



Anhembi
Morumbi

GESTÃO E QUALIDADE DE SOFTWARE

Melhorias de Processos de Software

Profa VANESSA *Cristina* Lourenço

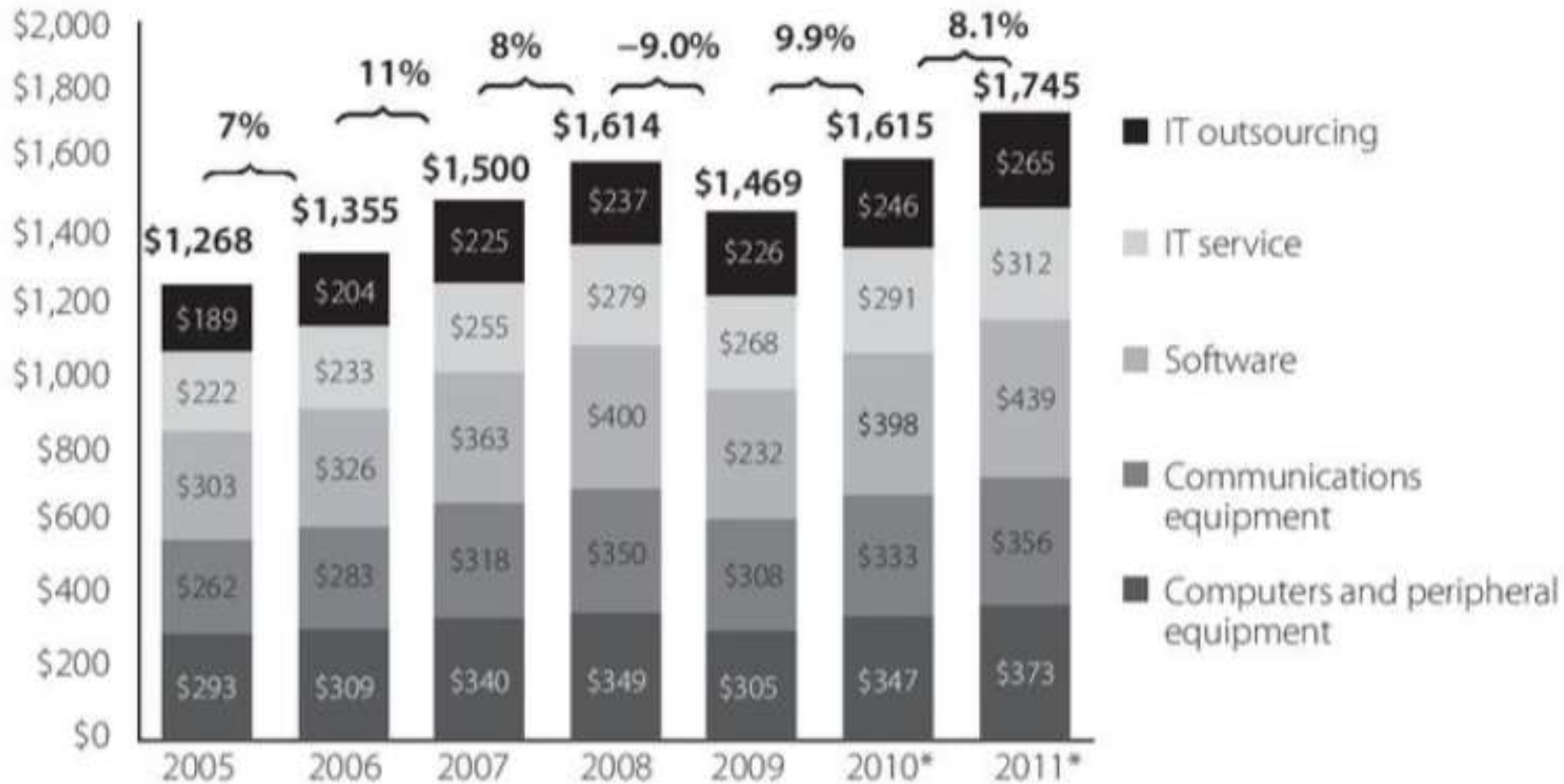
Prof. Nélon Issamu SHIMADA

Prof. Orlando Patriarcha

ecossistema
ânima

AGENDA

- 1 Qualidade de processos;
- 2 Processos de Desenvolvimento de Software.
- 3 Reuso.

Gráfico 1.1. Gastos mundiais com TI.

Fonte: Global IT Market Report (2010 apud SILVA FILHO, 2011, p. 63).

