

# Requisitos Arquiteturais

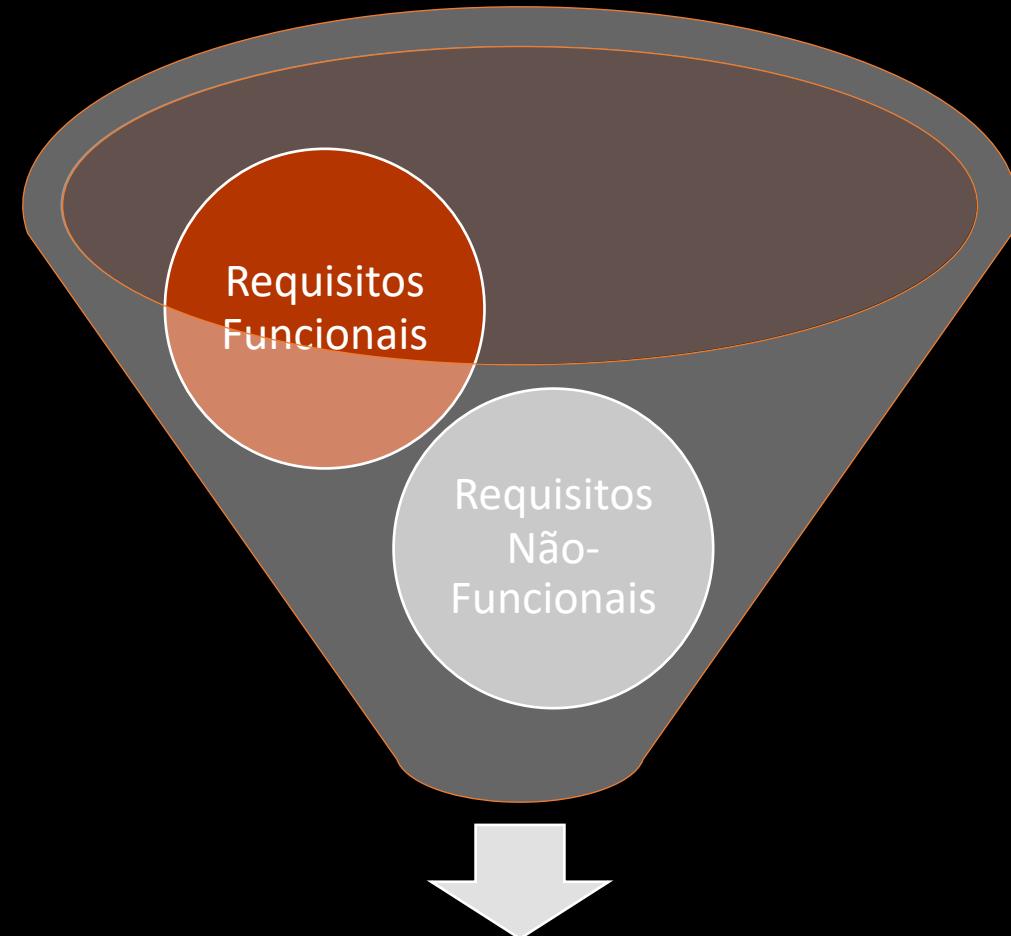
Marco Mendes

## Requisitos influenciam a arquitetura

---

- Os requisitos influenciam a arquitetura da solução, porque são selecionados elementos de solução que satisfazem os requisitos declarados.
- Os arquitetos devem se concentrar em **requisitos arquiteturalmente significativos** ao derivar a arquitetura de um produto.





Requisitos Arquiteturais

# Como determinar quais requisitos são arquiteturais?



# Requisitos Arquiteturais

São requisitos críticos para o negócio, complexos no contexto do projeto e que portanto afetam o desenho da arquitetura de um produto.

Esse requisito  
é arquitetural?

O sistema deve **operar em múltiplos navegadores e telefones celulares** dos alunos e professores.



# Requisitos bem definidos resultam em arquiteturas de maior qualidade

---

- Como os requisitos orientam a definição da arquitetura, conclui-se que um conjunto bem definido de requisitos resultará em uma solução de arquitetura melhor do que um conjunto de requisitos mal definidos.
- Ter arquitetos participando de captura de requisitos pode, dada a sua experiência, muitas vezes resultar em requisitos de melhor qualidade.



