



ESTUDO DAS METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE E SUA ABORDAGEM QUANTO AOS REQUISITOS NÃO-FUNCIONAIS DE SISTEMAS

Wilson Bissi¹; Everton Luckesi de Souza²; Márcia Cristina Dadalto Pascutti³

RESUMO: A Engenharia de Software é uma abordagem para o desenvolvimento padronizado de software de qualquer natureza. Ela define um conjunto de metodologias, técnicas e ferramentas que quando bem empregadas auxiliam efetivamente o processo de desenvolvimento de software. Uma das principais etapas de um processo de software é a especificação de requisitos e uma grande dificuldade dessa etapa é a obtenção dos requisitos não funcionais do sistema, que dizem respeito a desempenho, performance, usabilidade, manutenibilidade, portabilidade. Dessa forma, o objetivo deste trabalho é o estudo das metodologias que abordam os requisitos não-funcionais do sistema e metodologias de desenvolvimento ágil, realizando análises e comparações entre elas, demonstrando suas principais características, com intuito de obter maior conhecimento nesta área, tornando-a mais difundida na comunidade e que futuramente sirva como base de apoio para outros artigos da área. Neste trabalho as metodologias de desenvolvimento de software adotadas para estudos foram a MERUSA (Metodologia de Especificação de Requisitos de Usabilidade e Segurança orientada para a Arquitetura do sistema), XP (*Extreme Programming*) e SCRUM, dentre elas as duas últimas são consideradas metodologia de desenvolvimento ágil. Os resultados obtidos são de grande valia, pois existe uma escassez de material relativo a essas metodologias e principalmente sobre a MERUSA, que trata de forma muito eficaz os requisitos não-funcionais de sistema.

PALAVRAS-CHAVE: Engenharia de Software; Metodologias de Desenvolvimento de Software; Requisitos Não-Funcionais; Sistemas de Informação.

1 INTRODUÇÃO

Com o mundo globalizado e o grande número de informações, é imprescindível o uso de sistemas que controlem essas informações de maneira segura, ágil e eficaz. Assim, muitas empresas devem se adequar a esta forma de trabalho ou ficarão fora do mercado.

Este trabalho tem por objetivo analisar uma metodologia de especificação de requisitos não-funcionais, a metodologia MERUSA, e compará-la com outras metodologias tratadas pela Engenharia de Software, aumentando assim o conhecimento

¹ Acadêmico do Curso de Análise e Desenvolvimento de Sistemas. Departamento de Informática, Centro Universitário de Maringá – CESUMAR, Maringá – PR. Bolsista do Programa de Iniciação Científica do Cesumar (PICC). wbissi@gmail.com

² Acadêmico do Curso de Análise e Desenvolvimento de Sistemas. Departamento de Informática, Centro Universitário de Maringá – CESUMAR, Maringá – PR. Participante do Programa de Iniciação Científica do Cesumar (PICC). evertonluckesi@hotmail.com

³ Docente do Curso de Análise e Desenvolvimento de Sistemas. Departamento de Informática, Centro Universitário de Maringá – CESUMAR, Maringá – PR. pascutti@cesumar.br

sobre o assunto, para que este se torne mais difundido à comunidade e servindo como base de apoio para futuros artigos da área.

A funcionalidade, sofisticação e o porte dos sistemas de informação têm crescido rapidamente e o software como produto ainda utiliza processos de produção insatisfatórios. São vários os prejuízos financeiros causados devido a erros gerados no processo de desenvolvimento de software (LUTZ, 2000).

Quando a “Crise do Software” foi identificada nos anos sessenta, foram dirigidos grandes esforços para encontrar as causas do problema. As pesquisas determinaram que as deficiências nos requisitos dos sistemas foram o principal contribuinte para o surgimento da “Crise do Software”. O desenvolvimento da especificação de requisitos em muitos casos parece trivial, mas provavelmente é a parte do processo que conduz a mais fracassos que quaisquer outras (SOMMERVILLE, 1993).

Os requisitos não-funcionais (RNF's) dos sistemas ainda não são tratados de forma sistemática pelas ferramentas disponíveis, uma vez que estes são difíceis de serem quantificados e geralmente interagem de maneira conflitante entre eles e com os demais requisitos funcionais do sistema. Os requisitos não-funcionais do sistema são relativos a desempenho, confiabilidade, disponibilidade, segurança, manutenibilidade, usabilidade, portabilidade, requisitos culturais e políticos, requisitos legais, também chamados de atributos de qualidade.