Tabela 1. Previsão mundial de gastos com TI (milhões de dólares americanos)

	Gastos em 2021	2021 Crescimento (%)	Gastos em 2022	2022 Crescimento (%)	Gastos em 2023	2023 Crescimento (%)
Sistemas de Data Center	191.001	6.4	212.218	11.1	221.590	4.4
Programas	735.869	14,7	806.800	9.6	902.182	11,8
Dispositivos	808.580	16,0	767.872	-5,0	790.888	3,0
Serviços de TI	1.207.966	12,8	1.283.192	6.2	1.389.169	8.3
Serviços de comunicações	1.458.527	3.8	1.464.551	0,4	1.505.733	2,8
TI geral	4.401.944	10.2	4.534.632	3,0	4.809.561	6.1

Fonte: Gartner (julho de 2022)

Relatório do Caos do Standish Group

	2011	2012	2013	2014	2015
Projetos bem-sucedidos	29%	27%	31%	28%	29%
Projetos de sucesso parcial	49%	56%	50%	55%	52%
Projetos fracassados	22%	17%	19%	17%	19%

Estatísticas sobre a entrega de projetos de software

The Standish Group

Introdução:

Apenas 31% dos projetos de software terminam bem sucedidos.

Fonte: The Standish Group

The Standish Group

Introdução:

Dentre os projetos de software bem sucedidos, apenas 41% deles dão um bom retorno.

Fonte: The Standish Group

**Você compraria
se soubesse
que:**



Introdução:

Menos de 30% dos carros
fabricados por uma montadora
seriam seguros e confiáveis?

**Você compraria
se soubesse
que:**

Introdução:

De cada 100 imóveis entregues
por uma incorporadora, menos de
30% não cairiam?



**Você
depositaria seu
dinheiro se
soubesse que:**



Introdução:

Num banco que apresentasse 30% de garantia de que o seu dinheiro estaria seguro?

Atualização de software do F-35 de US\$ 14 bilhões é implantada apesar de falhas

17 de janeiro de 2022 | Redação Forças de Defesa



Escritório de testes do Pentágono diz que software é 'imaturo, deficiente'. Falhas nas atualizações do software do caça a jato datam de 2020

Mercedes faz recall de milhares de carros após descobrir bug no sistema de emergência

15 de fevereiro de 2021 2



000
a no

Enquanto o mercado segue atento ao **desenvolvimento de carros autônomos e inteligentes**, algumas montadoras ainda precisam aprimorar seus sistemas. Nesta semana, a Mercedes reconheceu uma falha de software que afeta **cerca de 1,29 milhões de veículos lançados entre 2016 e 2021**.



Motivos que fazem com que os
projetos de software
fracassem...

1. Escopo do projeto mal definido ou incompleto

Um escopo mal feito pode ser consequência de vários problemas:

- O analista não consegue entender com clareza e objetividade o que o usuário precisa,
- O usuário não consegue transmitir ao analista tudo aquilo que deseja e precisa
- O analista que levantou os requisitos com o usuário não consegue transmitir aos desenvolvedores tudo aquilo que lhe foi dito.

Quando o escopo é mal definido, ou é definido de maneira incompleta, certamente o projeto vai fracassar, pois não vai entregar o que o usuário deseja.

1. Escopo do projeto mal definido ou incompleto

O problema de um escopo mal definido fica ainda pior se:

- os usuários não forem envolvidos em nenhum momento durante o desenvolvimento do software,
- se as falhas de comunicação dos requisitos só forem percebidas no final do projeto, durante a entrega.

Depois da definição do escopo, é importante chamar os usuários para que analisem, verifiquem e façam a validação de tudo aquilo que foi identificado como requisitos pelos analistas.



Como o cliente explicou...



Como o líder de projeto entendeu...



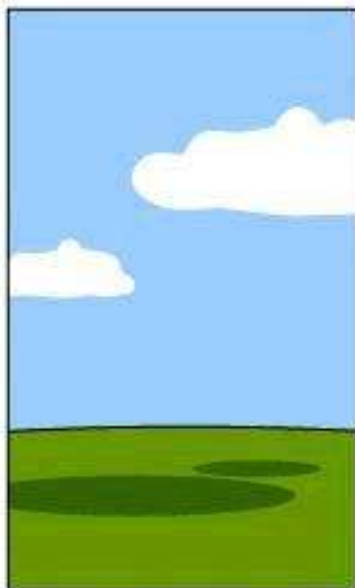
Como o analista projetou...



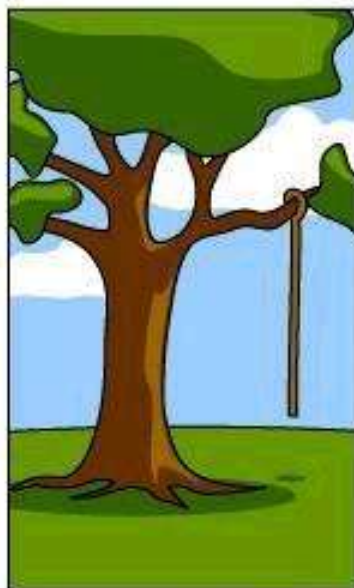
Como o programador construiu...



