

## **Gerência de Requisitos: O principal problema dos projetos de software**

José Roberto Blaschek

Aos requisitos estão associados os principais problemas do desenvolvimento de software<sup>1</sup>. Requisitos que não refletem as reais necessidades dos usuários, incompletos e/ou inconsistentes, mudanças em requisitos já acordados e a dificuldade para conseguir um entendimento comum entre usuários e desenvolvedores são as principais dificuldades relatadas, provocando re-trabalho, atrasos no cronograma, custos ultrapassados e a insatisfação dos clientes e usuários do software [2].

Muitos desses erros poderiam ser evitados se as organizações dispusessem de um processo de engenharia de requisitos definido, controlado, medido e aprimorado. No entanto, percebe-se que para muitos profissionais de Informática esses conceitos não são muito claros, o que certamente dificulta a ação dos gerentes no sentido de aprimorar os seus processos de desenvolvimento.

Este artigo aborda os conceitos de Requisito e de Processo de Engenharia de Requisitos, identificando suas principais atividades e um conjunto de boas práticas.

### **Conceito de Requisito e Engenharia de Requisitos**

Antes de abordar o Processo de Engenharia de Requisitos, é importante conceituar requisito, termo freqüentemente citado, debatido e utilizado na redação de contratos, sem que as partes possuam uma compreensão única de seu significado.

Os requisitos de um sistema de computação constituem uma especificação das características e propriedades do sistema ou uma descrição do que o sistema deve fazer, de como ele deve se comportar, bem como das suas restrições de operação.

É importante ressaltar que os requisitos descrevem “*o que o sistema deve fazer*” - e também “*o que ele não deve fazer*” - sem dizer “*o como fazer*”. Quando o requisito é expresso em termos do comportamento do sistema, este comportamento deve ser possível de ser percebido por um observador externo ao sistema. Como exemplo, pode-se citar, como forma correta e incorreta de descrever um requisito:

Forma correta: “o sistema deve informar os clientes com pagamento em atraso”.

Forma incorreta: “para cada pagamento existente no arquivo “pagamentos\_a\_vencer”, o sistema deve comparar a data de vencimento do pagamento com a data atual. Se a data de vencimento for anterior a data atual, o cliente deve ser incluído na relação de clientes com pagamento em atraso”.

Os requisitos de um sistema podem ser organizados em diferentes níveis de abstração: requisitos de negócio, requisitos de usuário e requisitos funcionais [3].

Os requisitos de negócio correspondem aos objetivos de negócio – ou do usuário – que devem ser satisfeitos pelo sistema. Normalmente são descritos através de um documento denominado visão ou escopo do sistema.

---

<sup>1</sup> 40% a 60% de todos os problemas encontrados em um projeto de software são causados por falhas ocorridas na fase de levantamento de requisitos [1].

Requisitos de usuário descrevem as atividades que os usuários deverão ser capazes de executar com a utilização do sistema.

Finalmente, os requisitos funcionais definem as funcionalidades que o sistema deve possuir para que os usuários possam executar suas atividades, de modo a atingir os objetivos do negócio.

Além destes, uma especificação de requisitos deve conter:

- a) Requisitos não funcionais: padrões, regulamentos e contratos com os quais o sistema deve ter conformidade; descrição de interfaces externas e requisitos de desempenho;
- b) Restrições: limitam as possibilidades de escolha do desenvolvedor no projeto e na implementação do produto (como por exemplo limitação de recursos e de infraestrutura); e
- c) Atributos de qualidade: ampliam a descrição das funcionalidades do sistema através da descrição de características de qualidade do produto, que sejam importantes para o cliente e para o desenvolvedor. A norma ISO/IEC 9126 define as características de qualidade para produtos de software: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade [4].

Antes de iniciar uma especificação de requisitos, é importante que a organização tenha definido um *template* do documento que deverá ser elaborado, visando um entendimento comum do que deve ser produzido. O IEEE Guide for Developing Systems Requirements Specifications é um exemplo de *template* detalhado, que pode ser utilizado com este propósito [5].

### **Processo de Engenharia de requisitos**

Para elaborar e manter uma especificação de requisitos é necessário que os desenvolvedores executem um conjunto estruturado de atividades destinadas a elicitar, documentar, analisar e validar requisitos. Este conjunto de atividades, iterativas por natureza, recebe o nome de Processo de Engenharia de Requisitos.

Quando a organização não dispõe deste processo formalmente definido e amplamente divulgado, os desenvolvedores elaboram as especificações de requisitos de uma forma empírica, executando atividades não padronizadas e definidas individualmente. Se isto ocorre, a qualidade da especificação dependerá exclusivamente da experiência e formação das pessoas, havendo assim uma elevada probabilidade de ocorrerem conflitos e re-trabalho.