# Exemplo de informação capturada em uma oficina

- *O desempenho e a escalabilidade são preocupações, devido aos períodos de pico de atividades como lançamento de notas, matrícula ou consultas de notas no fim do semestre.*
- *Esperamos picos de 2 milhões de acessos diários durante a primeira semana de Dezembro desse ano.*

# As decisões de arquitetura influenciam os requisitos

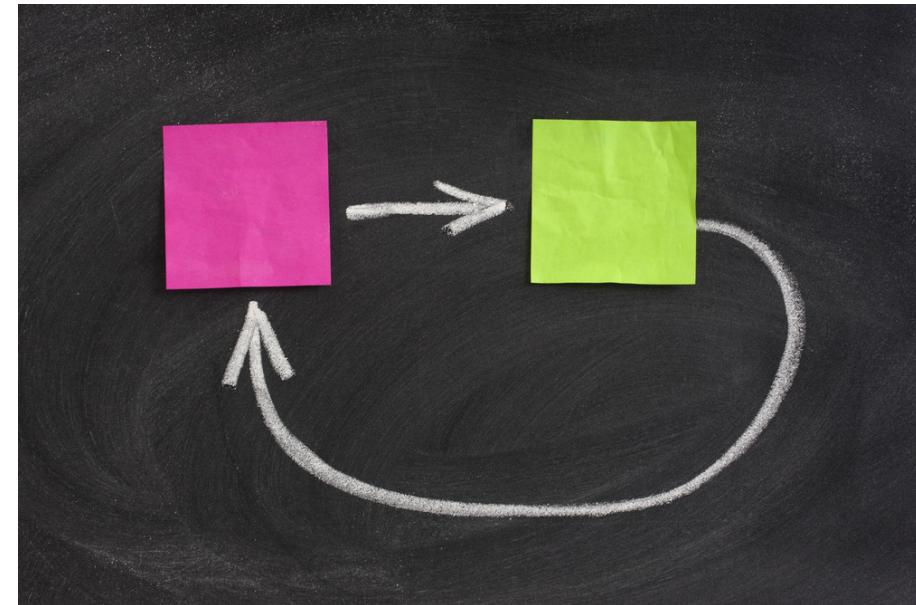
- ❖ Ao definir a arquitetura, o arquiteto deve fazer com frequência trocas entre requisitos, como equilibrar desempenho e custo.
- ❖ Como resultado, o feedback vai do arquiteto ao Analista de Negócios e Donos de Produtos, de modo que os requisitos possam ser ajustados conforme necessário.



# As decisões de arquitetura influenciam os requisitos

---

- A seleção de um elemento arquitetural específico também pode restringir os requisitos que podem ser atendidos.
- A seleção de um produto de prateleira (COTS/MOTS) pode impor certas restrições aos requisitos em termos da terminologia usada, da funcionalidade fornecida e de quaisquer qualidades alcançadas.



# Resumo

- Requisitos arquiteturais são significativos para o desenho da arquitetura.
- Arquitetos são responsáveis por coletar esses requisitos juntos aos analistas, donos de produtos e outros interessados.
- Nem todo requisito não-funcional é arquitetural.
- Alguns requisitos funcionais podem ser arquiteturais.



# Relação entre Requisitos Arquiteturais e Não- Funcionais

Marco Mendes

# Requisitos não-funcionais

Requisitos não funcionais incluem restrições e atributos de qualidade.

Uma restrição é uma restrição ao grau de liberdade que temos em fornecer uma solução.

Os atributos de qualidades são propriedades ou características do sistema com as quais seus interessados se preocupam e, portanto, afetam seu grau de satisfação com o sistema.

# Requisitos não- funcionais Exemplos

Requisitos não funcionais incluem restrições e atributos de qualidade.

Restrição: O sistema deve comunicar com o sistema legado CICS/COBOL/Oracle

Atributos de Qualidade: O tempo médio de consulta de notas nos telefones celulares deve ser menor que ou igual a 4 segundos.

## Atenção!

Nem todo requisito não-funcional  
é um requisito arquitetural.

---

Alguns requisitos não funcionais  
podem ser complexos, mas se não  
apresentarem valor de negócio alto  
não serão requisitos arquiteturais.



# Resumo

- Requisitos arquiteturais são contextuais.
- Nem todo requisito não-funcional é um requisito arquitetural.
- Nem todo requisito arquitetural é um requisito não-funcional.



