**Resumo - Cap. 3, 3.1, 3.2 e 3.3**

Os requisitos definem o que o sistema deve fazer e sob quais restrições. Os requisitos relacionados ao que o sistema deve fazer, ou seja, sua funcionalidade, são chamados de requisitos funcionais. Os requisitos relacionados à segunda parte, sob quais restrições, são chamados de requisitos não funcionais.

Em um sistema home-banking, os requisitos funcionais incluem, entre outras coisas, informar sobre o saldo e extrato da conta e fazer transferências. Os requisitos não funcionais estão relacionados à qualidade do serviço prestado pelo sistema, incluindo características como desempenho, nível de segurança, entre outros. Como Frederick Brooks expressou na frase que abre este capítulo, a definição de requisitos é uma etapa crítica na construção de qualquer sistema. Os problemas de especificação de requisitos também têm custos elevados. Eles podem exigir trabalho adicional quando for descoberto, depois que o sistema estiver pronto, que os requisitos foram especificados incorretamente ou que requisitos importantes não foram especificados. Os requisitos funcionais são frequentemente especificados em linguagem natural. Por outro lado, os requisitos não funcionais são especificados quantitativamente usando algumas métricas para evitar especificações genéricas.

Alguns autores, como Ian Sommerville, também classificam os requisitos entre os requisitos do usuário e do sistema. Os requisitos do usuário são requisitos de alto nível escritos pelos usuários, geralmente em linguagem natural e sem detalhes técnicos. Os requisitos do sistema são técnicos, precisos e escritos pelos próprios desenvolvedores. Normalmente, uma solicitação do usuário é expandida em um conjunto de requisitos do sistema. Assim, os requisitos do usuário estão mais próximos do problema, enquanto os requisitos do sistema estão mais próximos da solução.

Engenharia de requisitos é o nome dado a um conjunto de atividades relacionadas à descoberta, análise, especificação e manutenção dos requisitos do sistema. O termo engenharia é utilizado para reforçar que essas atividades devem ser realizadas de forma sistemática ao longo do ciclo de vida do sistema. As atividades relacionadas à descoberta e compreensão dos requisitos do sistema são chamadas de Elicitação de Requisitos. No contexto de TI, esse termo refere-se às interações dos desenvolvedores de sistemas com seus stakeholders, a fim de descobrir e entender os principais requisitos do sistema que está sendo construído. Diversas técnicas podem ser utilizadas para elicitar requisitos, por exemplo: entrevistas com stakeholders, aplicação de questionários, implementação de protótipos e análise de cenários de uso. Na Engenharia de Requisitos, a etnografia refere-se a uma técnica de elicitação de requisitos que recomenda que o desenvolvedor mergulhe no ambiente de trabalho dos stakeholders e observe como eles desenvolvem suas atividades. Uma vez obtidos os requisitos, eles devem ser documentados, validados, verificados e priorizados.

No desenvolvimento ágil, a documentação dos requisitos é feita de forma mais simples, por meio de histórias de usuários. Por outro lado, alguns projetos ainda exigem um documento de especificação de requisitos em que todos os requisitos do software a ser construído sejam documentados em linguagem natural. Em meados da década de 90, foi proposto um padrão para documentos de especificação de requisitos denominado IEEE Standard 830. Após sua especificação, os requisitos precisam ser verificados e validados com o objetivo de garantir que eles estejam corretos, precisos, completos, consistentes e verificáveis. Ao final, os requisitos devem ser priorizados. Às vezes, o termo requisitos é interpretado literalmente, ou seja, como uma lista de recursos obrigatórios e limitações do software. No entanto, as versões iniciais nem sempre implementarão o que os clientes especificam. Além disso, os requisitos também são mutáveis. Ademais, chamamos de rastreabilidade, é a capacidade de um determinado trecho de código identificar os requisitos implementados por ele e vice-versa.

Documentos de requisitos tradicionais possuem centenas de páginas e levam às vezes mais de um ano para ficarem prontos. Além disso, eles sofrem de problemas como: os requisitos podem mudar durante o desenvolvimento; descrições em linguagem natural podem estar ambíguas e incompletas, então os desenvolvedores têm que voltar a conversar com os clientes durante o desenvolvimento para tirar dúvidas. E quando essas conversas intermediárias acabam não ocorrendo, pode acarretar em riscos ainda maiores como por exemplo, quando o sistema estiver pronto, o cliente pode simplesmente achar que esse não é mais o sistema que ele queria, por *n* motivos próprios. Por isso, uma longa fase inicial de especificação de requisitos é cada vez mais rara. Por conta desses problemas, foi proposto uma técnica para solucioná-los, que ficou conhecida por Histórias de Usuários.

De acordo com Ron Jeffries, uma história de usuário é composta por três partes, todas começando com a letra C. São elas: cartão, conversas e confirmação. A parte Cartão, é usada por clientes para que eles descrevam uma funcionalidade que gostariam de ver implementada no sistema. As Conversas são o meio em que os clientes explicam e detalham, para os desenvolvedores, o que escreveram em cada cartão. E Confirmação é uma espécie de teste de alto nível para verificar se a história foi implementada corretamente. Dessa maneira, ao usar histórias de usuários, as atividades de engenharia de requisitos ocorrem durante o desenvolvimento, praticamente todos os dias. Como resultado, centenas de páginas de um longo documento de requisitos são substituídas por conversas frequentes em que o cliente explica os requisitos aos desenvolvedores da equipe.

Boas histórias devem possuir as seguintes características: ser independentes, ser abertas a negociação, devem agregar valor para o negócio, deve ser viável estimar seu tamanho, devem ser sucintas e devem ser testáveis. Essas características são lembradas mais facilmente pelo acrônimo INVEST(iniciais dos termos principais).