**Resumo**

**3.1 Introdução**

Os requisitos funcionais determinam o que o sistema deve fazer, enquanto os requisitos não-funcionais estabelecem as restrições sob as quais o sistema deve operar. Esses requisitos não-funcionais incluem desempenho, disponibilidade, segurança, privacidade, portabilidade e consumo de memória e disco. É importante lembrar que ambos os tipos de requisitos devem ser atendidos para que o sistema funcione adequadamente. Não adianta o sistema de home-banking criar todas as funcionalidades exigidas pelo banco. Mais deve possuir uma disponibilidade de 99,9%, a qual funciona, portanto, como uma restrição ao seu funcionamento.  
 A definição dos requisitos é uma etapa crucial no processo de construção de um sistema de software, como expresso por Frederick Brooks na sentença que abre o capítulo. Problemas na especificação dos requisitos podem ter um custo alto, levando a trabalhos extras, atrasos e até mesmo à rejeição do sistema pelos usuários. Portanto, é importante garantir que os requisitos funcionais e não-funcionais sejam especificados corretamente desde o início do processo de desenvolvimento.  
 Os requisitos funcionais são geralmente especificados em linguagem natural, enquanto os requisitos não-funcionais são quantificados com o uso de métricas. As métricas ajudam a evitar especificações genéricas e definem parâmetros quantitativos para o desempenho, espaço, confiabilidade, robustez, usabilidade e portabilidade do sistema. Por exemplo, em vez de simplesmente exigir que o sistema seja rápido e tenha alta disponibilidade, é preferível definir metas quantitativas, como um tempo de resposta máximo de 1 segundo para 99% das transações realizadas em uma janela de 5 minutos e uma disponibilidade de 99,99%.  
 Existem duas classificações comuns para requisitos: requisitos funcionais e não-funcionais, e requisitos de usuário e de sistema. Requisitos de usuário são escritos por usuários em linguagem natural e não incluem detalhes técnicos, enquanto requisitos de sistema são escritos pelos desenvolvedores e são mais precisos e técnicos. Os requisitos de usuário são normalmente expandidos em um conjunto de requisitos de sistema que detalham a solução técnica para atender ao requisito do usuário. Portanto, os requisitos de usuário estão mais próximos do problema a ser resolvido, enquanto os requisitos de sistema estão mais próximos da solução técnica.

**3.2 Engenharia de Requisitos**

A Engenharia de Requisitos é um conjunto de atividades que envolvem a descoberta, análise, especificação e manutenção dos requisitos de um sistema, devendo ser realizadas de modo sistemático e valendo-se de técnicas bem definidas. A Elicitação de Requisitos é uma das atividades da Engenharia de Requisitos e designa as interações dos desenvolvedores com os seus stakeholders para descobrir e entender os principais requisitos do sistema que se pretende construir.  
 Diversas técnicas podem ser usadas para elicitar requisitos, como entrevistas, questionários, workshops, protótipos e análise de cenários de uso, além de técnicas baseadas em estudos etnográficos. Etnografia, que tem sua origem na Antropologia, é uma técnica de elicitação que envolve o desenvolvedor integrar-se ao ambiente de trabalho dos stakeholders e observar suas atividades silenciosamente, sem interferir ou opinar sobre o que está sendo observado.  
 Depois de eliciados, os requisitos devem ser: (1) documentados, (2) verificados e validados e (3) priorizados. No desenvolvimento ágil, a documentação de requisitos é simplificada por meio de histórias de usuários, conforme estudado no Capítulo 2. No entanto, alguns projetos ainda exigem um Documento de Especificação de Requisitos, no qual todos os requisitos de software, incluindo requisitos funcionais e não funcionais, são documentados em linguagem natural (por exemplo, , português, inglês). Na década de 1990, foi proposto um padrão para Documentos de Especificação de Requisitos, denominado IEEE 830.  
 Os requisitos devem ser verificados e validados para garantir que sejam corretos, precisos, completos, consistentes e verificáveis. Os requisitos precisam ser corretos para evitar prejuízos, precisos para evitar ambiguidades, completos para não deixar de especificar requisitos importantes, consistentes para evitar conflitos entre stakeholders, verificáveis para testar se estão sendo atendidos e priorizados para considerar restrições de prazo e custos na implementação.  
 Adicionalmente, é importante considerar que os requisitos podem mudar ao longo do tempo devido a mudanças no ambiente externo, nos objetivos do negócio ou nas necessidades dos usuários. Portanto, é fundamental manter a documentação dos requisitos atualizada e realizar o rastreamento para garantir que as mudanças sejam refletidas tanto na documentação quanto no código fonte do sistema. A rastreabilidade permite identificar a relação entre os requisitos e o código fonte, facilitando a manutenção e a evolução do sistema.  
 A ilustração apresentada representa uma situação comum na Engenharia de Requisitos, que ocorre quando os requisitos de um sistema mudam com frequência ou os usuários não conseguem especificá-los de maneira precisa. Em casos como esses, é desnecessário investir muito tempo na elaboração de um Documento Detalhado de Requisitos, pois os requisitos podem ficar obsoletos quando o documento estiver pronto. Em vez disso, pode-se adotar Histórias de Usuários como documentos simplificados de especificação de requisitos e ter um representante dos clientes no time de desenvolvimento para esclarecer dúvidas e explicar os requisitos. Como essa é uma situação importante, a Seção 3.3 será dedicada ao estudo de Histórias de Usuários.