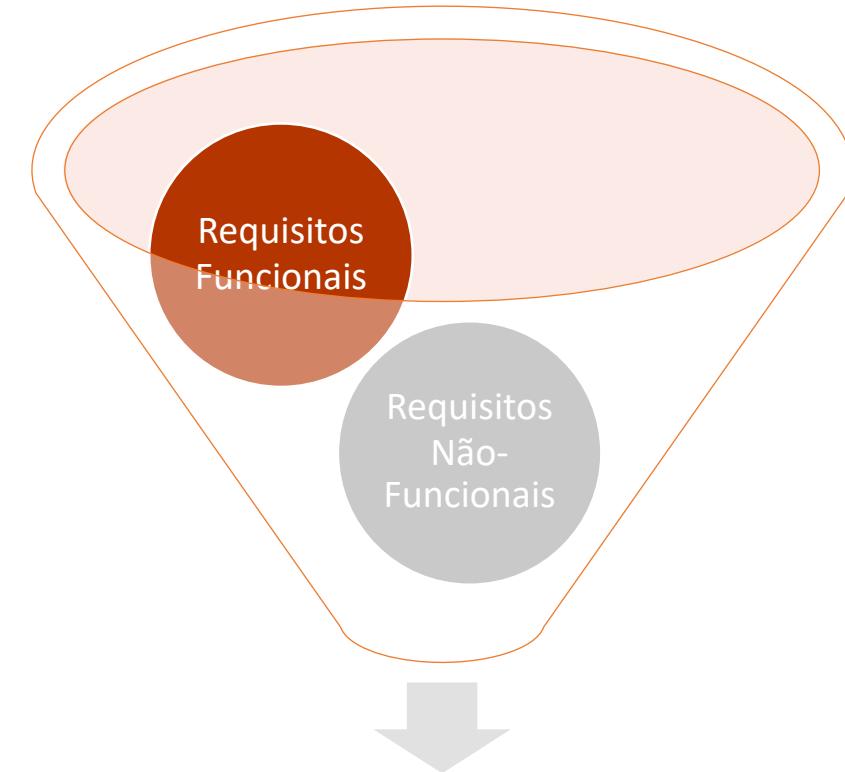
# O Modelo FURPS+ para Identificar Requisitos Arquiteturais

Marco Mendes

# Requisitos de significância arquitetural

---

A expressão arquiteturalmente significativo costuma ser usada para caracterizar o subconjunto de elementos relevantes para a arquitetura.





# Requisitos de significância arquitetural

---

*Um técnica útil é identificar  
condutores arquiteturais e seus  
racionais de escolha.*



# Exemplos: Sistema Acadêmico

O elemento está associado a algumas funcionalidades críticas do sistema, sem as quais você não possui um sistema viável.

O elemento está associado a alguma qualidade crítica do sistema, como desempenho, sem o qual você não tem um sistema viável.

O elemento está associado a uma restrição crítica na solução, como a necessidade de integração com um sistema externo específico.

---

## Integração do sistema acadêmico com o censo acadêmico do MEC

---

## Suportar 200.000 alunos consultando notas na ultima semana do semestre

---

## Integração com sistema ERP da TOTVS

---

# Modelo FURPS

- O modelo FURPS+ é um aliado para realizarmos uma varredura completa sobre potenciais requisitos arquiteturais.
- *Functionality - Funcionalidade*
- *Usability - Usabilidade*
- *Reliability – Confiabilidade*
- *Performance - Desempenho*
- *Supportability – Suportabilidade*
- *+ Restrições, Interfaces e Requisitos Físicos*

# Funcionalidade

- A categoria Funcionalidade da classificação FURPS + reúne todos os requisitos funcionais.
- Esses requisitos geralmente representam as características principais do produto. Em um aplicativo de armazém, podemos ter requisitos relativos a "Processamento de Pedido" ou "Controle de estoque".



# Funcionalidade

- Exemplos comuns:
  - ❖ Auditoria
  - ❖ Licenciamento
  - ❖ Regionalização
  - ❖ Ajuda
  - ❖ Envio de e-mails
  - ❖ Impressão
  - ❖ Relatórios
  - ❖ Segurança
  - ❖ Fluxos de trabalho
  - ❖ Pesquisa de informações



