

Arquitetura de Sistemas

Aula 4 - Definição de Requisitos

INTRODUÇÃO



Como vimos na aula 2, algumas metodologias tratam requisitos de forma menor, entendendo que, como mudarão até o fim do projeto, não vale a pena perder muito tempo neste trabalho. Outras metodologias, ao contrário, entendem que os requisitos devem ser levantados completamente até a exaustão, independentemente do tempo necessário para isso.

Nesta aula, veremos um conjunto de ações intermediárias desses dois conceitos. Nos dedicaremos, portanto, à apresentação do conceito que prega que levantar muito bem os requisitos é muito importante, e que eles devem ser validados através do conceito de prototipação e, ao mesmo tempo, não devem durar mais que 5% do tempo total do projeto.

Assim, apresentaremos uma nova maneira de se lidar com requisitos e sua relação com os resultados a serem alcançados, mostrando sua importância e aplicação no contexto da arquitetura de sistemas.

OBJETIVOS



Reconhecer a importância dos requisitos para melhorar o resultado na arquitetura de sistemas;

Analisar os elementos de processo e de desenvolvimento, na arquitetura de sistemas, que aceleram a definição de requisitos;

Identificar a relação entre os requisitos e o sucesso do projeto.

DEFINIÇÃO DE REQUISITOS

Podemos definir requisitos como condição ou exigência indispensável para um fim, uma ocupação e/ou uma realização.

Para começarmos é importante definirmos como as metodologias tratam requisitos e como essa disciplina fará a abordagem do tema, quais as diferenças encontradas no processo de levantamento de requisitos, no esforço gasto e em seus resultados, além de tratar da importância de requisitos para o resultado final do sistema.

Diferenças no levantamento dos requisitos:

Diagrama de funcionalidades de interface

Como dito na introdução desta aula, algumas metodologias tratam requisitos de forma menor, pois entendem que, como mudarão até o fim do projeto, não vale a pena perder muito tempo com eles, nem dedicar muito esforço nesse trabalho.

Nesses casos, os usuários e a equipe de desenvolvimento são chamados para novas definições e validações em cada parte do sistema que fica pronta.

Lembrando que para essas metodologias, os modelos conceituais de negócios e de sistemas são suficientes para que todos os envolvidos possam, cada um dentro de sua área de conhecimento, validar o que foi feito até ali.

Isso não é uma verdade absoluta, pois dificilmente um usuário leigo vai conseguir validar requisitos frente a modelos conceituais (negócio e sistemas) feitos para os iniciados em arquitetura de sistemas.

Requisitos vistos com muita importância

Outras metodologias, ao contrário, entendem que os requisitos devem ser levantados completamente até a exaustão, independentemente do tempo necessário para isso.

Nesse caso, o processo se torna moroso, caro e muito frustrante para os usuários, já que eles têm de participar de intermináveis reuniões de definição, sem que os resultados sejam visíveis aos seus olhos.

O mesmo ocorre com a equipe de desenvolvimento, que acaba por gastar muita energia e foco em algo que demora a ficar consistente.

No caso dessas metodologias, a validação de requisitos é feita através de textos longos e cansativos que, dificilmente, levarão os usuários a um correto entendimento de seus sistemas e a uma correta e consistente validação dos requisitos.

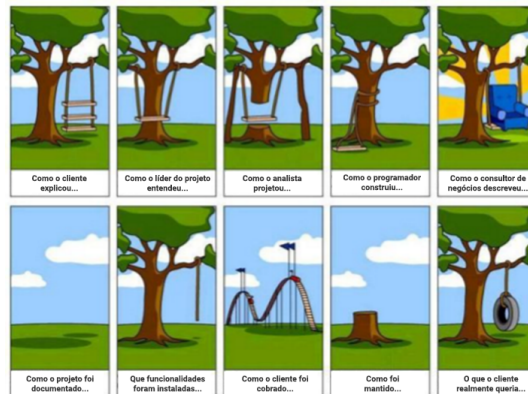
Conceito intermediário

Como dissemos, nesta disciplina, usaremos um conceito intermediário, em que o esforço total no processo de requisitos não pode ultrapassar os 5% do esforço total do projeto.

