenvolvido pelo SEI (*Instituto de Engenharia de Software*).
- O **CMMI** é uma evolução do **CMM** e procura estabelecer um modelo único para o processo de melhoria corporativo, integrando diferentes modelos e disciplinas.



# CMMI

---

- Supõe-se que uma organização que possui um processo maduro tem **maiores probabilidades** de produzir consistentemente bons produtos do que uma outra cujo processo é imaturo e caótico.



# CMMI

---

- O CMMI é um modelo de **referência** para a qualidade do processo de produção de software.
- Por meio de um **processo formal de avaliação**, uma organização é classificada em um “**nível de maturidade**” que varia de um a cinco.



# CMMI

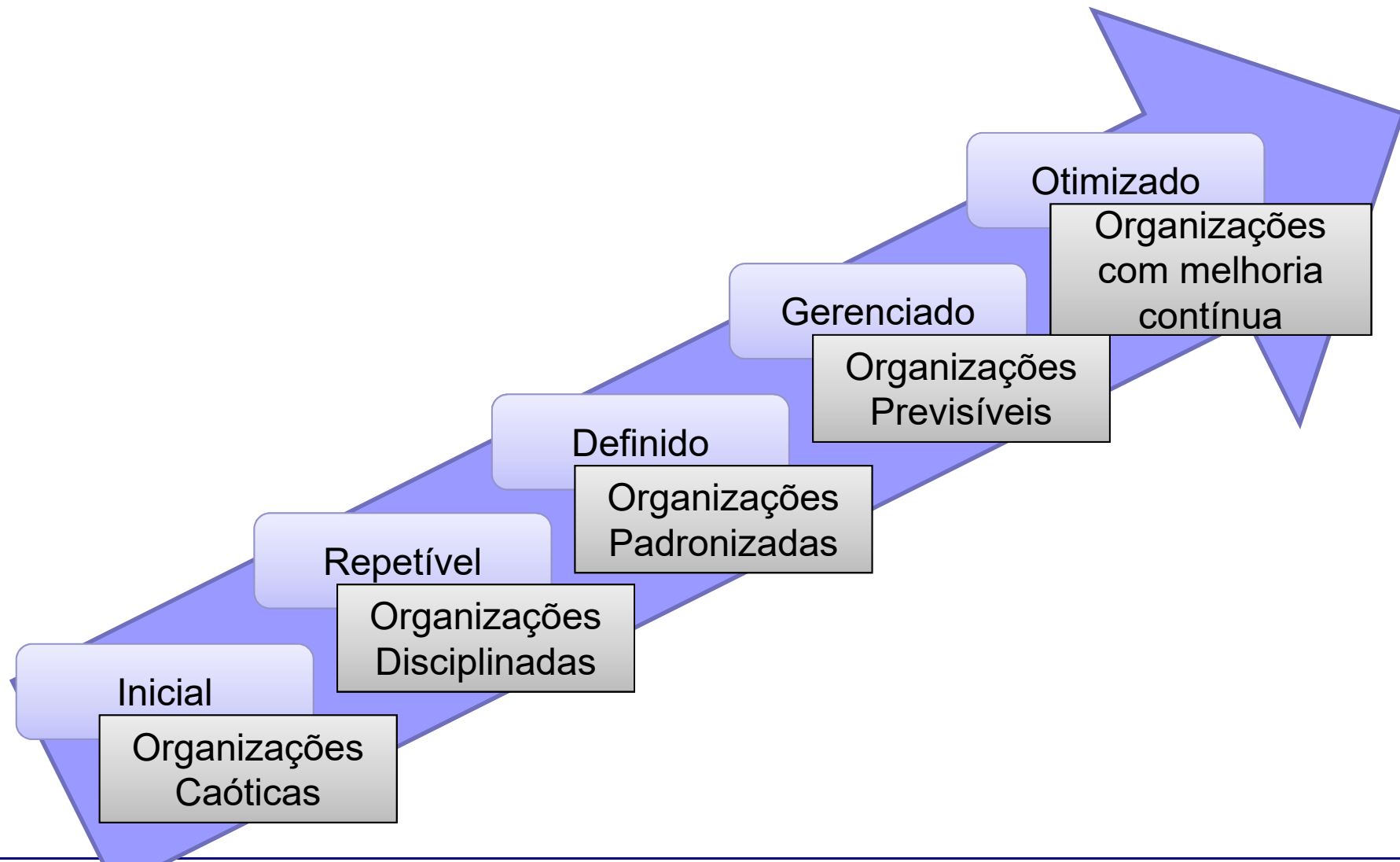
---

- O nível de maturidade indica em que medida os processos daquela organização **são maduros**.
- Quanto maior o nível de maturidade, melhores e mais maduros são os processos.