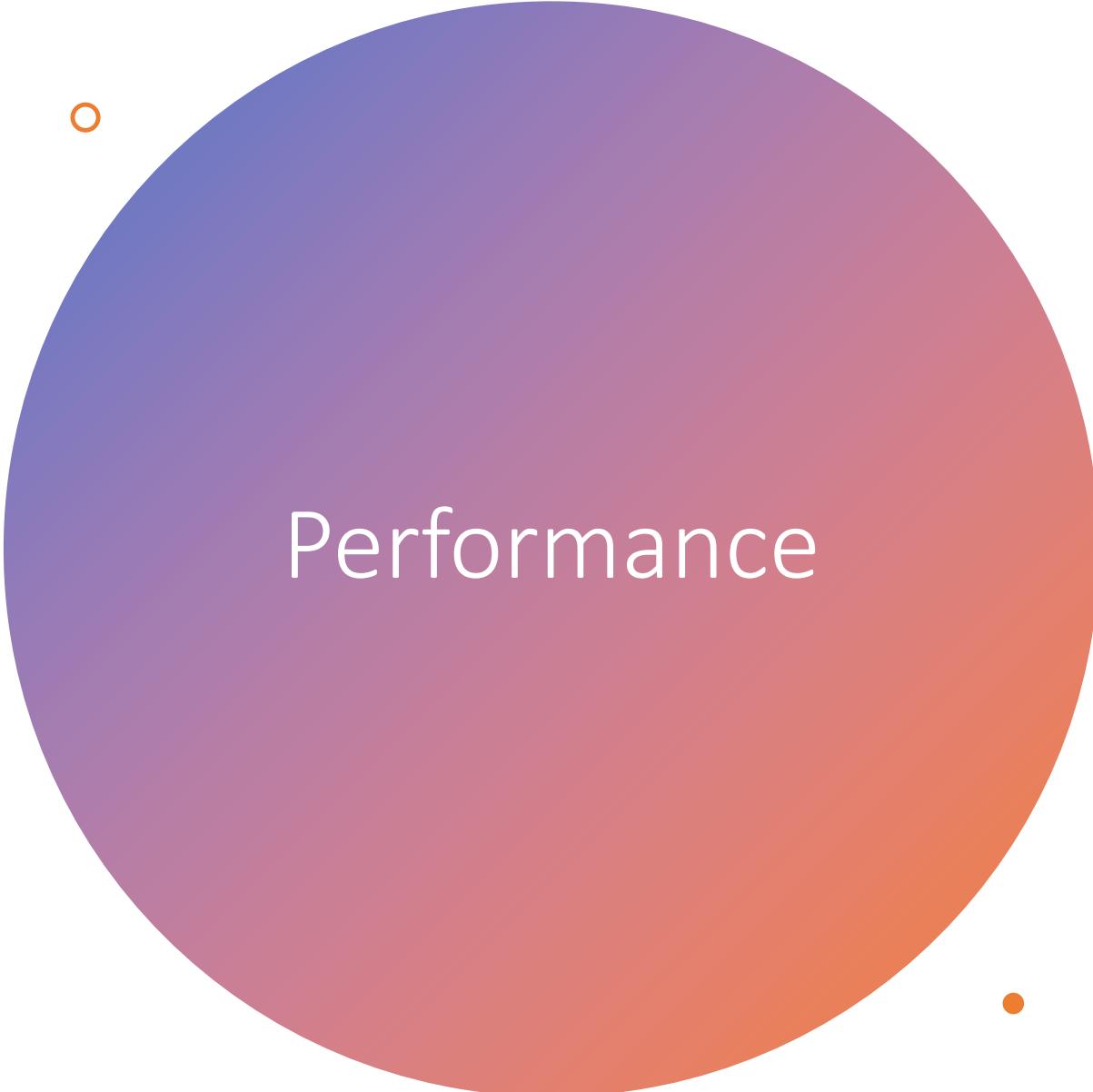
Usabilidade

- Exemplos comuns:
  - Consistência de uso
  - Qualidade estética
  - Acessibilidade
  - Esforço no uso



Confiabilidade

- Exemplos comuns:
  - Precisão nos cálculos realizados.
  - Disponibilidade
  - Recuperabilidade



Performance

- Exemplos comuns:
  - Tempo de resposta
  - Tempo de processamento
  - Vazão



Suportabilidade

- Exemplos comuns:

- Manutenibilidade
- Testabilidade
- Escalabilidade
- Instalabilidade
- Compatibilidade
- Regionalibilidade
- Configurabilidade



+  
Restrições ao  
Desenho

- Exemplos comuns:

- Interoperar com um banco de dados já existente.
- Operar dentro de uma infraestrutura física já existente.



+

Interfaces de  
comunicação

- Exemplos comuns:
  - Interoperabilidade com sistemas externos
  - Formatos de interface



## Aspectos Físicos

- Exemplos comuns:
  - Tamanhos mínimos e máximos de tela
  - Formatos de teclado
  - Peso e forma

# Resumo

- Requisitos arquiteturais são significativos para o desenho da arquitetura.
- O modelo FURPS+ fornece um gabarito de requisitos comuns para auxiliar o arquiteto a identificar potenciais riscos técnicos para o projeto e que derivam requisitos arquiteturais.
- Para saber mais:
  - *Capturing Architectural Requirements*, Peter Eeles
  - <https://www.ibm.com/developerworks/rational/library/content/03July/getstart/RP/capturing.pdf>



# Exemplo em um Sistema Acadêmico

Marco Mendes



## Condutor Acessibilidade e Usabilidade

- A acessibilidade é o atributo de qualidade que possibilita que um sistema seja universalizado e acessado pelo máximo de pessoas no maior número possível de dispositivos.
- Alunos entre 17 e 30 anos já se acostumaram a usar sistemas do estado da arte de classe mundial em seus celulares e com isso a usabilidade também se torna uma preocupação técnica para o nosso projeto