Além disso, os requisitos devem ser validados com um mecanismo que seja eficiente e eficaz para esse fim.

Essas definições de **esforço** e dos **mecanismos** veremos mais detalhadamente nos itens a seguir.

REQUISITOS DE SISTEMAS



Fonte: Domínio público (adaptação do autor)

Os requisitos de sistemas fazem parte da família de requisitos e compartilham sua definição básica. Porém, estão associados a funcionalidades e tipos que veremos mais detalhadamente ainda nesta aula.

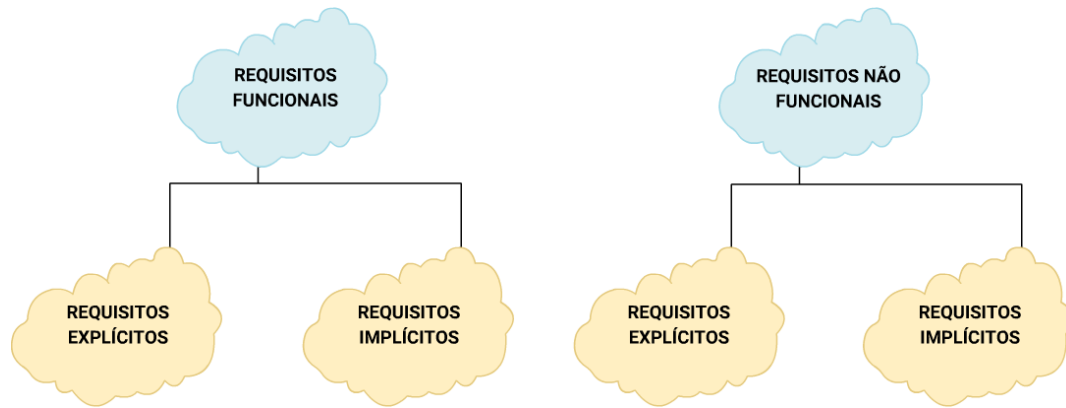
O conhecimento adquirido aqui servirá para que nossas partes interessadas e nossas equipes de desenvolvimento não enfrentem o problema apresentado no quadrinho.

ESFORÇO MÁXIMO E RESULTADOS ESPERADOS NA ETAPA DE REQUISITOS

É de se supor que gastar somente 5% do esforço do projeto em levantamento e validação de requisitos, como estamos propondo aqui nesta disciplina, nos leve a estar mais próximos das metodologias que dão pouca importância aos requisitos. Mas podemos garantir que isso está longe de ser verdade.

A delimitação de esforço máximo está associada a direcionar os trabalhos ao que realmente é importante para o levantamento, a validação de requisitos e também para o uso de ferramentas e técnicas baseadas em boas práticas de gestão e de desenvolvimento.

TIPOS DE REQUISITOS



Funcionais e não funcionais

Podemos separar os requisitos em duas categorias:

- • Requisitos funcionais, que são aqueles que tratam do que o sistema deve fazer, de sua funcionalidade;
- • Requisitos não funcionais, que tratam dos critérios que suportam os requisitos funcionais.

Como, por exemplo, qualidade para o software, os requisitos de performance, usabilidade, confiabilidade, robustez etc. Ou, então, os critérios relativos à qualidade para o processo de software ou uso de metodologias.