**3.3 Histórias de Usuários**

Os documentos de requisitos tradicionais, produzidos com a abordagem Waterfall, podem ter centenas de páginas e levar mais de um ano para serem concluídos. Entretanto, esses documentos apresentam problemas, tais como: (1) os requisitos podem mudar durante o desenvolvimento, tornando os documentos obsoletos; (2) a descrição dos requisitos em linguagem natural pode ser ambígua e incompleta, o que faz com que os desenvolvedores tenham que conversar com os clientes durante o desenvolvimento para esclarecer dúvidas; (3) caso essas conversas intermediárias não aconteçam, pode haver riscos de o cliente não ficar satisfeito com o sistema no final da codificação, pois suas prioridades, visão do negócio e processos internos podem ter mudado. Portanto, em sistemas comerciais, é cada vez mais raro ter uma fase inicial longa de especificação de requisitos.  
 Histórias de Usuários são uma técnica pragmática proposta por profissionais da indústria para solucionar problemas na Engenharia de Requisitos. Uma história é composta por três partes: o cartão, usado pelos clientes para escrever em poucas sentenças uma funcionalidade esperada no sistema; conversas entre clientes e desenvolvedores para detalhar o cartão; e confirmação, um teste de alto nível especificado pelo cliente para verificar se a história foi implementada conforme esperado. Métodos ágeis incluem um representante dos clientes no time de desenvolvimento para permitir uma comunicação constante e eliminar especificações textuais e completas de requisitos.  
 As especificações de requisitos por meio de histórias de usuário não se resumem a apenas duas ou três sentenças, como alguns críticos dos métodos ágeis podem afirmar. Na verdade, uma história de usuário é um lembrete do representante dos clientes para os desenvolvedores sobre um determinado requisito funcional que ele gostaria de ver implementado na próxima iteração ou sprint. Durante todo o sprint, o representante dos clientes se compromete a estar disponível para refinar a história e explicá-la para os desenvolvedores, além de considerar a história implementada desde que ela satisfaça os testes de confirmação que ele mesmo especificou. Essa é a maneira correta de interpretar uma história de usuário.  
 Na perspectiva dos desenvolvedores, o processo de histórias de usuários funciona com o representante dos clientes pedindo a história resumida no cartão, e a equipe se comprometendo a implementá-la no próximo sprint. O representante dos clientes está disponível para conversas frequentes e já definiu os testes que usará para considerar a história implementada. As histórias de usuários favorecem a comunicação verbal e são compatíveis com os princípios do Manifesto Ágil, que prioriza indivíduos e interações, software em funcionamento, colaboração com o cliente e resposta a mudanças.