Como o Consultor de Negócios descreveu...



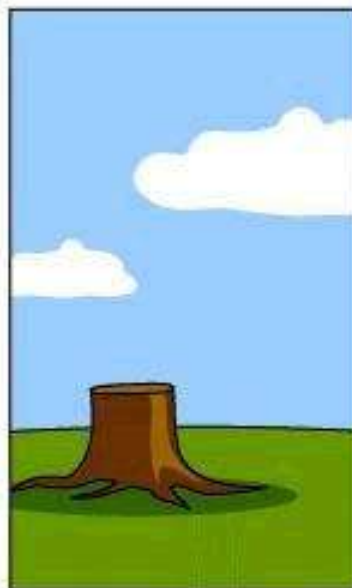
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

2. Identificação de riscos malfeita

- A identificação, ou **análise de riscos**, consiste em **identificar**, de maneira **antecipada**, todos **os problemas** que podem acontecer no decorrer do processo de desenvolvimento de um software.
- Dessa maneira, é possível **indicar** todos os **fatores** que podem acarretar em **comprometimento de prazos ou de orçamento** ao longo do projeto, possibilitando que a equipe se antecipe, **evitando** ou **minimizando os prejuízos**.

3. Pessoas que não sabem trabalhar em equipe

- É preciso que **todos** os integrantes do projeto de desenvolvimento do software saibam **trabalhar em equipe**.
- Quanto mais unidos a equipe estiver, maior esforço haverá em prol de um único objetivo:
 - Entrega do projeto com a qualidade que o cliente espera, dentro do prazo e do orçamento previsto.

4. Liderança ineficiente no projeto

- O **líder** do projeto deve **acompanhar** o andamento de todas as atividades **bem de perto**, para garantir que elas sejam feitas com qualidade e dentro do prazo estipulado.

5. Testes inexistentes ou malfeitos

- A realização de **testes** aumenta significativamente a chance de entregar um software com **sucesso**.
- É preciso que os **testes** sejam feitos de maneira **detalhada e cuidadosa**, verificando se **tudo** o que **foi pedido** pelo **usuário** está sendo entregue:
 - Desempenho,
 - Segurança,
 - Condições de usabilidade.

5. Testes inexistentes ou malfeitos

- Quando os **problemas** são **identificados** durante a realização dos testes, é possível que as **falhas sejam corrigidas** antes que o prazo ou o orçamento seja comprometido.

Regra 10 de Myers.

O custo de encontrar um defeito no sistema aumenta 10 vezes a cada etapa do processo em que esse erro avançar.

Regra 10 de Myers.



5. Testes inexistentes ou malfeitos

- As principais causas de insucesso nos projetos de software são evitáveis se houver:
 - uma boa gestão ao longo do projeto,
 - uma equipe comprometida com o objetivo do projeto.

O sucesso na entrega de um software depende também de:

O processo que é utilizado para desenvolvê-lo.

Processo de Desenvolvimento de Software



Deve ser capaz de unir as competências disponíveis na equipe de projeto ao que o usuário espera do produto final.



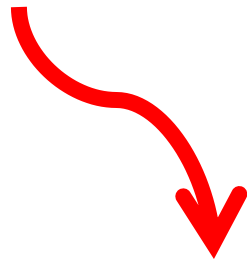
Modelos de Maturidade

BOEING 777



Boeing 777

- Possui cerca de 4,0 milhões de linhas de código.
- 1.300 processadores.



Impossível desenvolvê-lo sem um processo bem definido e maduro.

História

Air France 447: uma das maiores tragédias da aviação

Um Airbus A-330 decolou do Rio rumo a Paris em 2009, mas caiu no Atlântico e deixou 228 mortos. Entenda o desastre.

Por Redação Super Atualizado em 27 fev 2020, 07h25 - Publicado em 19 jul 2018, 15h27