# Condutor Interoperabilidade



- Exemplo: Durante vários o sistema acadêmico na nova tecnologia e o sistema legado irão conviver e, portanto, a interoperabilidade é um aspecto crítico.
- Além disso, o time de arquitetura também precisa interoperar com outros sistemas, como por exemplo:
  - O sistema do Ministério da Educação que realiza o censo acadêmico;
  - O sistema de segurança corporativa da faculdade baseado em Microsoft Active Directory;
  - O sistema ERP que cuida do processamento da folha de pagamento e geração de boletos para os alunos.

# Condutor

# Segurança

---

- Sistemas acadêmicos são desafiados por *hackers* em forma contínua e manipulam dados muito sensíveis.
- Existe um conjunto de dados sigilosos como informações de milhares de alunos que irão circular em ambiente Web e dados como notas, faltas e comunicação entre professores.
- A agenda de autenticação, autorização, auditoria e transporte seguro ganha importância e deve ser trabalhada pelo time de arquitetura.





# Condutor Escalabilidade

---

- Faculdades possuem um calendário acadêmico rígido e os sistemas devem respeitar essa questão. Por exemplo, os períodos de lançamento de notas, consultas de notas e matrículas em disciplinas são críticos e geram sobrecargas imensas aos servidores.
- Dessa forma, a escalabilidade se torna um aspecto crítico e traz uma agenda de preocupação técnica para o time de arquitetura.
- O nosso sistema acadêmico exemplo tem 40.000 alunos usuários e esperamos pico de 2.000.000 de acessos na primeira semana de Dezembro.

# Condutor – Manutenibilidade

- O diretor da área de TI quer minimizar os custos de operação do produto.
- Além disso, sistemas acadêmicos tem regras de negócio bastante complicadas que requerem uma gestão bastante cuidadosa.

```
public static DateTime GetPostDate(string postDate)
{
    DateTime sDt = new DateTime();
    DateTime emptyDate = new DateTime();
    emptyDate = DateTime.MinValue;
    CompareInfo myComp = CultureInfo.InvariantCulture.CompareInfo;
    if (myComp.IndexOf(postDate, " ") > -1)
    {
        System.IFormatProvider form = new System.Globalization.CultureInfo("pt-BR");
        try
        {
            sDt = DateTime.Parse(postDate, form, DateTimeStyles.None);
        }
        catch { sDt = emptyDate; }
    }
    else
    {
        int YYYY, mm, dd;
        if (postDate.Trim0 == "") sDt = emptyDate;
        else
            sDt = DateTime.ParseExact(postDate, "yyyy-MM-dd", form, DateTimeStyles.None);
    }
}
```

# Resumo

- Requisitos arquiteturais são significativos para o desenho da arquitetura.
- Busque condutores ligados a aspectos funcionais tais como acessibilidade, segurança ou manutenibilidade



# Requisitos SMART

Marco Mendes

# A técnica dos requisitos

# SMART

S

*Specific (Específicos).* Um requisito deve ser específico. Exemplos de requisitos específicos indicam o caso de uso, tela ou módulo a que eles se referem.

Exemplo no sistema acadêmico: O tempo médio de consulta de notas nos telefones celulares deve ser menor que ou igual a 4 segundos.

M

*Mensurable (Mensurável).* Requisitos arquiteturais devem ser mensuráveis, sempre que possível. Isso permite que ele seja testado.

Exemplo no sistema acadêmico: O tempo médio de consulta de notas nos telefones celulares deve ser menor que ou igual a 4 segundos.

A

Esta característica verifica se um requisito é atingível, independente dos recursos e tempo disponível. Pode parecer que todo requisito é atingível mas existem contraexemplos.

Se escrevéssemos que a tela de cadastro de alunos precisa ter disponibilidade de 100%, estariamos prometendo algo que não pode ser cumprido. Todo programa computacional opera sobre máquinas e redes, que apresentam tempos médios entre falhas (MTBF) em suas especificações

R

Vários requisitos são atingíveis, mas nem todos o são no contexto de um projeto. Projetos têm tempo limitado e recursos físicos e financeiros limitados.

Se, por exemplo, buscássemos que um sistema Web operasse em um ambiente tolerante a falhas, mas não tivéssemos orçamento para buscar máquinas de redundância, esse requisito não seria realizable.

T

Determina a origem e validade sobre o requisito.

Alguns estudos mostram quase 30% dos requisitos de softwares em produção não são usados. Um dos motivos para isso é a introdução de requisitos nos softwares as vezes ocorre de forma indisciplinada e as vezes pelos próprios desenvolvedores.