<https://www.youtube.com/watch?v=C2BFOEvtu3w>




# Os 5 Níveis de Maturidade do CMMI

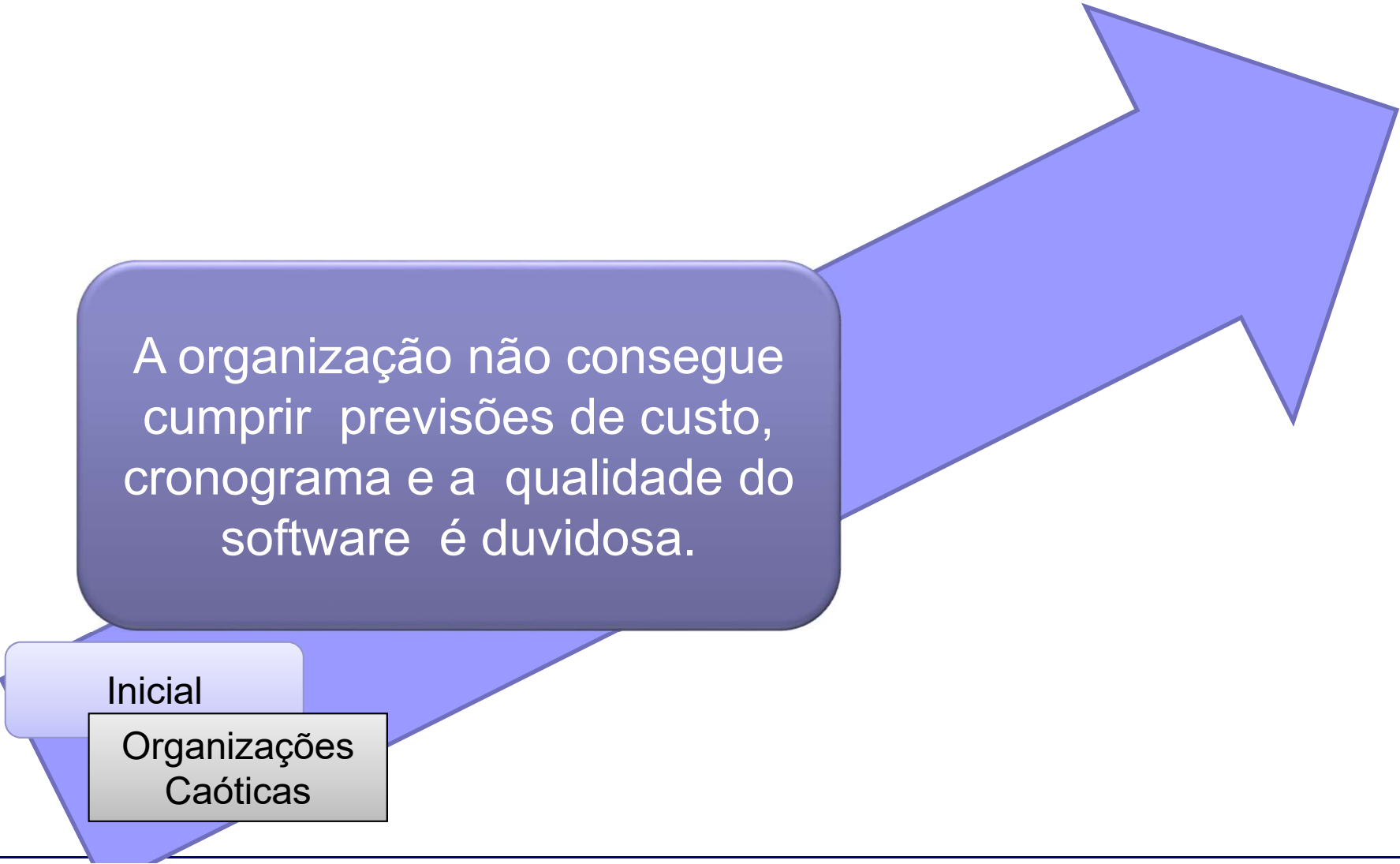




# Os 5 Níveis de Maturidade do CMMI

- 
- Os 5 Níveis do CMMI são organizados de modo a estabelecer as prioridades na condução de programas de melhoria do processo de software.
  - Cada Nível é considerado como pré-requisito para o nível seguinte.
  - Não é possível o “salto” de níveis.
  - Cada nível se apóia nas competências que a organização desenvolveu no nível imediatamente inferior.

