

Gestão de conhecimento aplicando na melhoria de processos de teste de software

Érica Ferreira de Souza¹, Nandamudi L. Vijaykumar²

¹Programa de Doutorado em Computação Aplicada – CAP
Instituto Nacional de Pesquisas Espaciais – INPE

²Laboratório Associado de Computação e Matemática Aplicada – LAC
Instituto Nacional de Pesquisas Espaciais – INPE

{erica.souza, vijay}@lac.inpe.br

Abstract. *Software development organizations are seeking to add quality to the generated products. Testing processes are strategic elements to manage projects and product quality. Therefore, such organizations have shown a growing interest in mechanisms to improve these processes, for example, knowledge management. This paper presents a literature review focusing on the following issues: testing processes and knowledge management.*

Resumo. *Organizações de desenvolvimento de software vêm buscando, cada vez mais, agregar qualidade aos produtos gerados. Os processos de teste são elementos estratégicos para a condução de projetos e qualidade do produto. Diante disso, tais organizações têm mostrado um crescente interesse por mecanismos de melhoria desses processos, como, por exemplo, gestão do conhecimento. Este trabalho apresenta uma revisão da literatura focando nos assuntos: processo de teste e gestão de conhecimento.*

Palavras-chave: *Processos de teste, gestão de conhecimento e melhoria de processos.*

1. Introdução

Durante as últimas décadas, tem-se visto o desenvolvimento de software cada vez mais moderno, devido ao surgimento de novas tecnologias em diversas aplicações, e com isso, processos para desenvolver e qualificar o software, devem ser embutidos. Isto vale para aplicações críticas, como espaciais, sistemas de controle em usinas nucleares, equipamentos médicos e outros tão críticos.

A atividade de teste de software, devido a sua grande importância, tem despertado interesse de pesquisadores e profissionais na área. Assim como o processo de desenvolvimento de software, a atividade de teste também está incorporada a um processo, pois é composta por diversas etapas com o objetivo de agregar qualidade aos produtos gerados [Bastos et al. 2007]. Sendo a atividade de teste um processo, a melhoria deste processo deve ser considerada.

A comunidade acadêmica tem trabalhado em dados coletados ao longo dos projetos, que podem fornecer informações importantes para a tomada de decisão, envolvendo a identificação e realização de ações corretivas e que podem promover melhoria dos processos. É nesse contexto que a gestão do conhecimento (GC) tem apresentado um papel

fundamental na melhoria de processos de software. Para lidar com os temas acima abordados, este trabalho apresenta uma revisão da literatura e estado da arte focando nos assuntos: processo de teste e gestão de conhecimento.

1.1. Qualidade dos Processos de Teste

Testar um software significa avaliar, através de uma execução controlada, se o seu comportamento ocorre de acordo com o que foi especificado [MYERS 2004]. A partir da década de 90, o teste de software começou a ser tratado não mais como uma atividade do processo de desenvolvimento, e sim como um processo independente.

O processo de teste é composto por diversas etapas ou fases que têm início simultâneo com o processo de desenvolvimento de software. Cada etapa de teste é realizada mediante uma série de técnicas de teste e diversos critérios pertencentes a cada uma delas. As três principais técnicas de teste utilizadas são: Teste Estrutural (ou caixa-branca), Teste Funcional (ou caixa-preta) e Teste Baseado em Defeitos [Delamaro et al. 2007].

A área de melhoria dos processos tem sido investigada sistematicamente, dadas as evidências de que a qualidade do processo pode influenciar significativamente na qualidade do produto final. Nesse sentido, a melhoria dos processos é um desafio importante para as organizações. Diversas abordagens vêm sendo desenvolvidas ao longo dos anos para a implantação de métodos e modelos para melhoria de processos. Algumas das abordagens mais conhecidas são: Ciclo *Plan/Do/Check/Action* (PDCA), Método *Goals Questions (Indicators) Measures* (GQ(I)M), Modelo *Initiating, Diagnosing, Establishing, Acting and Learning* (IDEAL) e os modelos de maturidade [MPS.BR 2005].

1.2. Influência da Maturidade do Processo na Qualidade do Produto

A comunidade científica têm sido motivada a desenvolver pesquisas buscando garantir a qualidade do produto de software. As organizações podem promover a melhoria do processo de software, por meio do aprimoramento da capacidade dos processos ou da maturidade da organização. Vários trabalhos relacionados com o tema melhoria do processo são apresentados nos últimos anos, comprovando a influência da maturidade do processo com a qualidade do produto.

Na área de teste de software e melhoria dos processos de teste, por serem fatores críticos na qualidade do produto, observa-se um aumento nas pesquisas relacionadas ao tema. [Lamas 2010], propôs o detalhamento de um estrutura de maturidade operacional para Gestão de Teste de Software. A estrutura visa a gestão de defeitos, assim como a aferição de resultados de teste de software. Para o propósito foi criado um *framework* denominado *Organizational Testing Management Maturity Model* (OTM3) e foi utilizado como estudo de caso o projeto de Integração e Cooperação Amazônica para Modernização do Monitoramento Hidrológico (ICA-MMH).