Implícitos e explícitos

Para os dois tipos de requisitos devemos, ainda, separá-los em:

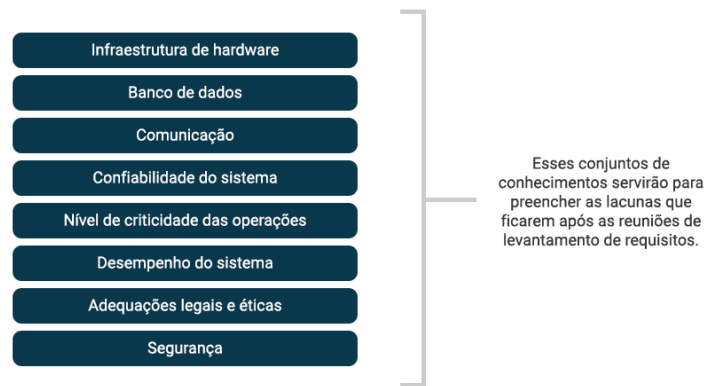
- • Requisitos implícitos, que são aqueles que o usuário vai conseguir externar e repassar à equipe de desenvolvimento;
- • Requisitos implícitos, que são aqueles que o usuário não vai conseguir repassar, ou por falta de conhecimento ou por entender ser muito óbvio e julgar que não vale a pena falar.

Saiba Mais

Para complementar as informações faltantes (requisitos implícitos), é necessário que a equipe de desenvolvimento tenha um framework de conhecimentos bem vasto. Dessa maneira, é capaz de preencher as lacunas, tanto para requisitos funcionais como para requisitos não funcionais.

FRAMEWORK DE CONHECIMENTOS EM TI COMO BASE PARA REQUISITOS

O framework de conhecimentos em TI tratado aqui refere-se aos conhecimentos que vão além dos requisitos, como, por exemplo:

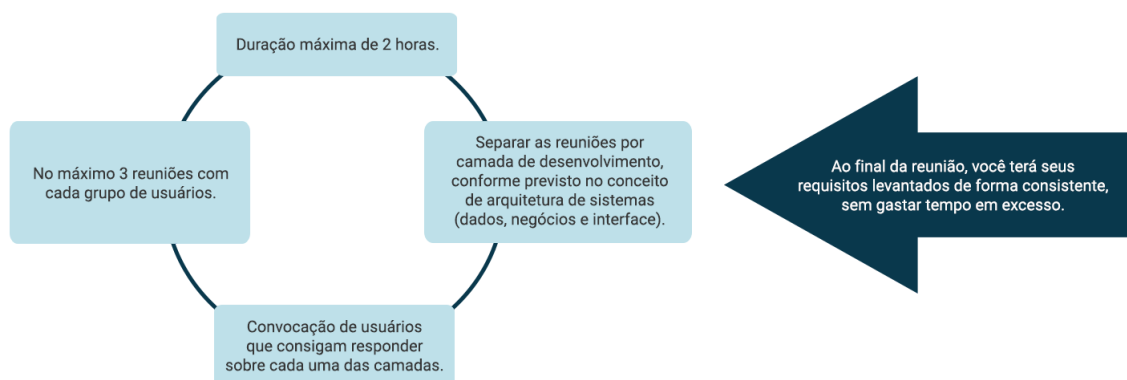


Atenção

Note que é normal que as lacunas existam, pois os usuários não conhecem e nem precisam conhecer elementos acima. Esse é um papel que obrigatoriamente deve ser desempenhado pela equipe de desenvolvimento, dentro do contexto previsto na arquitetura de sistemas.

REUNIÃO DE LEVANTAMENTO DE REQUISITOS

Os trabalhos de levantamento e validação de requisitos deve conter uma reunião de levantamento de requisitos, com as seguintes características:



VALIDAÇÃO DE REQUISITOS

A validação de requisitos deve ser feita em duas etapas, chamadas de pré-validação e validação de requisitos.

PRÉ-VALIDAÇÃO DE REQUISITOS

- Convoque uma reunião de 2h com todas as partes interessadas do seu projeto;
- Separe as partes interessadas do projeto em grupos de 4 integrantes;
- Gere uma cópia do documento de requisitos para cada um dos grupos;
- Cada grupo deve revisar o documento por no máximo 15 minutos, com as observações que acharem pertinentes;
- Fazer com que todos os grupos revisem todas as cópias dos documentos.

