

A IMPORTÂNCIA DOS REQUISITOS NA CONCEPÇÃO DE SOFTWARES DE QUALIDADE

Autores: AMANDA VIEIRA FARIAS, RYAN LUCAS SILVA LEMOS, IAGO VINÍCIUS SILVA COSTA, MÁRCIA ROBERTA PINTO MOURA, SIMONE CRISTINA MOREIRA, CHRISTINE MARTINS DE MATOS

Introdução

O desenvolvimento de *software* é uma atividade de fundamental importância na sociedade contemporânea, uma vez que o uso de computadores nas mais diversas áreas do conhecimento humano têm gerado uma sucessiva demanda por soluções computadorizadas. Entretanto, um grande desafio volta-se para a questão da qualidade no desenvolvimento de sistemas, em que o levantamento de requisitos é primordial para se alcançar os objetivos pretendidos pelos clientes e/ou usuários na solução de problemas organizacionais.

Aspirando a melhoria da qualidade dos produtos de *software* e a ampliação da produtividade no processo de desenvolvimento, despontou a área de Engenharia de *Software* nos primórdios da década de 70. Boehm (1983) a conceitua como a aplicação disciplinada de princípios e métodos no projeto e construção de *software* de qualidade de forma economicamente viável. Como parte dessa aplicação tem-se um conteúdo de fundamental importância e que nos últimos anos tem se valorizado, devido à grande demanda por *softwares* de fácil utilização e de ótima qualidade, tratando-se da denominada engenharia de requisitos.

Em termos metodológicos, o presente trabalho caracteriza-se pela abordagem sucinta da definição de requisitos e de Engenharia de Requisitos, bem como suas primordiais atividades, retratando ainda as principais classificações de requisitos de *software*. Em seguida, conceitua-se os Ambientes de Engenharia de *Software* e os conceitos e as principais características de *Capability Maturity Model Integration* (CMMI) e Melhoria de Processo do *Software* Brasileiro (MPS.BR.) Por fim, é feita uma definição e análise da questão de qualidade de *software*.

Material e métodos

A. Engenharia de Requisitos

De forma sucinta, Paula Filho (2000, p. 15) define “Os requisitos são as características que definem os critérios de aceitação de um produto. A engenharia tem por objetivo colocar nos produtos as características que são requisitos.”.

Sommerville (2007, p.82) define a Engenharia de Requisitos como “o processo de descobrir, analisar, documentar e verificar as funções e restrições de um sistema de *software*”. Pode-se ainda a definir como um procedimento que envolve todas as atividades que colaboram para a elaboração de uma documentação de requisitos e de sua manutenibilidade ao decorrer dos anos.

Na literatura, requisitos são condições básicas e necessárias para se obter algo ou para alcançar determinado propósito. Sendo assim, Sommerville (2007) classifica os requisitos em requisitos funcionais e requisitos não funcionais. O primeiro concerne sobre o que o sistema deve realizar, ou seja, suas funções e informações. Já o segundo concerne aos padrões que qualificam os requisitos funcionais, ou seja, estão associados ao uso da aplicação em termos de performance, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas.

B. Ambientes de Engenharia de Software

Silva (2005, p.2) define os Ambientes de Engenharia de *Software* (AES) “como uma coleção de ferramentas que fornece apoio automático, parcial ou total, às atividades de Engenharia de *Software*”. Ele ainda afirma que os AES podem auxiliar como importantes benefícios para uma organização, tais como: diminuição de custos, crescimento do rendimento, melhoria da gestão e maior qualidade do *software* ?nal.

O termo em questão pode ser aplicado a vários sistemas de alcances bem diferentes: desde um conjunto de poucas ferramentas executando sobre um mesmo sistema, até um ambiente totalmente integrado capaz de gerenciar e controlar todos os dados, processos e atividades do ciclo de vida de um *software*.

C. MPS.BR E CMMI

Devido as necessidades do mercado e a demanda sucessiva por *softwares* mais robustos e projetos paralelos nas empresas, despontaram modelos com a alegação de aumentar a qualidade dos processos organizacionais de desenvolvimento de *software* adequados em vários níveis de competência.

Mediante esses fatos, alguns modelos foram desenvolvidos para auxiliar a orientação de atividades que envolvam projetos de *software*. Sendo eles: *Capability Model Maturity (CMM)*; *Capability Maturity Model for Software (SW-CMM)*; *Capability Maturity Model Integration (CMMI)*; Melhoria de Processo de *Software* Brasileiro (MPS.BR); *Software Process Improvement and Capability Determination (SPICE)*; entre outros.

O modelo CMMI pode ser definido como um modelo de referência que caracteriza as práticas indispensáveis para o desenvolvimento e avaliação de maturidade de *software* em uma organização. Segundo Magno *et al* (2011) são normas tratadas neste modelo: gerenciamento de requisitos, manipulação de riscos, medição de desempenho, planejamento de trabalho, tomada de decisão, entre outros.

Ademais, o CMMI tem como propósito essencial funcionar de modo que possa conduzir em direção ao avanço dos processos da organização, fundamentando-se em atividades como a gestão do desenvolvimento de *software*, prazos e custos antecipadamente determinados. Portanto, o CMMI está intimamente relacionado com a produção de *software* com a máxima qualidade e a menor tendência a falhas.