# Os 5 Níveis de Maturidade do CMMI



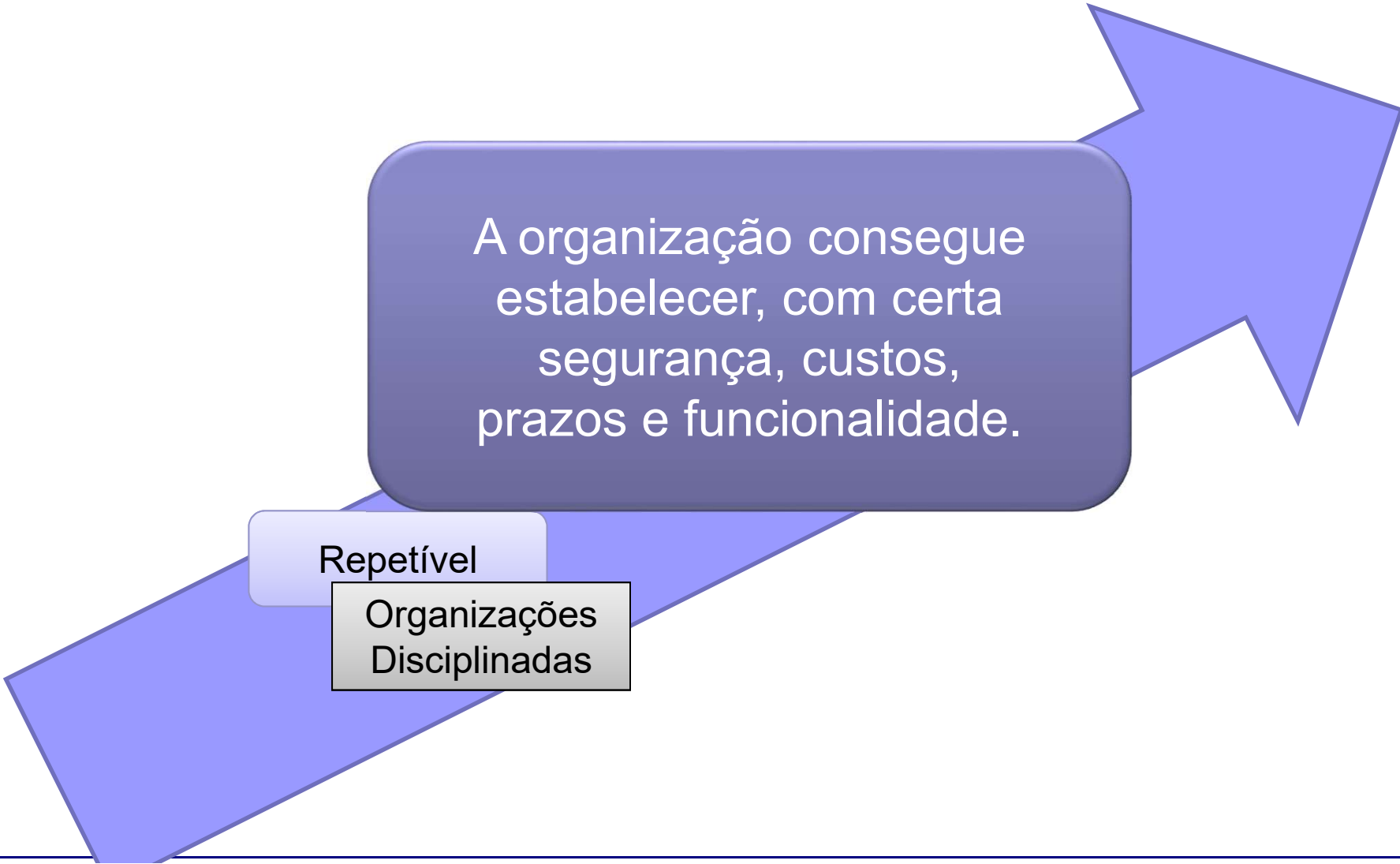
A organização não consegue cumprir previsões de custo, cronograma e a qualidade do software é duvidosa.

Inicial

Organizações  
Caóticas



# Os 5 Níveis de Maturidade do CMMI



A diagram illustrating the 5 levels of CMMI maturity. It features a large, light blue arrow pointing upwards and to the right. A dark blue rounded rectangle is positioned over the arrow, containing the text: 'A organização consegue estabelecer, com certa segurança, custos, prazos e funcionalidade.' Below this rectangle, the word 'Repetível' is written in a light blue box, and 'Organizações Disciplinadas' is written in a grey box. The entire diagram is set against a white background with a dark blue horizontal bar at the top.

A organização consegue estabelecer, com certa segurança, custos, prazos e funcionalidade.

Repetível

Organizações  
Disciplinadas

# Os 5 Níveis de Maturidade do CMMI

The diagram illustrates the progression of CMMI maturity levels. It features a large, light blue, upward-pointing arrow that serves as a background. On this arrow, there are three distinct boxes. The top box is a dark purple rounded rectangle containing text about cost, schedule, functionality, and quality control. Below it is a light blue rounded rectangle with the word 'Definido'. At the bottom is a grey rectangle with the text 'Organizações Padronizadas'.

Custo, cronograma e funcionalidade estão sob controle e a qualidade do software é acompanhada.

