



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
BACHARELADO EM ENGENHARIA DE SOFTWARE



Título do trabalho

Igor Marques da Silva

Natal-RN
Dezembro 2014

Igor Marques da Silva

Título do trabalho

Monografia de Graduação apresentada ao Departamento de Informática e Matemática Aplicada do Centro de Ciências Exatas e da Terra da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de bacharel em Ciência da Computação.

Orientador(a)

Nome e titulação do(a) professor(a) orientador(a)

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA – DIMAP

Natal-RN

Dezembro de 2014

Sumário

1	Introdução	p. 3
2	Objetivos	p. 5
3	Métodos	p. 6
3.1	Estudo de aplicações existentes	p. 6
3.2	Inquérito contextual	p. 6
3.3	Sessões de interpretação da equipe	p. 6
3.4	Prototipação e implementação da ferramenta	p. 7
3.5	Implantação e observação do uso da ferramenta	p. 7
3.6	Avaliação da ferramenta	p. 7
4	Plano de Trabalho	p. 8
	Referências	p. 9

1 Introdução

No contexto de trabalhadores do conhecimento, como na área de desenvolvimento de software, é diária a troca de conhecimento com o objetivo de se obter um melhor desempenho para a organização inteira (DRUCKER, 1993) (WIIG; JOOSTE, 2004). A gerência de conhecimento lida com o reuso de conhecimento em suas diferentes formas, como: design de código, requisitos, modelos, dados, padrões e lições aprendidas (LEVY; HAZZAN, 2009). Dentre esse reuso, se encontra, por exemplo, como se deu a implementação de uma determinada funcionalidade em um determinado projeto de software.

Atualmente, as maneiras mais comuns de troca de informações entre desenvolvedores são via oral, escrita ou repasse de referências (documentação, links externos, etc) (STOREY et al., 2014) (OLSON; OLSON, 2000) (CUBRANIĆ et al., 2004) e, em alguns casos, pela natureza do meio, não se mantém registrada para consulta (OLSON; OLSON, 2000).

É comum também desenvolvedores implementarem funcionalidades semelhantes em diferentes contextos usando abordagens ad-hoc (SANGMOK, 2011). Nesse caso, desenvolvedores mais experientes em um determinado projeto tendem a atuar como mentores (CUBRANIĆ et al., 2004) e tal ato, como atividade de gerência de conhecimento, tende a consumir recursos (WIIG; JOOSTE, 2004).

A elaboração de uma ferramenta capaz de agregar referências de código e tarefas a soluções pode trazer enormes benefícios a equipes de desenvolvimento (CUBRANIĆ et al., 2004), visto que a rotatividade de membros nesse tipo de equipe geralmente está associada ao sucesso ou fracasso de projetos (HALL et al., 2008).

Este estudo propõe a elaboração de tal ferramenta com a participação de uma equipe real de desenvolvimento de software. Esta equipe, posteriormente, fará o uso do produto elaborado em seu contexto de trabalho.

A equipe em questão é a 4Soft, empresa júnior dos cursos de Engenharia de Software e Tecnologia da Informação da Universidade Federal do Rio Grande do Norte (UFRN). A empresa atua na área de desenvolvimento de software web para clientes de diversos

ramos e é formada exclusivamente por alunos dos cursos de Bacharelado em Engenharia de Software e Bacharelado em Tecnologia da Informação da UFRN (4SOFT,).

O estudo também prevê uma análise crítica dos impactos do uso na mesma no contexto de empresa.

2 Objetivos

Este trabalho tem por finalidade analisar que influência uma ferramenta de documentação de funcionalidades de software traz para equipes de desenvolvimento. Além disso, objetiva verificar como tal ferramenta pode contribuir para a redução significativa de tempo despendido para a explicação repetitiva de tarefas já executadas anteriormente e como a troca de conhecimento entre membros pode fluir de maneira melhor e com baixo custo.

A ferramenta está prevista de possuir os seguintes requisitos/:

1. Gerenciamento de catálogo de funcionalidades implementadas em um projeto com integração com repositório de código no GitHub
2. Anexo de fontes externas que auxiliem a implementação de determinada funcionalidade
3. Anexo de descrição de etapas da implementação de determinada funcionalidade
4. Recuperação de funcionalidades no catálogo
5. Anexo de variações da implementação de determinada funcionalidade

Este trabalho visa então responder às seguintes perguntas de pesquisa:

1. Quais são os requisitos para o suporte ferramental em apoio à transferência e reuso do conhecimento organizacional em empresas de desenvolvimento de software?
2. Que outras ferramentas existentes oferecem tal suporte?
3. Como estimular a adoção e o uso desse suporte ferramental em equipes de desenvolvimento de software?
4. Que outros benefícios e limitações tal suporte provê a equipes de desenvolvimento?