Os escombros só foram localizados em 2011. LatinContent/Getty Images

PUBLICIDADE

SUPER
INTERESSANTE

A CIÊNCIA
DIRETO DA FONTE

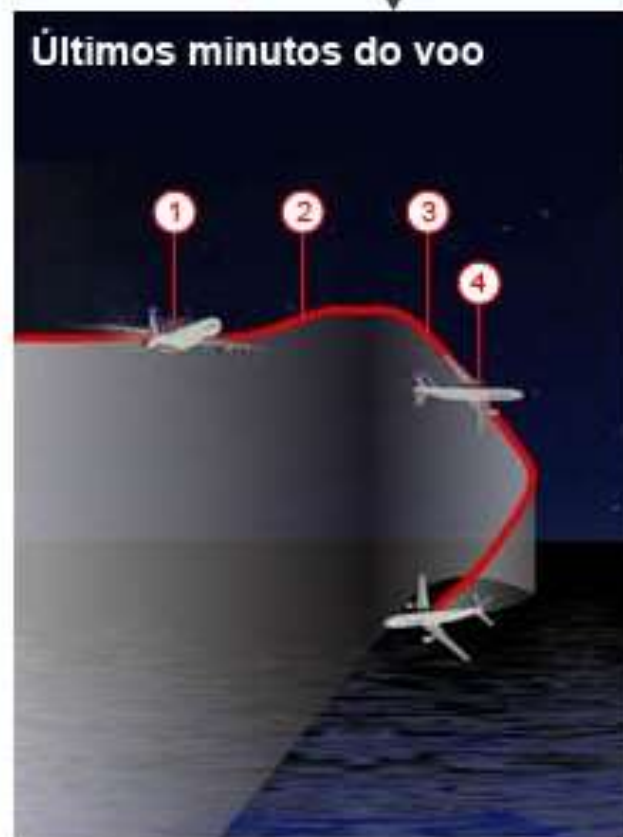
Local do acidente



Desvio de rota do Airbus

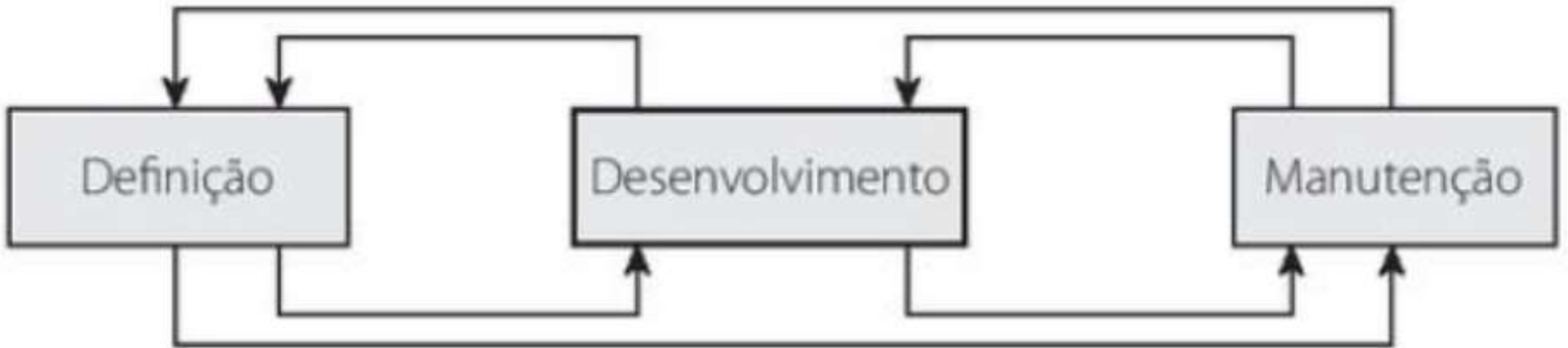


Últimos minutos do voo



Processo de Desenvolvimento de Software

Figura 1.3 Fases genéricas no desenvolvimento de software.



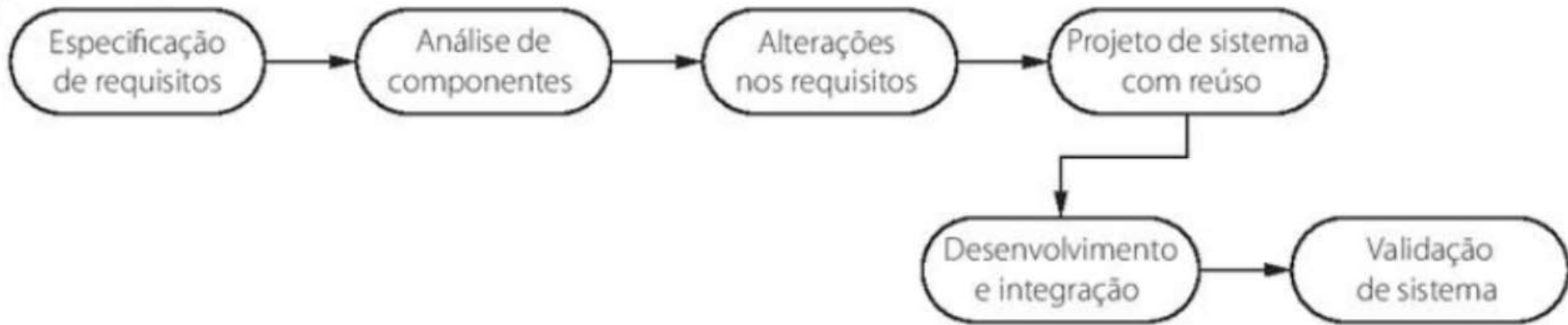
Fonte: Silva Filho (2011, p. 62).