As atividades que compõem o Processo de Engenharia de Requisitos e um conjunto de boas práticas que apóiam a sua execução serão a seguir apresentadas [3]:

- a) Elicitar – Elicitar requisitos é o nome usualmente atribuído à atividade voltada para descobrir (identificar, deduzir, extrair, evocar, obter) os requisitos de um sistema, através de entrevistas com os interessados pelo sistema, de documentos do sistema existente (manual ou automatizado), da análise do domínio do problema ou de estudos de mercado.

Na elicitação de requisitos constituem boas práticas a redação de uma declaração de visão e escopo do sistema; a definição dos procedimentos para desenvolvimento dos requisitos, a identificação das classes de usuários e dos diferentes grupos de interesse no sistema, a identificação dos casos de uso, a realização de sessões JAD, a análise dos

fluxos de trabalho dos usuários, a definição dos atributos de qualidade do sistema e o desenvolvimento de mecanismos que possibilitem o reuso de requisitos.

b) Analisar – Na análise, os requisitos elicitados são compreendidos e detalhadamente analisados por todos os interessados no sistema. Nessa atividade surgem muitos conflitos, sendo comum haver a necessidade de negociação para que os requisitos sejam aceitos por todos.

As boas práticas de análise referem-se à elaboração de um diagrama de contexto do sistema; criação de protótipos; análises de viabilidade; priorização dos requisitos e a criação de um dicionário de dados.

c) Documentar – Uma vez compreendidos, analisados e aceitos, os requisitos devem ser documentados com um nível de detalhamento adequado, produzindo a especificação de requisitos do software. Pode ser utilizada a linguagem natural ou diagramas, como os propostos pela UML.

São boas práticas da documentação de requisitos a adoção de *templates* adequados para especificação de requisitos; a identificação da fonte de todos os requisitos; a atribuição de um *label* para todos os requisitos e o registro das regras de negócio.

d) Validar – Após terem sido documentados, é necessário que os requisitos sejam cuidadosamente validados, principalmente quanto à consistência e a completude. Esta atividade visa identificar problemas nos requisitos, antes do início da construção. A importância desta atividade é caracterizada pelo fato de que a correção de um erro nesta fase possui um custo muito inferior do que a correção nas fases mais adiantadas do processo de desenvolvimento (até 200 vezes menor [1]).

Na validação dos requisitos devem ser utilizadas técnicas para inspeção e revisão da especificação de requisitos; redigidos casos de testes a partir dos requisitos e definidos os critérios de aceitação do sistema.

Para o gerente do projeto, responsável por atender às expectativas dos diferentes interessados no sistema, recomenda-se como boas práticas a seleção de um modelo de ciclo de vida apropriado, considerando que os modelos incrementais e iterativos – como o Unified Process [6] e Extreme Programming [7] - têm apresentado melhores resultados; elaboração do plano do projeto com base nos requisitos do sistema – é comum projetos comecem sem sequer terem escopo definido; renegociação dos compromissos tão logo se perceba que eles não possam ser cumpridos, gerenciamento dos riscos associados a requisitos e a criação de condições para que os requisitos possam ser rastreados ao longo do projeto.

## **Conclusão**

É importante que os gerentes de projeto reconheçam que não é possível desenvolver sistemas de qualidade, cumprir prazos e custos e atender às expectativas dos usuários sem ter um processo de desenvolvimento de requisitos definido, compreendido e utilizado por todos os desenvolvedores.

Convém ressaltar que não existe um processo ideal que possa ser utilizado em qualquer projeto. Será sempre necessário realizar adaptações, em função das características do sistema, dos usuários, da equipe, da organização, do tipo de sistema e da tecnologia a ser utilizada, dentre outros fatores.

## Referências

- [1] Leffingwell, D; Calculating the Return on Investment from More Effective Requirements Management; *American Programmer* 10(4); 13-16; 1997.
- [2] Kotonya, G; Sommerville, I; Requirements Engineering: Process and Techniques; John Wiley & Sons; 1998.
- [3] Wiegers K.E.; Software Requirements; Microsoft Press; 1999.
- [4] Weber, K.C; Rocha, A.R.C. da; Qualidade e Produtividade em Software; Makron Books,
- [5] Software Requirements Engineering, 2a Edição; Los Alaminos, CA: IEEE Computer Society Press; 1977.
- [6] Kruchten; P; The Rational Unified Process : An Introduction; Addison Wesley; 1998.
- [7] Kent, B; Extreme Programming Explained: Embrace Change; Addison-Wesley; 1999.

## URLs

<http://www.rational.com/products/rup/poster.jsp> - Rational Unified Process (RUP)

<http://www.extremeprogramming.org/> - Extreme Programming (XP)

<http://www.pmi.org/> - Project Management Institute

**José Roberto Blaschek** é Doutor em Ciências pela COPPE/Sistemas - UFRJ e Professor do Curso de Gerência de Projetos de Software da PUC-Rio. E-mail: **blachek@attglobal.net**

---

Esta coluna é produzida especialmente para a Developers' Magazine pelo ISLIG-Rio - Iniciativa Local do PMI-ISSIG, Grupo de Interesse em Sistemas de Informação do PMI.