# S

*Specific* (Específicos). Um requisito deve ser específico. Exemplos de requisitos específicos indicam o caso de uso, tela ou módulo a que eles se referem.

Exemplo no sistema acadêmico: O tempo médio de consulta de notas nos telefones celulares deve ser menor que ou igual a 4 segundos.

# M

*Mensurable* (Mensurável). Requisitos arquiteturais devem ser mensuráveis, sempre que possível. Isso permite que ele seja testado.

Exemplo no sistema acadêmico: O tempo médio de consulta de notas nos telefones celulares deve ser menor que ou igual a 4 segundos.

# A

Esta característica verifica se um requisito é atingível, independente dos recursos e tempo disponível. Pode parecer que todo requisito é atingível mas existem contraexemplos.

Se escrevêssemos que a tela de cadastro de alunos precisa ter disponibilidade de 100%, estariamos prometendo algo que não pode ser cumprido. Todo programa computacional opera sobre máquinas e redes, que apresentam tempos médios entre falhas (MTBF) em suas especificações

# R

Vários requisitos são atingíveis, mas nem todos o são no contexto de um projeto. Projetos tem tempo limitado e recursos físicos e financeiros limitados.

Se, por exemplo, buscássemos que um sistema Web operasse em um ambiente tolerante a falhas, mas não tivéssemos orçamento para buscar máquinas de redundância, esse requisito não seria realizável.

# T

Determina a origem e validade sobre o requisito.

Alguns estudos mostram quase 30% dos requisitos de softwares em produção não são usados. Um dos motivos para isso é a introdução de requisitos nos softwares as vezes ocorre de forma indisciplinada e as vezes pelos próprios desenvolvedores.

## Exemplo de Requisito Ambíguo

---

*“O sistema acadêmico deve ser rápido e capaz de processar grandes quantidades de requisições simultâneas.”*

# Exemplo de Requisitos SMART

---

*A tela de matrícula em disciplina para alunos deve possuir um tempo de resposta menor que 8 segundos e suportar 2000 usuários simultâneos em horários de pico (8:00 às 19:00 na semana de matrícula).*

# Resumo

- Requisitos arquiteturais bem escritos reduzem a ambiguidade e facilitam o projeto de arquitetura.
- A técnica SMART é simples e pode ser aplicada para cada requisito arquitetural.



# Pegadinhas na Coleta de Requisitos Arquiteturais

Marco Mendes

## Pegadinha #1

### *Tratar Desejos como Requisitos*

- Uma desejo não é um requisito; é uma expressão do que uma parte interessada gostaria de ter no sistema.
- Ele não se torna um requisito até que um acordo explícito seja alcançado com as partes relevantes e quaisquer conflitos com os requisitos existentes tenham sido resolvidos.



# Pegadinha #1

## Tratar Desejos como Requisitos

- Quando você aplica práticas ágeis e um representante do cliente é considerado parte da equipe, uma solicitação pode ser um requisito e qualquer negociação posterior pode ser contornada sem maiores problemas.
- Mas compreender a diferença entre solicitações e requisitos ainda é importante mesmo nessas circunstâncias pois um indivíduo pode declarar solicitações conflitantes ou ambíguas que precisam de reconciliação com os requisitos existentes antes que possam ser acomodados.



# Exemplo: Desejo versus Requisito

- O sistema deve **operar em qualquer modelo de telefone celular para qualquer de nossos alunos...** Ou
- O sistema deve ter disponibilidade de 100%.

# Pegadinha #2

## *A mentalidade do carrinho de compras*

---

- Ao coletar solicitações das partes interessadas, é importante evitar a mentalidade de carrinho de compras das partes interessadas. Na ausência de qualquer outro critério, um exercício de coleta de pedidos rapidamente se degrada em uma conversa infrutífera na qual o usuário decide que tudo que é oferecido é absolutamente necessário.
- No entanto, acomodar essas solicitações não é gratuito e isso não é um carrinho de compras. É muito fácil cair na armadilha de apresentar aos seus interessados o equivalente a uma lista de compras da qual eles escolhem.



## Pegadinha #2 – *A mentalidade do carrinho de compras*

- O truque é garantir que as partes interessadas compreendam o custo de suas compras, sendo capaz de posicionar o efeito de selecionar um requisito específico em termos amigáveis para as partes interessadas, como custo, cronograma e qualidade.