Ao final desses passos, você terá seus requisitos validados por todas as partes interessadas no projeto. Além disso, conseguirá o apoio e o patrocínio necessário, pois todos são responsáveis pelo documento gerado e não somente a equipe de desenvolvimento.

Saiba Mais

A nomenclatura técnica desta técnica é **“Técnica de Delphi”**.

A segunda etapa é desenvolvida dentro da prototipação que veremos detalhadamente agora:

PROTOTIPAÇÃO

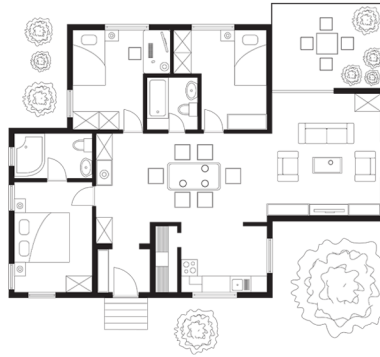
A prototipação é a melhor maneira de se validar requisitos.

Com essa técnica, conseguimos criar um **modelo conceitual forte** para ser validado pelas partes interessadas.

MAS O QUE É UM CONCEITO DE MODELO CONCEITUAL FORTE?

A melhor maneira de responder isso é fazer um comparativo entre modelos conceituais de outras áreas de conhecimento que tenha modelos conceituais fortes, como os modelos conceituais de negócio e de sistemas. A partir dessa comparação, fica mais fácil entender o conceito proposto.

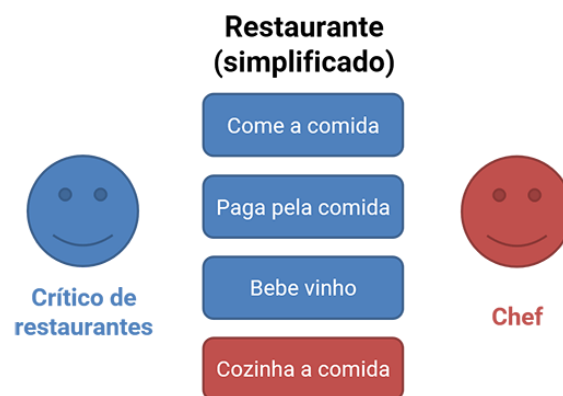
Por exemplo, se usarmos como base para exemplificar um modelo conceitual forte, podemos partir da arquitetura e de uma planta de casa:



Fonte da Imagem:

Não precisamos ser arquitetos para conseguir entender o conceito proposto a partir da planta (modelo). Isso representa a definição de um modelo conceitual forte.

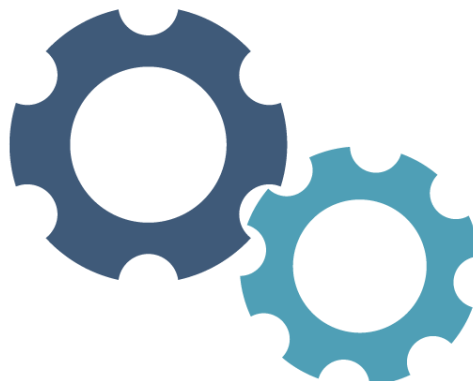
E O MODELO CONCEITUAL FRACO?



Quando olhamos para os diagramas de casos de uso, ou de classes, de objetos ou de componentes, verificamos que é necessário ser iniciado na arquitetura de sistemas para entender e validar se estão em conformidade com as necessidades do sistema a ser desenvolvidos (requisitos).

Isso representa um modelo conceitual fraco, ou seja, precisa de explicação externa para ser entendido.

COMO CRIAR PARA A TI UM MODELO CONCEITUAL FORTE?



Fonte da Imagem:

Como, então, criar para área de TI um modelo conceitual forte, que seja facilmente entendido pelas partes interessadas e, ao mesmo tempo, esteja em adequação às boas práticas da arquitetura de sistemas?

A resposta está na prototipação de sistemas!