3 Métodos

Este trabalho prevê o desenvolvimento da ferramenta de documentação de implementação funcionalidades. Nela, será possível vincular recursos do repositório do projeto no GitHub (issues, commits, etc) à requisitos e suas implementações, bem como outros referenciais (links para perguntas no Stack Overflow, desenhos, por exemplo) de forma a gerar um guia ou tutorial de como realizar tal implementação novamente no futuro.

A empresa 4Soft terá participação significativa em todo o estudo, desde a concepção até no uso da ferramenta.

Os procedimentos para o estudo são os seguintes:

3.1 Estudo de aplicações existentes

Será feita uma busca por aplicações semelhantes. Suas limitações serão analisadas de modo que a ferramenta proposta as supra.

3.2 Inquérito contextual

Entrevistas e seções de brainstorm serão feitas com os participantes da empresa júnior mencionada. Será analisado como se dá seu processo de trabalho e como pode se dar o fluxo de atividades na ferramenta através de entrevistas e aplicação de questionários de satisfação.

3.3 Sessões de interpretação da equipe

Reuniões com a equipe de pesquisadores da UFRN que trabalharão no projeto serão feitas para definir o escopo da ferramenta, bem como seu se dará seu design e implementação

3.4 Prototipação e implementação da ferramenta

Nesta etapa, inicialmente, protótipos de baixa fidelidade serão elaborados. Posteriormente, serão expostos a todos os participantes e seu feedback será colhido e analisado. A partir daí, protótipos de maior fidelidade serão elaborados de maneira iterativa e incremental.

3.5 Implantação e observação do uso da ferramenta

A ferramenta então estará disponível para uso nos contextos descritos na seção anterior deste trabalho. A adesão dos desenvolvedores a ferramenta terá analisada nesta etapa, bem como seu uso (quantidade de artefatos de documentação criados, por exemplo) monitorado.

3.6 Avaliação da ferramenta

Ao fim do período anterior, uma nova bateria de entrevistas e aplicação de questionários de satisfação serão realizadas para avaliar qualitativamente como se deu a utilização da ferramenta, bem como se deram os efeitos de seu uso.

A versão final do trabalho será então redigida.

4 Plano de Trabalho

O cronograma de atividades se dará conforme o seguinte planejamento:

1. Estudo de aplicações existentes: Dezembro/2014
2. Inquérito contextual: Janeiro/2015
3. Sessões de interpretação da equipe: Janeiro/2015
4. Prototipação e implementação da ferramenta: Janeiro/2015 a Maio/2015
5. Implantação e observação do uso da ferramenta: Junho/2015 a Setembro/2015
6. Avaliação da ferramenta: Outubro/2015 a Novembro/2015

Referências

4SOFT. Disponível em: <<http://www.4softjr.com.br>>. Acesso em Novembro 29, 2014.

CUBRANIĆ, D. et al. Learning from project history: A case study for software development. In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2004. (CSCW '04), p. 82–91. ISBN 1-58113-810-5. Disponível em: <<http://doi.acm.org/10.1145/1031607.1031622>>.

DRUCKER, P. F. D. P. F. *Post-Capitalism Society*. 1st. ed. UK, Oxford: Butherworth-Heinemann, 1993.

HALL, T. et al. The impact of staff turnover on software projects: The importance of understanding what makes software practitioners tick. In: *Proceedings of the 2008 ACM SIGMIS CPR Conference on Computer Personnel Doctoral Consortium and Research*. New York, NY, USA: ACM, 2008. (SIGMIS CPR '08), p. 30–39. ISBN 978-1-60558-069-2. Disponível em: <<http://doi.acm.org/10.1145/1355238.1355245>>.

LEVY, M.; HAZZAN, O. Knowledge management in practice: The case of agile software development. In: *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2009. (CHASE '09), p. 60–65. ISBN 978-1-4244-3712-2. Disponível em: <<http://dx.doi.org/10.1109/CHASE.2009.5071412>>.

OLSON, G. M.; OLSON, J. S. Distance matters. *Hum.-Comput. Interact.*, v. 15, n. 2, p. 139–178, 2000.

SANGMOK, H. *Improved source code editing for effective ad-hoc code reuse*. Tese (Doutorado) — Massachusetts Institute of Technology, 2011. Disponível em: <<http://hdl.handle.net/1721.1/67583>>.

STOREY, M.-A. et al. The (r) evolution of social media in software engineering. In: *Proceedings of the on Future of Software Engineering*. New York, NY, USA: ACM, 2014. (FOSE 2014), p. 100–116. ISBN 978-1-4503-2865-4. Disponível em: <<http://doi.acm.org/10.1145/2593882.2593887>>.

WIIG, K. M.; JOOSTE, A. Exploiting knowledge for productivity gains. In: _____. [S.l.]: Springer, 2004. cap. Handbook on Knowledge Management, p. 289–308.