## Exemplo

### Pergunta ingênua

- *O sistema acadêmico deve ser seguro?*
- 

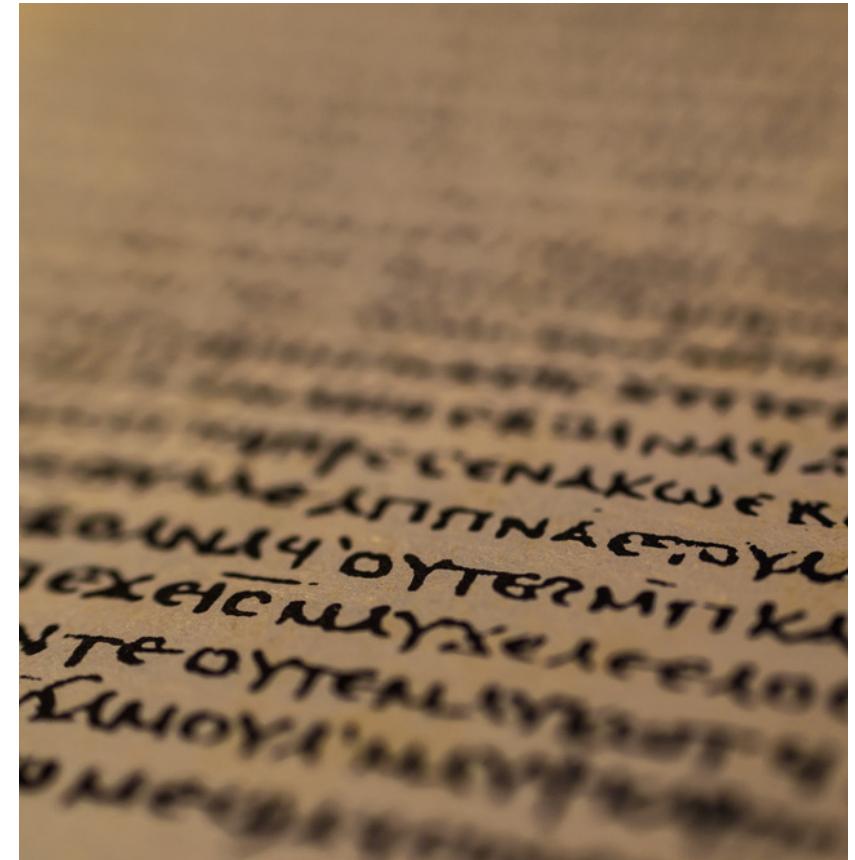
Exemplo:  
Criando um  
contexto para  
uma discussão  
mais rica.

- Um sistema altamente seguro provavelmente dobrará o custo do sistema como um todo.
- O quanto estamos dispostos a investir em termos de segurança no nosso sistema acadêmico?