Definido

Organizações  
Padronizadas

# Os 5 Níveis de Maturidade do CMMI

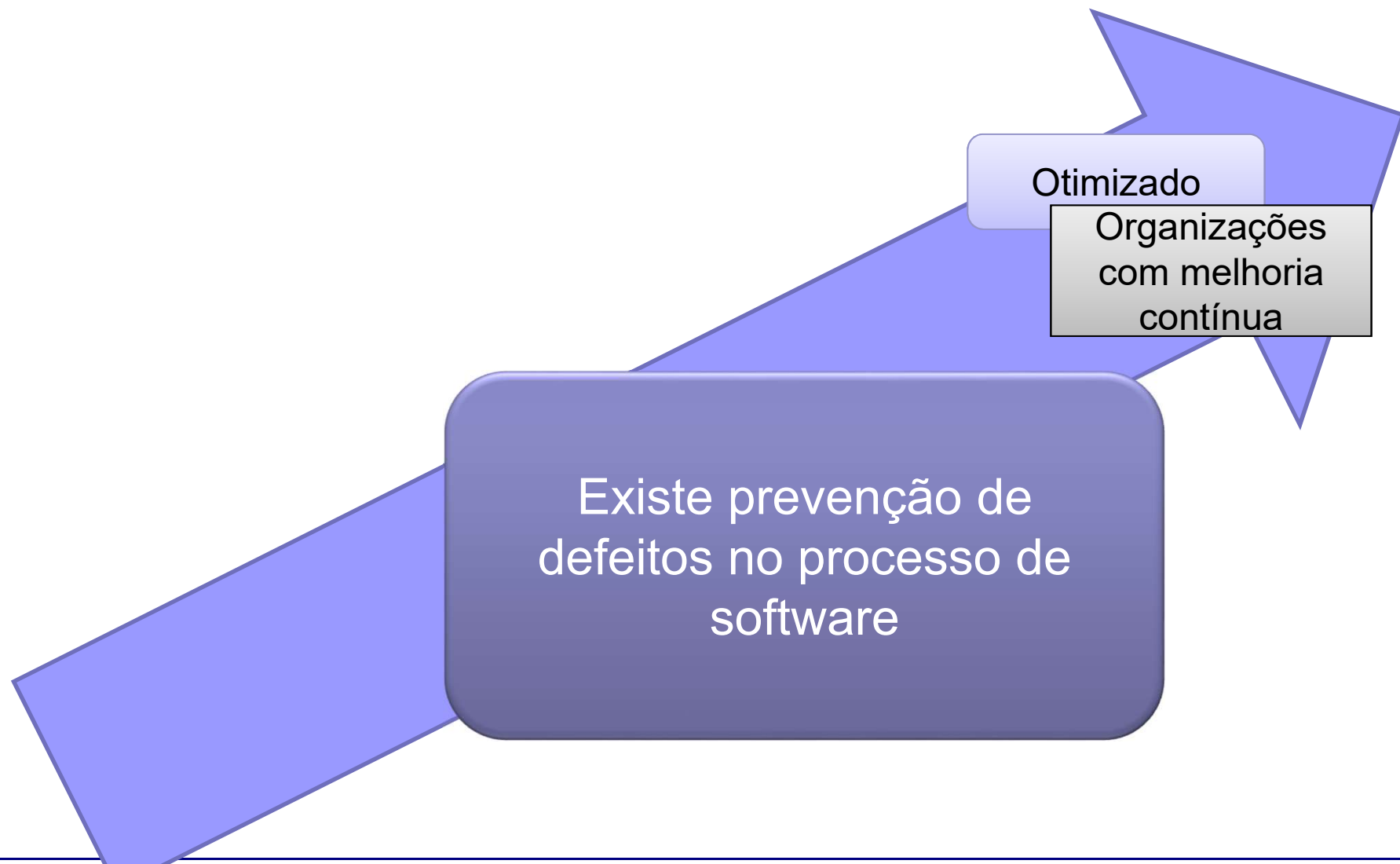
Existe controle de qualidade sobre produtos e processos.

Gerenciado

Organizações  
Previsíveis

- Os processos de software, tanto para atividades administrativas quanto para de engenharia estão documentados, padronizados e integrados em um processo de software padrão para a organização.
- Todos os projetos usam uma versão aprovada do processo de software padrão da organização para desenvolvimento e manutenção de software.

