



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



Estudo e Implementação de uma Ferramenta de Documentação em Suporte à Reuso de Conhecimento em Equipes de Desenvolvimento de Software

Igor Marques da Silva

Natal-RN
Dezembro 2014

Igor Marques da Silva

Estudo e Implementação de uma Ferramenta de Documentação em Suporte à Reuso de Conhecimento em Equipes de Desenvolvimento de Software

Proposta de 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 Engenharia de Software.

Orientador

Prof. Dr. Fernando Marques Figueira Filho

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
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). Um dos casos de reuso é o conhecimento de como se deu a implementação de uma determinada funcionalidade em um determinado projeto de software. Exemplo: diferentes projetos podem implementar cadastro de usuário via Google ou Facebook, cada um com certas especificidades.

É comum também que essa implementação em diferentes contextos seja feita através de abordagens *ad-hoc* (reimplementação completa de trechos de código com poucas modificações e sem modularização visando reuso) (SANGMOK, 2011). Nesse caso, desenvolvedores mais experientes em um determinado projeto ou que já implementou tal funcionalidade tendem a atuar como mentores (CUBRANIĆ et al., 2004) e tal ato, como atividade de gerência de conhecimento, acarreta no despendimento de recursos, principalmente dos mentores (WIIG; JOOSTE, 2004).

Atualmente, as maneiras mais comuns de troca de informações entre desenvolvedores (incluindo, obviamente, a mentoria) são via oral, escrita ou repasse de referências (documentação, links externos, etc) (STOREY et al., 2014; OLSON; OLSON, 2000; CUBRANIĆ et al., 2004). Em alguns casos, pela própria natureza do meio, comunicação não se mantém registrada física ou virtualmente. Um dos exemplos de comunicação sem registro é a comunicação exclusivamente oral em que, ao fim do diálogo, as informações trocadas ficam apenas na memória dos envolvidos, sem maneiras de terceiros consultarem posteriormente aquelas informações (OLSON; OLSON, 2000).

Além disso, existe o fator da rotatividade de membros em equipes, sempre associado a custos de transferência de conhecimento e treinamento (HALL et al., 2008), possuindo

relação com sucesso ou fracasso de projetos de software (HALL et al., 2008).

Assim, 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). A ferramenta atua como um catálogo, agregando referências de código, informações externas e comentários fornecidos por desenvolvedores da equipe com o intuito de auxiliar outros desenvolvedores a buscar em fontes da própria equipe como se deram implementações de funcionalidades semelhantes às que já foram feitas em outros projetos.

Este estudo propõe a elaboração de tal ferramenta com a participação de uma equipe real de desenvolvimento de software. A equipe em questão é a 4Soft¹, empresa júnior² dos cursos de Engenharia de Software e Tecnologia da Informação da Universidade Federal do Rio Grande do Norte (UFRN³). 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 Software⁴ e Bacharelado em Tecnologia da Informação⁵ da UFRN.

O estudo também prevê uma análise crítica dos impactos do uso da mesma no contexto de empresa. Dentre os impactos previstos estão a redução da necessidade de mentoria ou consulta para os casos de reuso de funcionalidades em diferentes contextos. Tais impactos visam contribuir para a redução de gastos de recursos relacionados a tempo de implementação de funcionalidades, esforço de recuperação de informações, orientação de colaboradores e trabalho dos desenvolvedores que atuam como mentores dentro da equipe.

O Capítulo 2 descreve em detalhes o objetivo pretendido pelo trabalho e a quais perguntas pretende-se responder. Em seguida, o Capítulo 3 aborda a metodologia utilizada para a execução do estudo e que atividades serão desempenhadas pelos envolvidos. O Capítulo 4, por fim, descreve o cronograma das atividades descritas no capítulo anterior.

¹<http://www.4softjr.com.br>

²http://en.wikipedia.org/wiki/Junior_enterprise

³<http://www.ufrn.br>

⁴<http://www.dimap.ufrn.br/pt/graduacao/engenharia-de-software/apresentacao>

⁵http://www.imd.ufrn.br/curso_bacharelado.php

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.

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 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 GitHub¹ (*issues*, *commits*, etc) a requisitos e suas implementações, bem como outros referenciais (*links* para perguntas no Stack Overflow², 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.

As etapas do estudo são:

Estudo de aplicações existentes

Será feita uma busca por aplicações que realizem atividades semelhantes às propostas. Suas limitações serão analisadas pela equipe de pesquisa e um panorama inicial será traçado de modo que a ferramenta proposta possa suprir as necessidades iniciais e as limitações encontradas.

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. As informações necessárias serão coletadas através de entrevistas e aplicação de questionários de satisfação.

Sessões de interpretação da equipe

Reuniões com a equipe de pesquisadores que trabalharão no projeto serão feitas para definir com mais detalhes o escopo da ferramenta, bem como será sua arquitetura e implementação.

¹www.github.com

²www.stackoverflow.com

Prototipação e implementação da ferramenta

Nesta etapa, inicialmente, protótipos de baixa fidelidade serão elaborados pelo pesquisador. Posteriormente, serão expostos a todos os participantes do projeto (empresa júnior e pesquisadores) e seu *feedback* será colhido e analisado. A partir daí, a aplicação passará para a etapa de implementação seguindo processo iterativo e incremental de desenvolvimento de software.

Implantação e observação do uso da ferramenta

A ferramenta então estará disponível para uso de todos os membros da empresa 4Soft. A adesão dos desenvolvedores a ferramenta será analisada nesta etapa, bem como seu uso monitorado (quantidade de artefatos de documentação criados, por exemplo).

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 e a satisfação de seus usuários. A versão final do trabalho com os resultados 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

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