Já o MPS.BR, objetiva estabelecer um padrão de melhoramento e avaliação de processo de *software*. É importante destacar que o MPS.BR ainda determina as regras com o intuito de promover a sua implementação e avaliação, oferecendo sustentação e garantia de que o MPS.BR venha a ser aplicado de maneira coerente com as suas condições. O âmago principal do MPS.BR está nas pequenas e médias organizações/empresas brasileiras de *software*. Geralmente empresas com poucos recursos e que necessitam melhorar radicalmente seus processos de *software* em um ou dois anos.

D. Qualidade de Softwares

Compreende-se como qualidade de um produto o seu grau de similaridade com os respectivos requisitos. Em conformidade com essa definição de qualidade, pode-se citar como exemplo um carro popular ser de ótima qualidade, e um carro de luxo ser de péssima qualidade.

“O que decide a qualidade é comparação com os respectivos requisitos: o confronto entre a promessa e a realização de cada produto. Geralmente a qualidade de um produto decorre diretamente da qualidade do processo utilizado na produção dele.” (PAULA FILHO, 2000, p. 19).

Certas características implícitas são ansiadas em todo *software* e norteiam o estágio de qualidade de um sistema. São vistos como os principais aspectos de qualidade de um *software*, as funcionalidades, a confiabilidade, a usabilidade, a eficiência, a manutenibilidade e a portabilidade. Portanto, a qualidade do produto de *software* está fortemente relacionada com a qualidade do processo que o produziu.

Resultados e discussão

Neste resumo foi retratado o que seriam os requisitos e de sua relevância no meio do desenvolvimento de *softwares*, levando em consideração a sua influência na Engenharia de Requisitos, que é um determinado processo que envolve quaisquer atividades que cooperam com o intuito de elaborar o documento de requisitos e sua manutenção no decorrer dos anos. Bem como se buscou trazer as principais classificações de requisitos, segundo a visão do autor Sommerville (2007).

Também foram apresentadas as arquiteturas de *softwares* que vem sendo utilizadas para o desenvolvimento de sistemas, cada vez maiores e complexos. Outro importante assunto retratado foram os Ambientes de Engenharia de *Software* que busca trazer uma maior clareza na elaboração de *softwares* e uma qualidade superior para os similares.

Atualmente, vive-se uma realidade de mercado, que possui uma deficiência enorme ligada à manutenção de *softwares* desenvolvidos, contudo, para ajudar na solução desse problema, os modelos de maturidade de Processos são fortemente utilizados. Sendo assim, foram apresentados dois modelos o MPS.BR e CMMI, onde buscou-se trazer uma breve definição e caracterização dos mesmos.

Por fim, fica inequívoco que um produto de *software* indica uma qualidade de acordo com o nível de contentamento dos anseios dos clientes perante todos os itens do produto, além, de uma indispensabilidade de levantamento de requisitos, com o máximo de detalhes e precisão na coleta das informações, como fator de garantia do resultado final: o produto de *software*.

Conclusão/Conclusões/Considerações finais

O desenvolvimento de normas de concepção tem se tornado uma das fontes fundamentais de progresso nas práticas de programação. Todavia, a evolução constante da dimensão de sistemas de *software* seguidora da complexidade tem concebido um conjunto de requisitos de maior confiabilidade, economia e desempenho.

A busca pela qualidade de *software* provoca a observação de uma série de fatores dos mais diversificados, englobando questões que vão a contar da tomada de decisões por parte dos clientes até a escolha de métricas mais acertadas, passando por detalhes de cunho provisório e econômico referentes aos investimentos a serem feitos pelas empresas de desenvolvimento.

Demonstrou-se por meio deste resumo como a análise de requisitos é de extrema importância para a confecção de *softwares* de qualidade, se fazendo indispensável em todo o processo de desenvolvimento de um sistema computacional, e ainda quais os aspectos a se observar, sendo eles frequentemente requisitos não funcionais, que definem diretamente na qualidade de *software*.

Realização:

SECRETARIA DE
DESENVOLVIMENTO
CIENTÍFICO, TECNOLÓGICO
E INOVAÇÃO

Apoio:



Referências

BOEHM, Barry W. **Software Engineering Economics**. Upper Saddle River, NJ, USA: Prentice Hall, 1983. Disponível em: <<http://csse.usc.edu/TECHRPTS/1984/usccse84-500/usccse84-500s.pdf>>. Acesso em: 17 maio 2017.

MAGNO, A. *et al.* **Modelos de Maturidade: CMMI, MPS-BR, PMMM**. Salvador: UNEB, 2011. 12f. Disponível em: <<http://docplayer.com.br/2829956-Modelos-de-maturidade-cmmi-mps-br-pmmm.html>>. Acesso em: 19 maio 2017.

PAULA FILHO, Wilson de Pádua. **Manual do Engenheiro de Software: Módulo técnico**. Belo Horizonte: UFMG, 2000.

SILVA, Renato Afonso Costa. **Inteligência Artificial Aplicada a Ambientes de Engenharia de Software: Uma Visão Geral**. INFOCOMP. Lavras/MG, v. 4, nº 4, dezembro 2005. Disponível em: <<http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/109>>. Acesso em: 20 maio 2017.

SOMMERVILLE, Ian. **Engenharia de Software**. 8 ed. São Paulo: Pearson Education do Brasil, 2007.