Reúso

- Cultura de Engenharia de Software que se baseia, literalmente, no reuso de elementos entre programas.
- É quase uma reciclagem de códigos.

Reúso

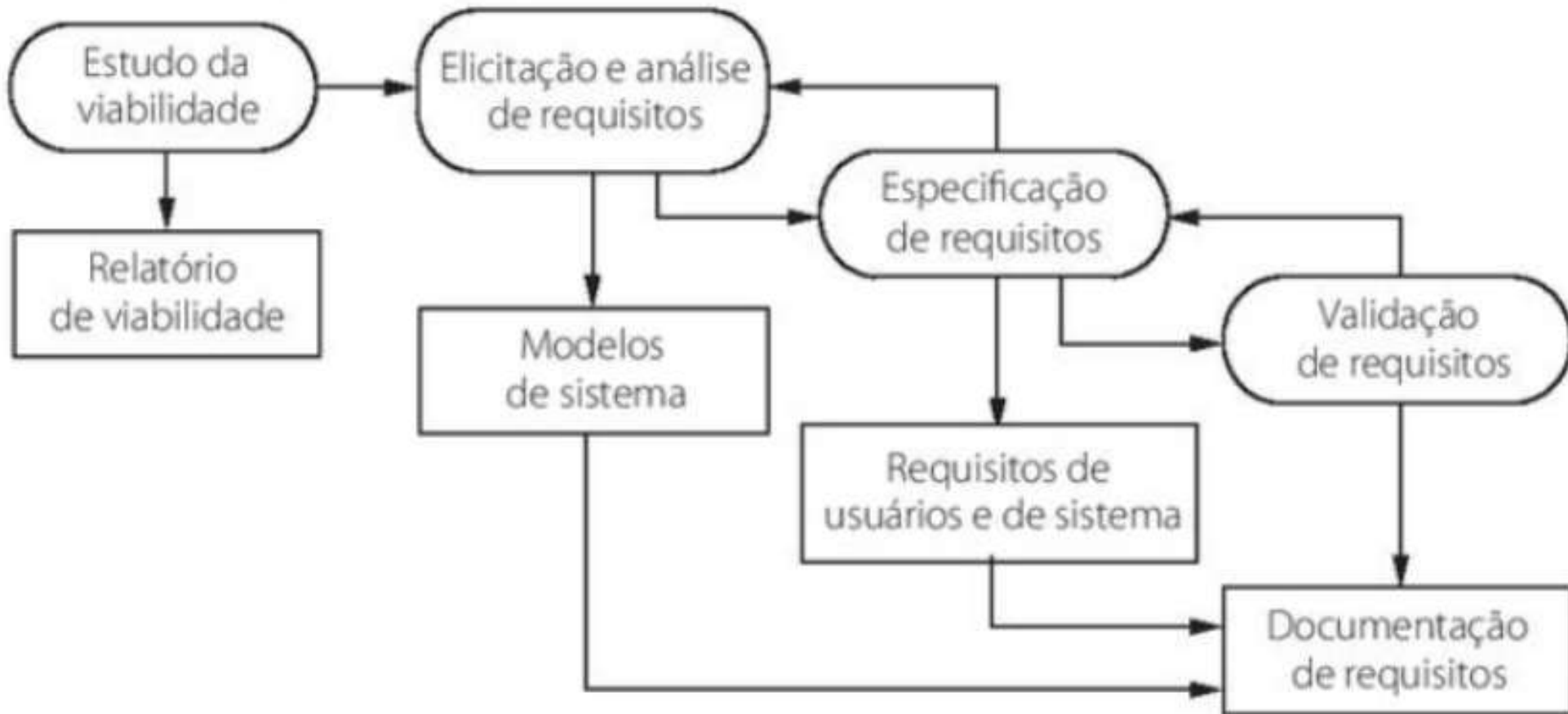
Figura 1.4 Engenharia de software orientada a reúso.



Fonte: Sommerville (2011, p. 23).