É com o protótipo que as partes interessadas vão conseguir facilmente verificar se seus requisitos estão presentes na definição do que será feito.

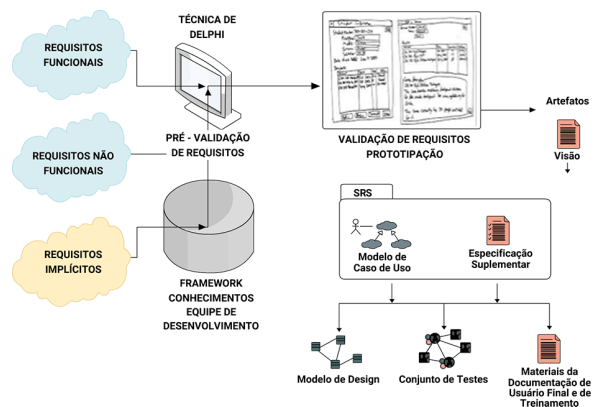
Atenção

É importante deixar claro às partes interessadas leigas que o protótipo é somente a casca e serve para validar funcionalidade e que ainda existe muito trabalho a ser feito para que o sistema fique pronto.

MODELAGEM CONCEITUAL DE NEGÓCIOS E DE SISTEMAS A PARTIR DOS REQUISITOS VALIDADOS

Agora que temos todos os requisitos levantados e validados de forma consistente e confiável pelas partes interessadas, podemos iniciar os trabalhos de modelagem conceitual de negócios e de sistemas.

Essa modelagem será feita de acordo com os elementos estudados na aula anterior.



RESULTADOS ESPERADOS



Fonte da Imagem:

Como resultado final desse conjunto de processos, espera-se que todas as necessidades das partes interessadas estejam presentes no sistemas desenvolvidos, na forma de funcionalidades, restrições e uso de tecnologias, atendendo plenamente a tudo o que se espera do novo sistema.

É de responsabilidade da equipe de desenvolvimento o atingimento dos resultados esperados.

1. A gestão de requisitos é uma etapa importante do projeto por ser um conjunto de atividades que tem como principal objetivo ajudar a equipe de projeto a:

☐

A) Utilizar ferramentas de engenharia de software para modelar os requisitos do sistema, através da UML.

☐

B) Identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue.

☐

C) Construir um modelo técnico refinado de funções, características e restrições do software.

☐

D) Negociar com os clientes os conflitos de prioridade de requisitos e identificar e analisar os riscos associados a cada requisito.

☐

E) Avaliar os requisitos quanto à qualidade, garantindo que ambiguidades, inconsistências, omissões e erros tenham sido detectados e corrigidos.

☐☐☐☐☐

Justificativa

2. Os requisitos não funcionais normalmente surgem por meio das necessidades dos usuários, podem ser restrições de orçamento, políticas organizacionais ou mesmo por fatores externos, como regulamentos de segurança e legislações de privacidade. Juntamente com a classificação dos requisitos não funcionais estão os requisitos de produto, os quais:

☐

A) Especificam ou restringem o comportamento do software, incluindo requisitos de desempenho, especificações de rapidez de execução e requisitos de confiabilidade que estabelecem, por exemplo, a taxa aceitável de falhas.

☐

B) São os requisitos gerais de sistemas derivados das políticas e procedimentos da organização do cliente e do desenvolvedor, como, por exemplo, os requisitos de processo operacional.

☐

C) Definem os requisitos do processo de desenvolvimento, como, por exemplo, a linguagem de programação, o ambiente de desenvolvimento ou normas do processo a serem usadas.

☐

D) Abrangem todos os requisitos que derivam de fatores externos ao sistema e seu processo de desenvolvimento. Podem incluir requisitos reguladores, que definem o que deve ser feito para que o sistema seja aprovado para uso.

☐

E) Incluem os requisitos legais, que devem ser seguidos para garantir que o sistema opere dentro da lei, e os requisitos éticos, que asseguram que o sistema será aceitável para seus usuários e o público geral.



Justificativa

Glossário