

# Produto de *Software*

# Engenharia de de *software*

- A Engenharia de *Software* surgiu nos anos 70 para transpor os processos de negócio em *software* customizável.
- Os requisitos vinham dos clientes e eram refinados junto à equipe de desenvolvimento.

# Engenharia de *software* ontem e hoje

- O foco no projeto de *software* dominou a indústria por mais de duas décadas e moldou todo conjunto de conhecimentos que compõem a Engenharia de *Software*.
- Quanto mais os processos de negócio se desenvolveram, mais notável se tornou a percepção que muitos negócios não precisavam de *software* customizado.
- Produtos genéricos de *software* podem muito bem atender a diversos problemas de negócio simples.

# Produto de software

## Definição

*"Produtos de software são sistemas computacionais genéricos vendidos a governos, empresas e consumidores." (Sommerville, 2020, trad. nossa)*

# Produto de *software*

Desenhados para:

- suporte a processos de negócio;
- ferramentas de produtividade;
- jogos;
- sistemas de informação pessoal.

# Projeto de *software* × Produto de *software*

- Um das diferenças maiores entre projeto e produto de *software* se dá nos requisitos.
- No projeto de *software*, o cliente define os requisitos e paga por eles.
- Se o processo do cliente muda, então mudam os requisitos.
- No produto de *software*, quem define os requisitos é o desenvolvedor (empresa) baseado na observação dos problemas de potenciais clientes.

# Projeto de *software* × Produto de *software*

- O custo do projeto de *software* tende a ser mais baixo para o cliente, pois o custo de produção é dividido entre mais clientes.
- Sommerville afirma que o controle do produto de *software* pelo desenvolvedor gera mais riscos para o cliente.
- Isso é discutível. Dependerá do contrato.

# Projeto de *software* × Produto de *software*

- Sommerville também coloca que produtos excelentes comumente falham porque outros produtos inferiores chegam ao mercado primeiro.
- Isso é também discutível. Depende de diversos fatores, como a expansão do mercado.
- Por exemplo, a Amazon é a pioneira em *Cloud Computing*, porém Google, Microsoft, Alibaba, IBM e outras conseguem algum nível de competição.
- Outro exemplo foram as pioneiras Microsoft e Symbian nos *smartphones*, mas foram logo ultrapassadas por Blackberry, Apple e demais.



# Visão do produto

- Segundo Sommerville, o ponto de partida para o desenvolvimento de um produto deveria ser a **visão do produto**.
- Esta é a descrição sucinta e simples do produto a ser desenvolvido e o seu diferencial perante os concorrentes.
- Todas funcionalidades e características do *software* devem estar de acordo com a visão do produto.

# Visão do produto

Sommerville define três questionamentos que amparam a criação da visão do produto, a saber:

- **O que** é o produto a ser desenvolvido? **O que** faz desse produto diferente dos seus competidores?
- **Quem** é o público-alvo deste produto?
- **Por que** clientes compram esse produto?

# Visão do produto

No livro *Crossing the Chasm*, Geoffrey Moore sugere a seguinte estruturação para escrever a visão do produto baseada em palavras-chave:

- PARA (cliente alvo);
- QUE (necessidade ou oportunidade);
- O (nome do produto) É UM (categoria do produto);
- QUE (benefício-chave ou razão para comprar);
- DIFERENTE DE (competidor);
- NOSSO PRODUTO (diferenciação).

# Visão do produto

**PARA** estudantes surdos e professores, **QUE** necessitam de um portal educativo com *interface* adaptada para o público surdo, **O** Portal Mão Amiga **É UM** *software* gerenciador de conteúdo educacional **QUE** possui *interface* completamente adaptada às necessidades dos alunos surdos. **DIFERENTE DO** Moodle e similares, **NOSSO PRODUTO** permite que o surdo acesse aos conteúdos sem auxílio de intérprete ou treinamento.

# Prototipação

- Os produtos de *software* surgem geralmente da identificação de oportunidades.
- É sugerida criação de protótipo como resultado de uma primeira versão.
- Um protótipo é uma ferramenta muito boa tanto para provar que a proposta de do produto é factível quanto para atrair investidores.

# Prototipação

- Para *softwares* de uso interno, como os utilizados por grupos de pesquisa, um protótipo pode ser o bastante.
- Quando este *software* for utilizado externamente, recomenda-se "jogá-lo fora" e refazê-lo seguindo boas práticas e parâmetros de confiança e segurança.

# Referências

- Sommerville, Ian. Software Engineering - Global Edition. 10ed. 2016. Pearson Education.
- Sommerville, Ian. Engineering Software Products: An Introduction to Modern Software Engineering. 1ed. 2021. Pearson Education.