Especificação (engenharia de requisitos)

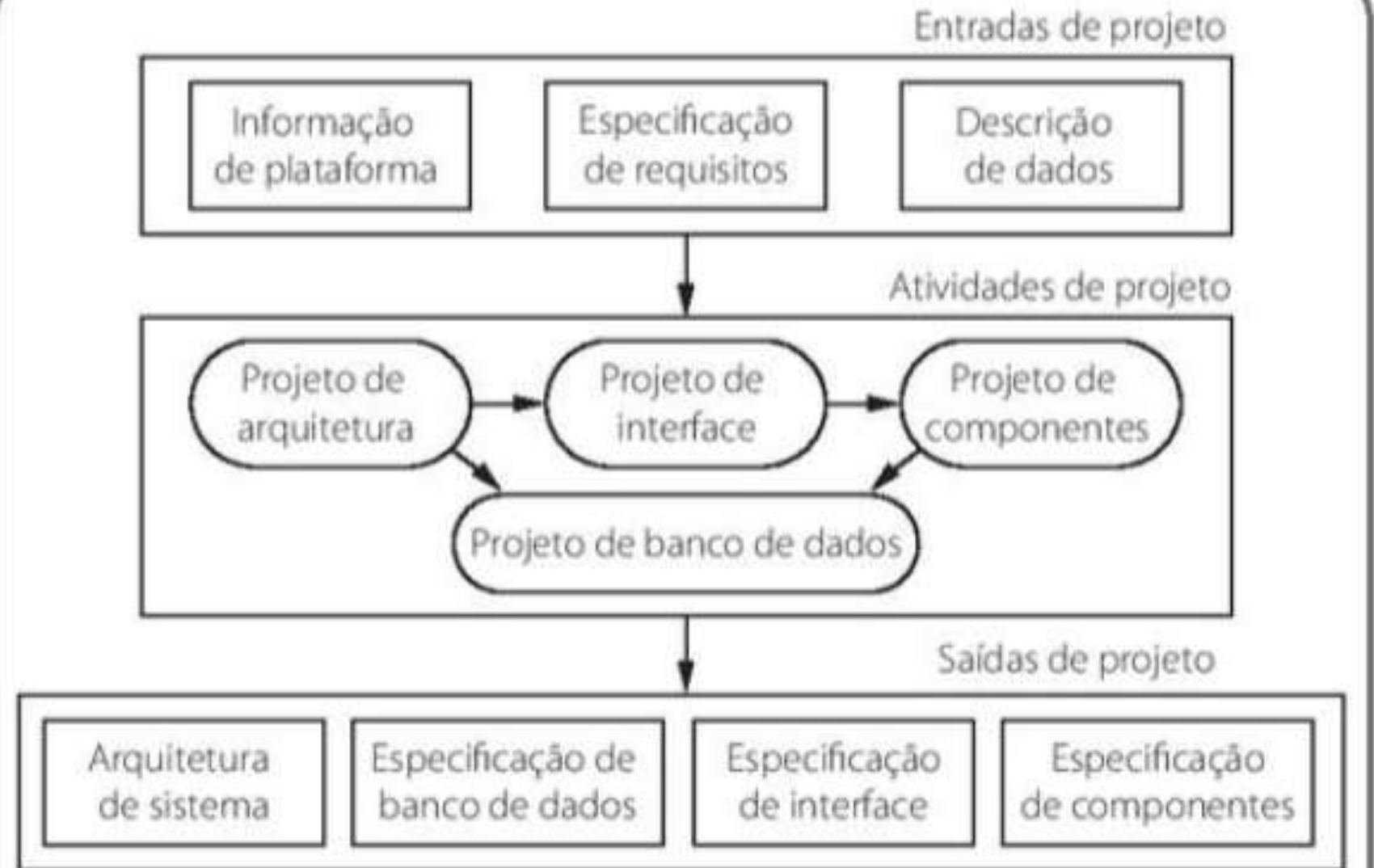
Figura 1.5 A engenharia de requisitos.



Fonte: Sommerville (2011, p. 24).

Implementação

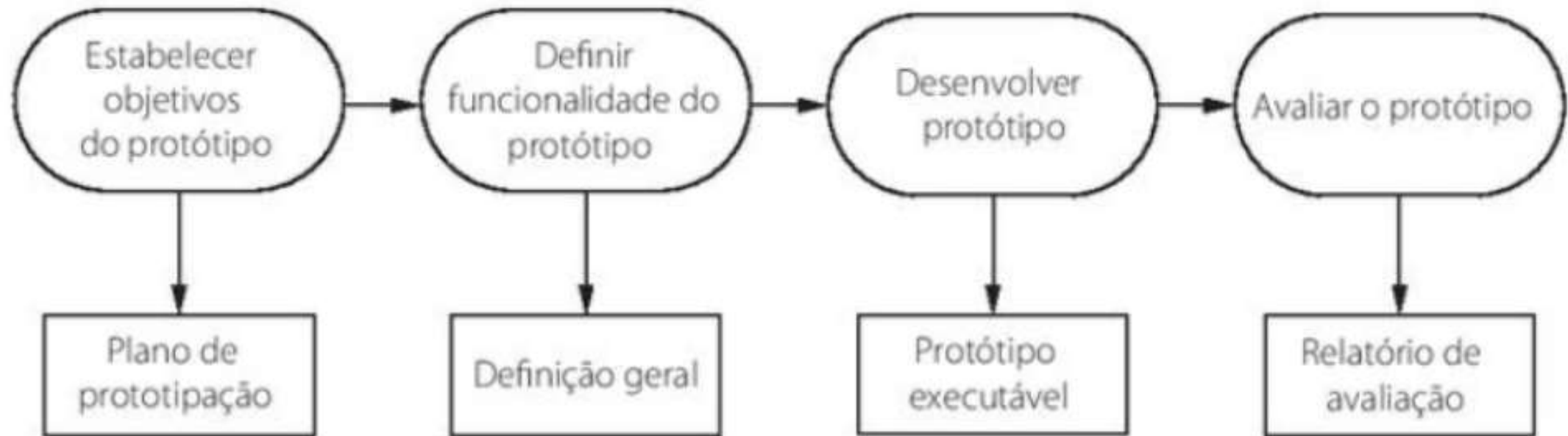
Figura 1.6 Processo do projeto em um modelo genérico.