# Os 5 Níveis de Maturidade do CMMI

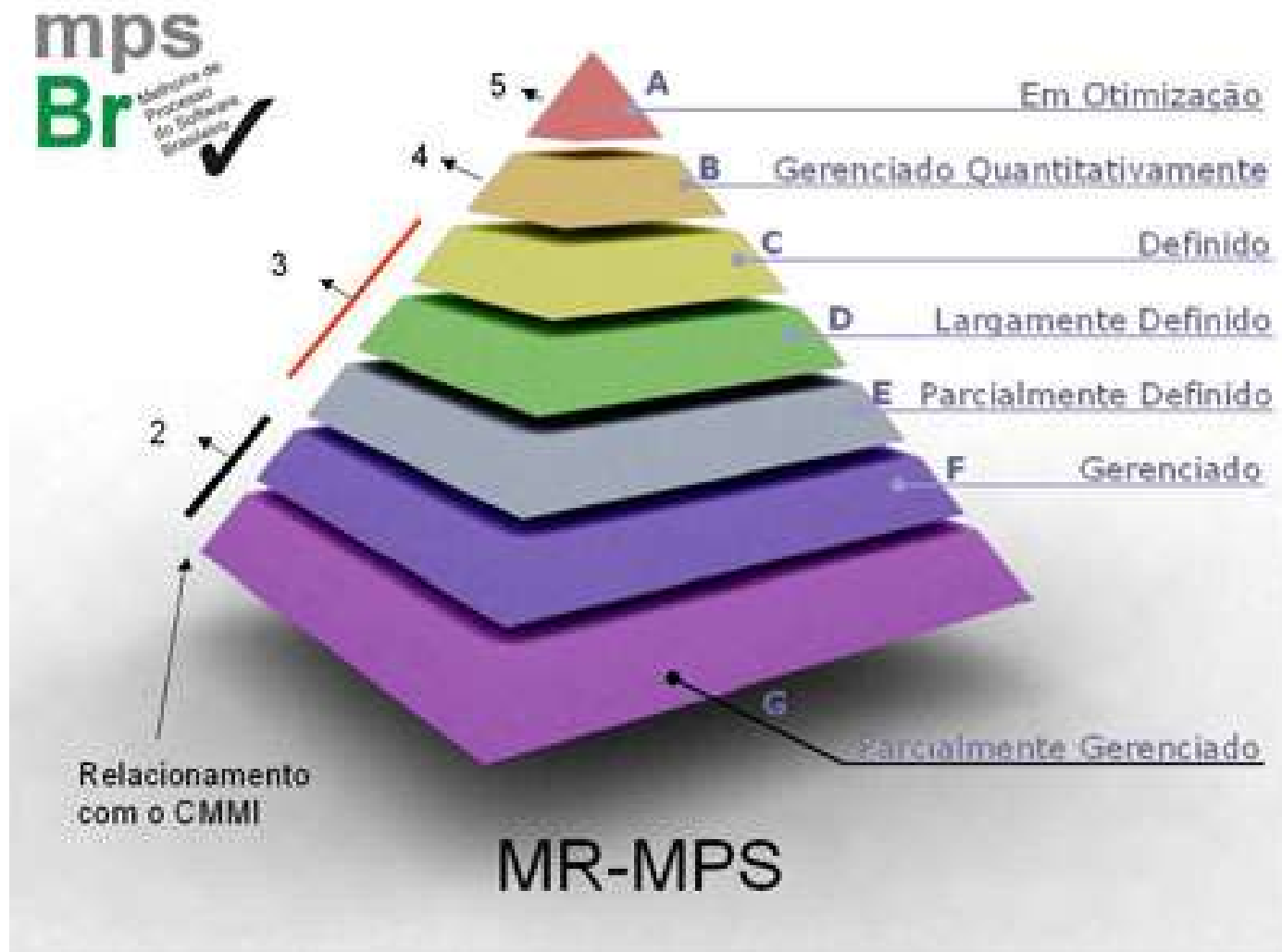


# Melhoria de Processo de Software - MPS

- **Melhoria do Processo de Software Brasileiro**, é simultaneamente um movimento para a melhoria e um modelo de qualidade de processo voltada para a realidade do mercado de pequenas e médias empresas de desenvolvimento de software no Brasil.
- Ele é baseado no **CMMI**, nas normas **ISO/IEC 12207** e **ISO/IEC 15504** e na realidade do mercado brasileiro.
- No Brasil, uma das principais vantagens do modelo é seu custo reduzido de certificação em relação as normas estrangeiras, sendo ideal para micro, pequenas e médias empresas.
- Um dos objetivos do projeto é replicar o modelo na América Latina, incluindo o Chile, Argentina, Costa Rica, Peru e Uruguai.