2 MATERIAIS E MÉTODOS

A obtenção das informações descritas neste trabalho se deu através do levantamento teórico metodológico, que consistiu no levantamento de técnicas para avaliação das metodologias MERUSA, XP e SCRUM. No levantamento foram consultadas várias bibliografias de livros e pesquisa em grupo de usuário em sites relacionados a essas metodologias, para observar a relevância com que cada metodologia tratava os cinco requisitos buscados neste trabalho, são eles: Usabilidade, Segurança, Performance, Portabilidade e Agilidade.

Ao ter conhecimento de como cada metodologia tratava os requisitos citados acima, foram realizadas pesquisas por artigos que demonstravam casos de sucesso ou fracasso com o uso dessas metodologias para analisar em que ponto cada uma “deixava a desejar”, para assim concluir esta pesquisa.

3 RESULTADOS E DISCUSSÃO

Inicialmente foram levantadas informações sobre as metodologias expostas neste trabalho, onde foram realizadas pesquisas bibliográficas pelos integrantes do projeto.

Após a obtenção destas informações foi elaborada a Tabela 1, que compara a relevância com que cada metodologia trata os requisitos não-funcionais. Esta tabela consiste na comparação dessas metodologias entre si de acordo com o material pesquisado e suas características, de maneira que se possa observar com mais clareza os pontos fortes e fracos de cada metodologia quanto aos RNF's, retratando assim a relevância de cada uma em seu contexto.

Tabela 1. Comparação entre as metodologias.

CRITÉRIOS	MERUSA	XP	SCRUM
Usabilidade	Muito	Muito	Muito
Segurança	Muito	Pouco	Pouco
Performance	Muito	Indefinido	Indefinido

V EPCC

CESUMAR – Centro Universitário de Maringá
Maringá – Paraná – Brasil

Portabilidade	Pouco	Muito	Muito
Agilidade	Pouco	Muito	Muito

Fonte: (BISSI; LUCKESI, 2006).

Os critérios abordados na Tabela 1 são requisitos relevantes para qualquer metodologia de desenvolvimento de software. As metodologias citadas neste trabalho atendem a todos esses requisitos, algumas dando mais ênfase, outras menos. Seguindo essa linha de pensamento, foi elaborada uma forma de avaliar como cada metodologia aborda esses requisitos, classificando sua aplicação em cada metodologia como: Pouco (requisitos abordados com pouca relevância); Muito (requisitos abordados com muita relevância) e Indefinido (requisitos não tratados nas metodologias). Essa classificação foi elaborada com base nas particularidades de cada metodologia fazendo um comparativo entre elas, e observando quais destas tinham vantagens/desvantagens sobre a outra.

4 CONCLUSÃO

Este trabalho apresentou a importância das metodologias de desenvolvimento de software, com enfoque às técnicas e abordagens na especificação de requisitos não-funcionais do software. Três metodologias em particular foram abordadas, sendo elas: MERUSA, XP e SCRUM.

Foram traçados comparativos em suas formas de abordagem de desenvolvimento de software e a avaliação de seus requisitos em suas fases de desenvolvimento, dentro de cada metodologia.

A MERUSA por se tratar de uma metodologia a ser aplicada a longo prazo, dentro dos requisitos estudados neste trabalho, dedica uma maior atenção aos requisitos relativos à Usabilidade, Segurança e Performance do sistema, é uma metodologia muito utilizada para o desenvolvimento de sistema integrados e distribuídos. Por outro lado, dedica menor atenção aos relativos à Portabilidade e Agilidade.

Já as metodologias XP e SCRUM dedicam mais atenção aos requisitos relativos à Usabilidade, Portabilidade e Agilidade. Elas estão entre uma nova geração de metodologias que priorizam a diminuição de tempo para a elaboração de sistema, são consideradas metodologias ágeis, devido a isso dedicam menor atenção aos requisitos de Segurança e Performance do sistema. São utilizadas no desenvolvimento de sistemas de qualquer porte, independente da quantidade de integrantes nas equipes de desenvolvimento, análise e especificação.

5 REFERÊNCIAS

LUTZ, R. R. **Software Engineering for Safety: A Roadmap**. ICSE'2000 – Proc. 22^o International Conference on Software Engineering, In: The Future of Software Engineering, Anthony Finkelstein (Ed.), ACM Press, Limerick, 2000.

SOMMERVILLE, I. **Integrating Ethnography into the Requirements Engineering Process**. First International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, 1993.