Fonte: Sommerville (2011, p. 26).

Prototipação

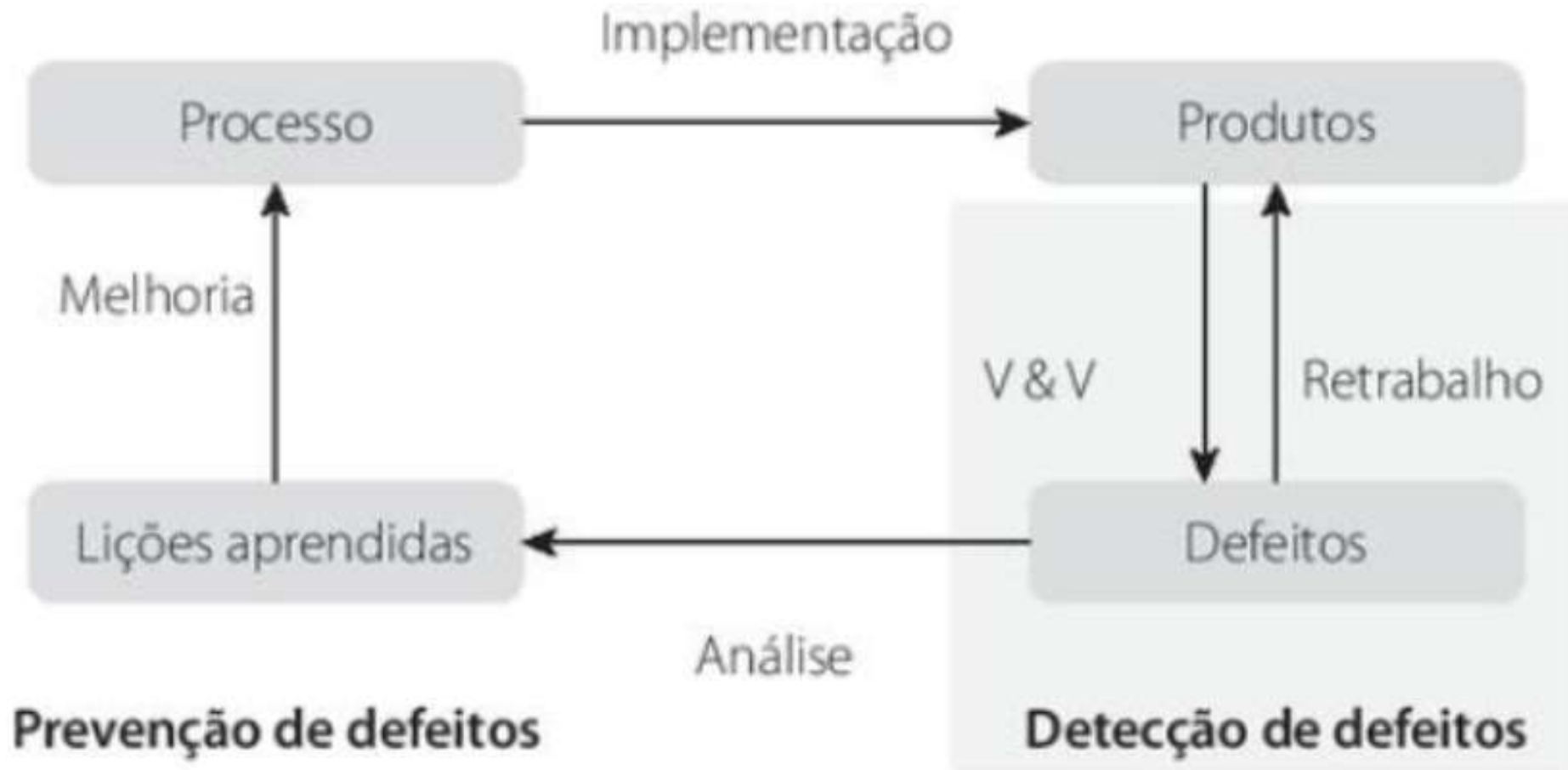
Figura 1.7 Processo genérico de desenvolvimento de protótipos.