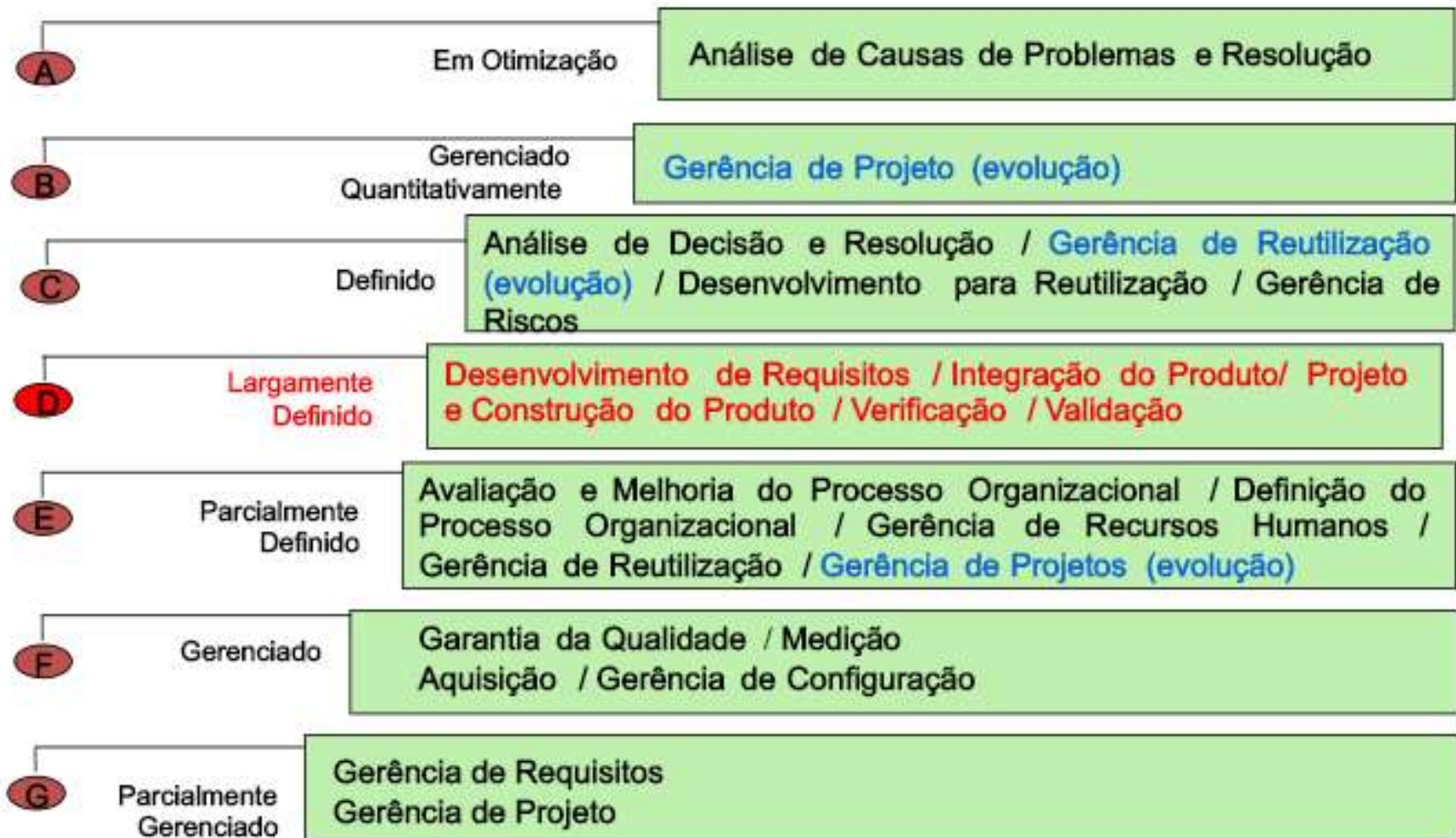
<http://pt.wikipedia.org/wiki/MPS.BR>

# Melhoria de Processo de Software - MPS

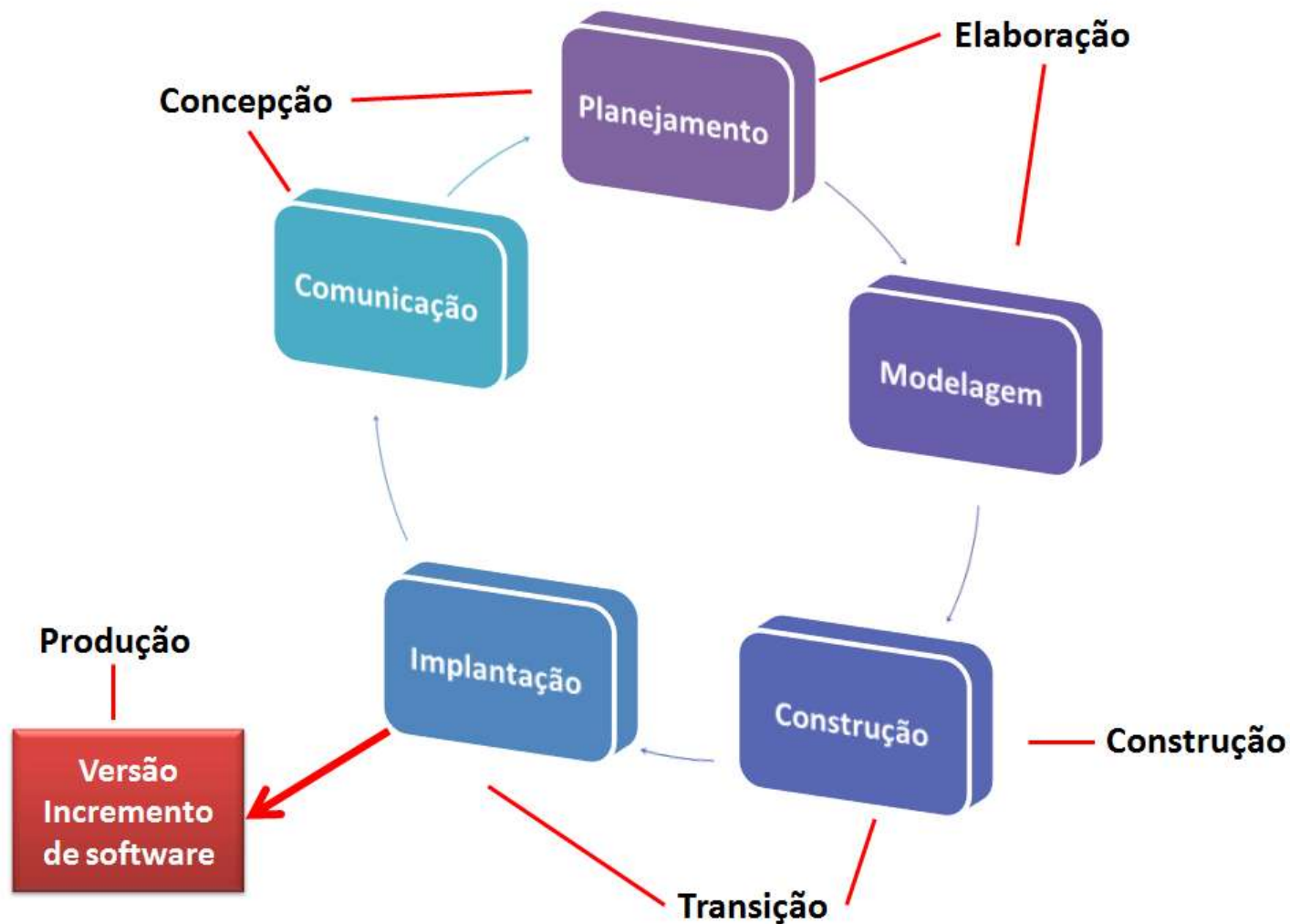




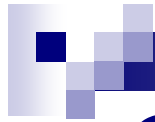
# Melhoria de Processo de Software - MPS