# Pegadinha #3 - O questionário é técnico

---

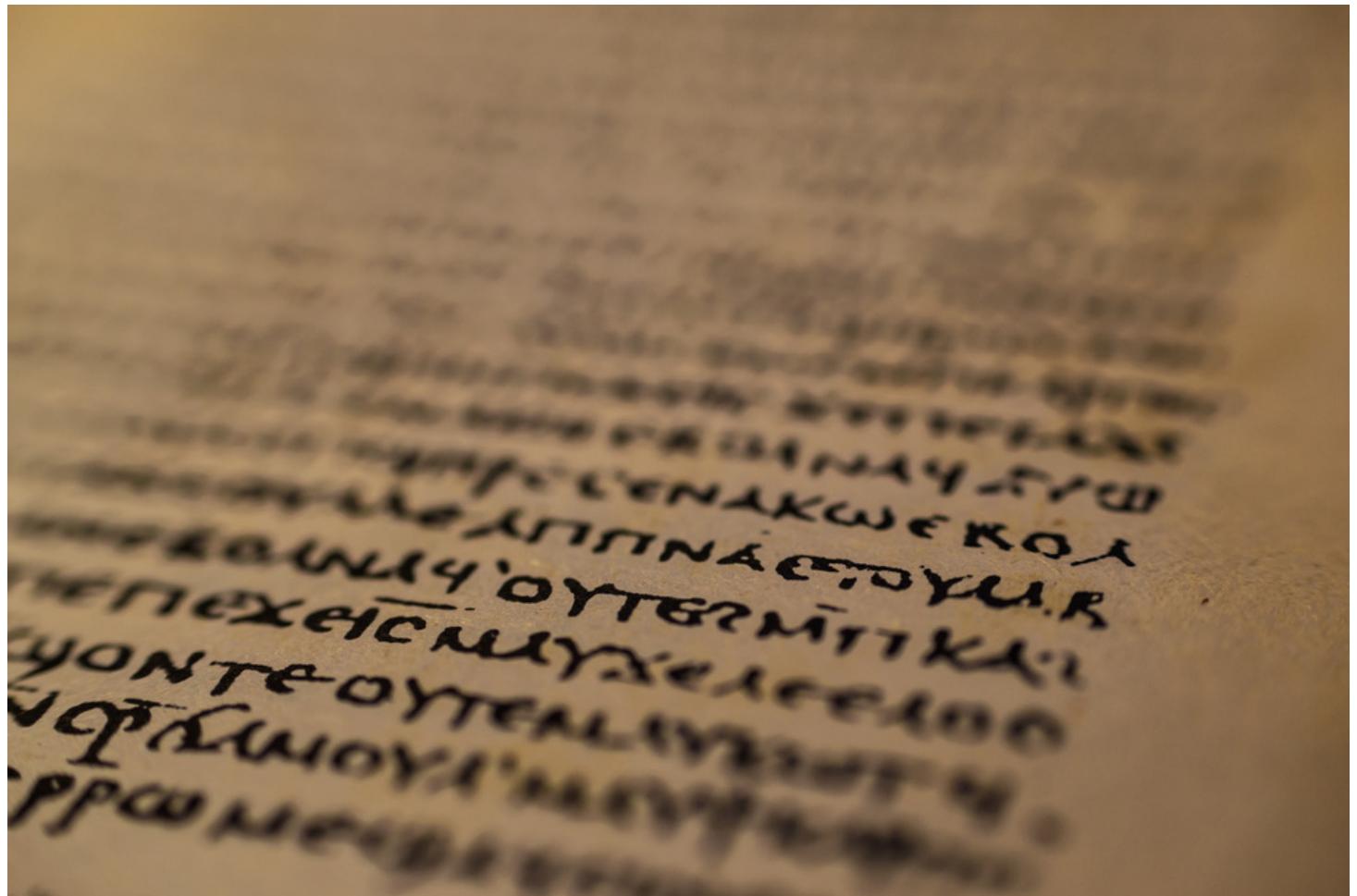
- Um questionário dado a uma parte interessada pode ser tratado como um documento técnico cujo conteúdo se encontra fora da área de interesse da parte interessada.
- A razão para essa percepção é que muitas partes interessadas estão mais confortáveis e mais familiarizadas com os conceitos relacionados aos negócios.
- Exemplo: Uma parte interessada típica entenderá o conceito de comprar uma passagem de madrugada, por exemplo, mas ficará menos à vontade falando sobre a disponibilidade de 99%.



## Pegadinha #3 - O questionário é técnico

---

- A dica é garantir que as partes interessadas compreendam o valor do tempo gasto respondendo às questões colocadas no questionário.
- Muitas vezes, é mais fácil mostrar esse valor dando exemplos de problemas que surgem na ausência de um questionário!



# Pegadinha #4 – Perguntas genéricas

- Outra armadilha é que os pedidos são muito gerais. Um desejo no nosso sistema acadêmico pode declarar que o tempo de resposta de qualquer transação deve ser menor que três segundos, por exemplo.
- No entanto, é mais provável que o tempo de resposta para transações específicas deva obedecer a essa solicitação específica, enquanto outras transações demoram significativamente mais tempo.

# Pegadinha #4

## – Perguntas genéricas

Se você não fizer solicitações tão específicas quanto possível, e se essas solicitações se tornarem requisitos, os elementos do sistema provavelmente serão superdimensionados, pois esses elementos serão criados com base em requisitos que podem não se aplicar.

## Pegadinha #5

### Os pedidos não são mensuráveis

- Todas as solicitações devem ser mensuráveis.





## Exemplo Pedidos não mensuráveis

- Um exemplo de uma solicitação que não pode ser medida é "O sistema acadêmico deve ser seguro".

Essa afirmação é ambígua porque a definição de "seguro" é subjetiva e simplesmente não pode ser medido.

## Pegadinha #5 – Os pedidos não são mensuráveis



- Você deve garantir que todos os pedidos declarados sejam específicos e mensuráveis, tanto quanto possível, embora algumas elaborações de desejos sejam permitíveis à medida que as solicitações são refinadas através do processo de requisitos.



## Pegadinha #6

### *Falar com as pessoas erradas*

- Outra armadilha às vezes ocorre quando todas as questões estão sendo colocadas para todas as partes interessadas.
- Muitos tipos de partes interessadas participam do processo de coleta de solicitações e você precisa fazer as **perguntas** certas para as **pessoas** certas.

# Resumo

- Não trate desejos como se fossem requisitos.
- Cuidado com a mentalidade de carrinho de compras.
- Não mantenha o questionário técnico.
- Não faça perguntas rasas e genéricas.
- Mantenha os requisitos mensuráveis.
- Envolva as pessoas certas nas conversas certas.





**PUC Minas**  
**Virtual**