Fonte: Sommerville (2011, p. 30).

Prevenção e Detecção

Figura 1.8 Detecção versus prevenção.



Fonte: Vasconcelos et al. (2006, p. 81).

Planejamento de Qualidade

Figura 1.9 Planejamento de qualidade.



Fonte: Vasconcelos et al. (2006, p. 83).

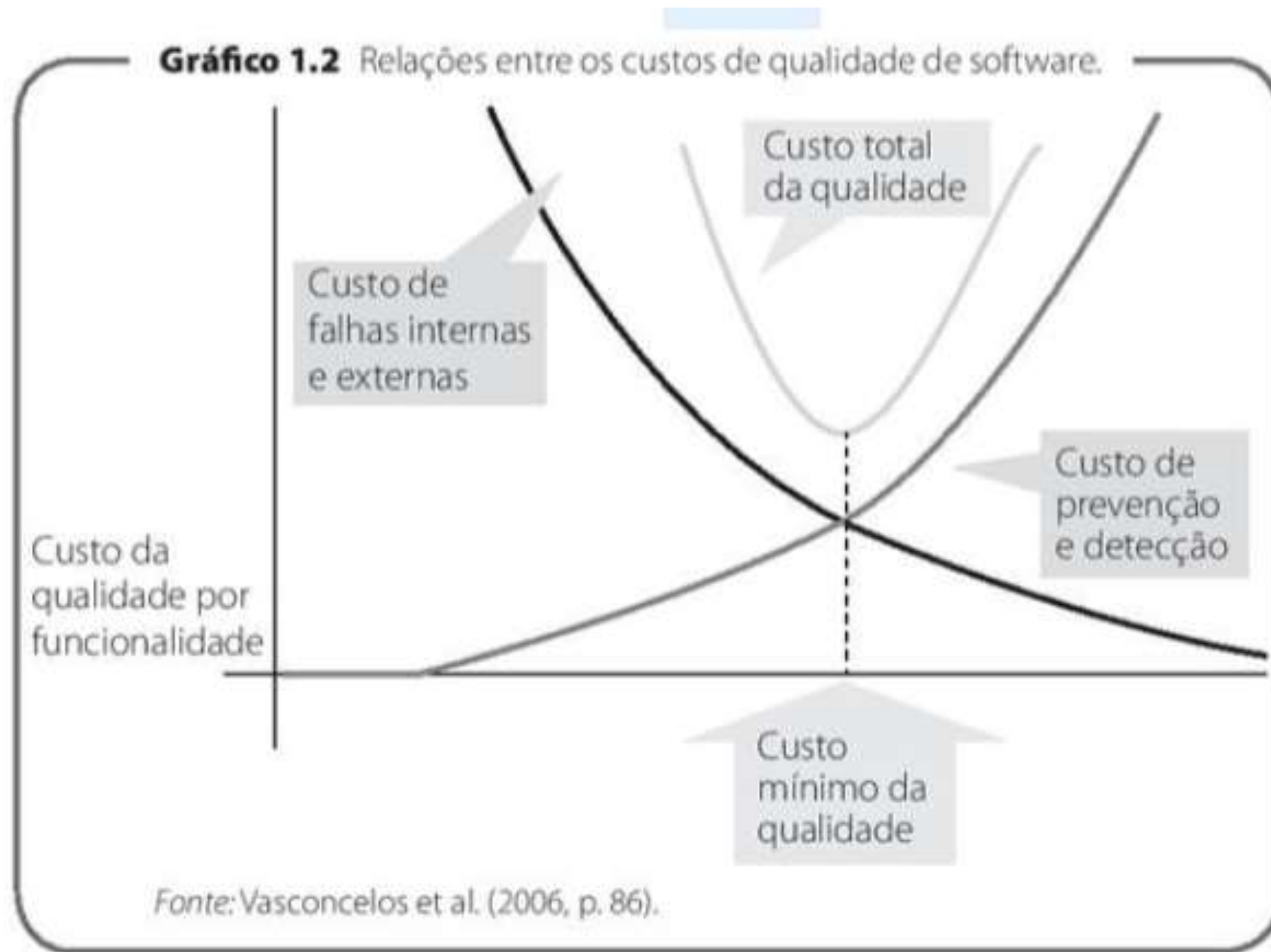
Garantia da Qualidade

Figura 1.10 Visão genérica da garantia de qualidade de software.



Fonte: Vasconcelos et al. (2006, p. 83).

Garantia da Qualidade



Dúvidas



Obrigado !

nelson.shimada@anhembi.br

shimada2@hotmail.com



11 99652 3146