# Atividades do Arcabouço Genérico







# Comunicação

Essa atividade de arcabouço envolve alta colaboração e comunicação com o cliente e outros interessados e abrange o levantamento de requisitos e outras atividades relacionadas



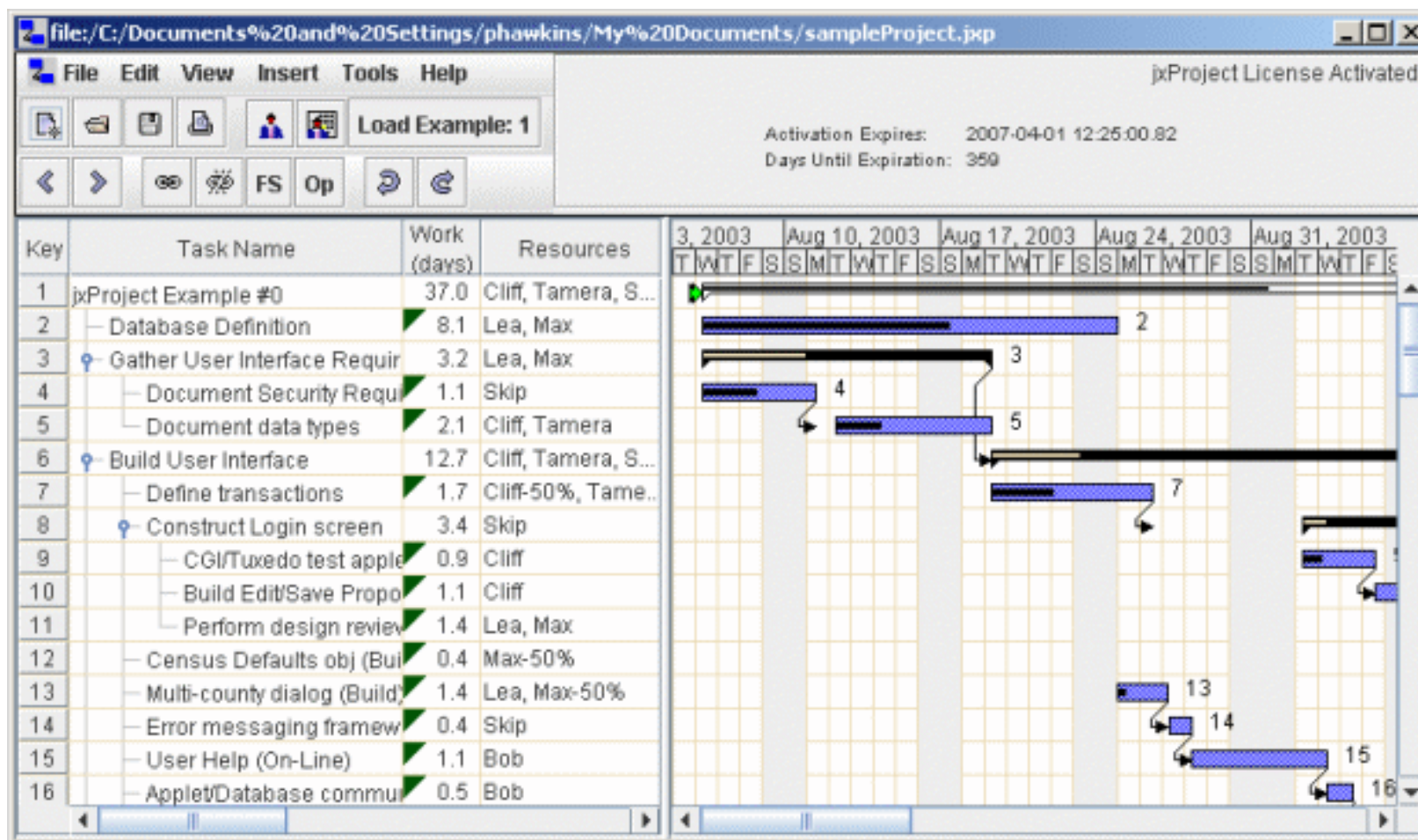
# Planejamento

Essa atividade estabelece um plano para o trabalho de engenharia de software que se segue.

Descreve as tarefas técnicas a ser conduzidas, os riscos prováveis, os recursos que serão necessários, os produtos de trabalho a ser produzidos e um cronograma de trabalho.

Essa atividade de arcabouço envolve alta colaboração e comunicação com o cliente e outros interessados e abrange o levantamento de requisitos e outras atividades relacionadas

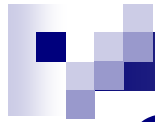
# Planejamento





# Modelagem

Essa atividade inclui a criação de modelos que permitam ao desenvolvedor e ao cliente, entender melhor os requisitos do software e o projeto que vai satisfazer a esses requisitos.



# Construção

Essa atividade combina a geração de código (quer manual ou automática) e os testes necessários para revelar erros no código.





# Implantação

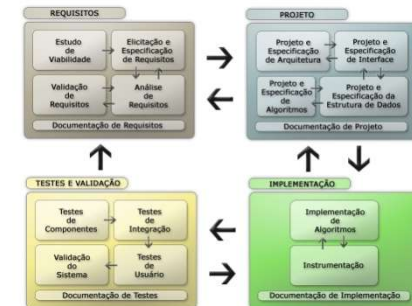
O software (como entidade completa ou incremento parcialmente completo) é entregue ao cliente, que avalia o produto entregue e fornece feedback com base na avaliação.

Essas cinco atividades genéricas de arcabouço podem ser usadas durante o desenvolvimento de pequenos programas, durante a criação de grandes aplicações para a internet e para a engenharia de grandes e complexos sistemas baseados em computador. Os detalhes do processo de software serão diferentes em cada caso, mas as atividades de arcabouço permanecem as mesmas.



# OS MÉTODOS

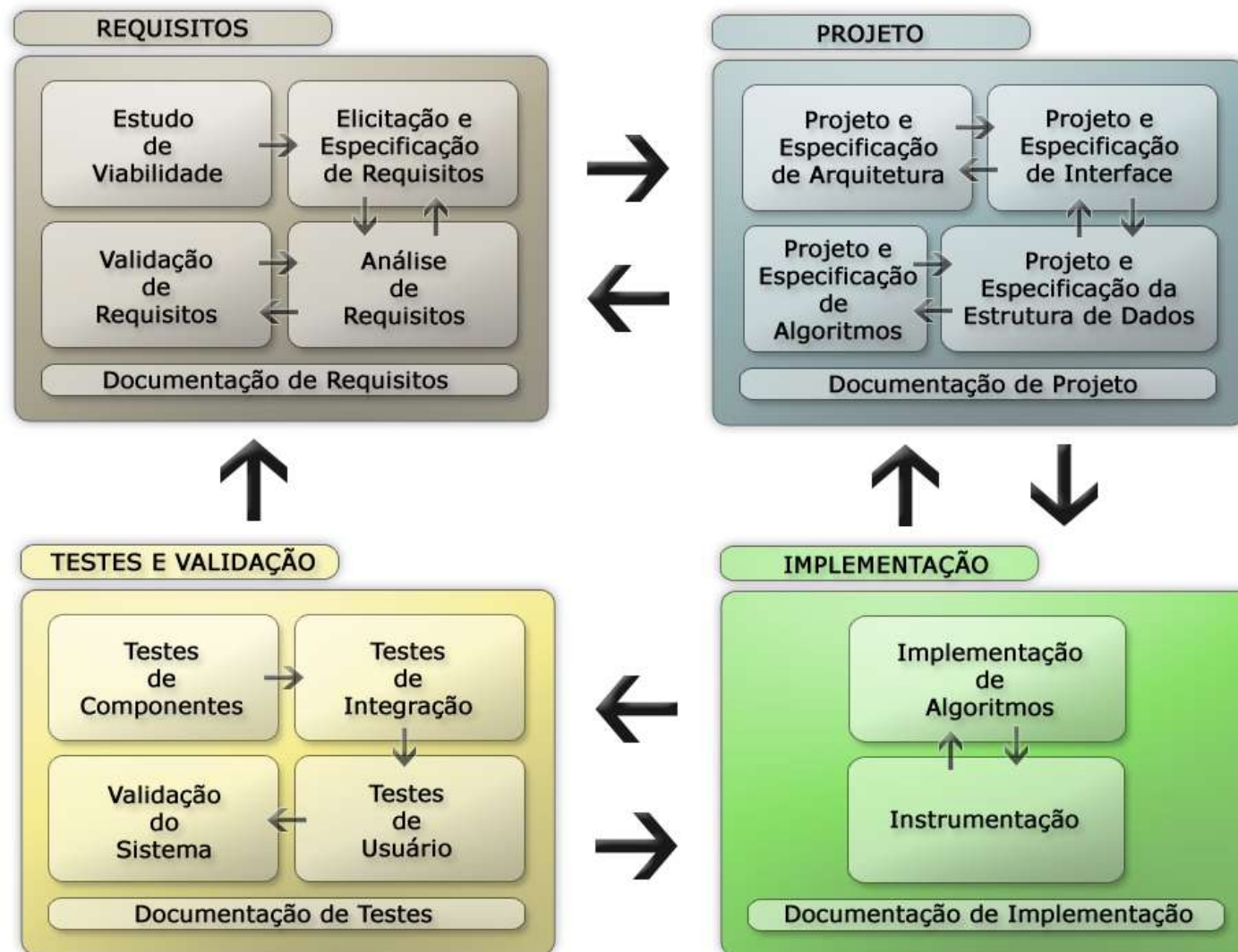
# Os Métodos

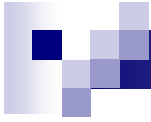


Fornecem a técnica de "Como fazer" para construir softwares. Eles abrangem um amplo conjunto de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes e manutenção.

Os métodos de engenharia de software repousam num conjunto de princípios básicos que regem cada área da tecnologia e incluem atividades de modelagem e outras técnicas descritas.

# Os Métodos





# AS FERRAMENTAS



# As ferramentas



Fornecem apoio automatizado ou semi-automatizado para o processo e para os métodos.

Quando ferramentas são integradas de modo que a informação criada por uma ferramenta possa ser usada por outra, um sistema de apoio ao desenvolvimento de software, chamado **Engenharia de Software Apoiada por Computador**, é estabelecido.

# As ferramentas



Parallels®  
Plesk Panel



**jQuery**  
write less, do more.



JavaScript



**Site Inteligente**  
A ferramenta on-line de gerenciamento de conteúdo perfeita



Magento