No projeto ICA-MMH, foi gerado um conjunto significativo de dados de teste a partir do processo de teste que segue a estrutura do OTM3. Para análise desses dados, no trabalho de [Ferreira 2010], foram utilizadas técnicas de mineração de dados, a fim de levantar informações relevantes para melhoria do processo de teste. O objetivo era identificação de regras ou correlação entre os dados de teste. O trabalho mostrou como resultados preliminares, a coerência existente entre os padrões encontrados e o processo de teste, e também mostrou ser um bom ponto de partida para realizar estudos semelhantes.

2. Gestão de conhecimento na qualidade do software

Com o crescimento exponencial de dados vindos de diversas fontes de conhecimento distintas dentro da organização, torna-se necessário a automatização das tarefas de aquisição, processamento, análise e disseminação do conhecimento. As organizações necessitam gerenciar de forma efetiva as informações geradas em seu ambiente de produção para promover a melhoria contínua dos processos.

Diante deste contexto, a GC tem se posicionado para exercer um papel fundamental para as organizações. O principal objetivo da GC é promover o surgimento de conhecimento novo, seu armazenamento e compartilhamento por toda a organização [O’Leary and Studer 2001]. Na área de engenharia de software não é diferente. Atualmente, diversas organizações de desenvolvimento de software buscam utilizar o conhecimento para desenvolverem competências específicas e capacidade inovadora, que se traduzem em serviços com menor custo e tempo, e produtos com maior índice de qualidade.

No ambiente de engenharia de software transformar toda a informação em conhecimento aplicável não é uma tarefa fácil por se tratar de um grande volume. O maior desafio para gerenciar o conhecimento é que boa parte do conhecimento é tácito (não documentado) e, dificilmente, são dedicados esforços para torná-lo explícito. Surge então a necessidade de codificar o conhecimento, para torná-lo acessível e tratável. Para apoiar as atividades envolvidas na GC, diversas tecnologias são aplicadas, e uma tecnologia particularmente importante e muito estudada nos últimos anos é a ontologia.

De acordo com [Gruber 1993], a ontologia é uma especificação formal explícita de uma conceituação compartilhada. O estabelecimento de ontologias pode contribuir para a estruturação e classificação das informações. O grande interesse em ontologias, hoje, dá-se pela necessidade de haver, cada vez mais, uma maior interoperabilidade e reutilização de informações entre os sistemas e as pessoas dentro de uma organização.

Ontologias são apontadas, hoje, como sendo cruciais na GC em melhoria dos processos. Muitos dos trabalhos que exploram o uso da GC e ontologia se baseiam no conceito de experiências passadas. Em um ambiente de desenvolvimento de software, por exemplo, o reuso das experiências e conhecimento coletados podem ajudar a evitar que falhas se repitam e auxilia na solução de problemas recorrentes do processo aplicado, promovendo sua melhoria.

2.1. Influência da GC e sua Representação na Qualidade do Software

Grande parte dos trabalhos encontrados na literatura que envolvem GC e ontologia, estão relacionados com o processo de desenvolvimento de software e enfocam a reutilização de experiências e processos. São poucos os trabalhos relacionados à melhoria do processo de teste de software, e os trabalhos encontrados que utilizam de tal abordagem são recentes.

De acordo com [Wu and Xuemei 2009], a aplicação da GC de forma efetiva no processo de teste de software é a chave para melhorar a qualidade dos testes realizados. [Wu and Xuemei 2009] propôs um modelo de GC para processo de teste de software baseado em experiências e habilidades de integrantes da equipe. Os elementos que fazem parte do modelo são: organização da equipe de teste, conhecimento de teste e comunicação. A ideia do modelo é construir uma base de conhecimento com experiências relatadas pelos

membros da equipe de teste. Segundo [Wu and Xuemei 2009], é difícil encontrar trabalhos que abordam a GC aplicada no campo de teste de software.

Tecnologias-chave são discutidos no trabalho de [Desai 2011], tais como, sistema de GC, ciclo de GC e teste de software. Em geral, o autor do trabalho descreve que a reutilização do conhecimento na área de teste de software envolve módulos e casos de teste. [Desai 2011] também destaca a perda de conhecimento que ocorre quando um indivíduo experiente deixa a organização e cita a GC como uma possível solução.

3. Conclusões

O presente trabalho apresentou a influência da GC na qualidade do software, principalmente como essa tecnologia é aplicada na melhoria de processos. Mesmo sendo um fator crítico de qualidade do produto, o processo de teste compõe-se de poucas pesquisas nesse sentido. A literatura ainda carece de trabalhos que estabeleçam estratégias de melhorias no processo de teste a serem usadas na tomada de decisões. É interessante envolver pesquisas voltadas para criação de modelos de suporte à decisão em ambientes de teste de software que propicie a representação e o armazenamento do conhecimento.

Referências

- Bastos, A., Rios, E., Cristalli, R., and Moreira, T. (2007). *Base de conhecimento em testes de software*. Martins Editora Livraria, São Paulo, 2 edition.
- Delamaro, E., Mandonado, J. C., and M., J. (2007). *Introdução ao teste de software*. Ed. Elsevier, Rio de Janeiro.
- Desai, A. (2011). Knowledge management and software testing. *International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) - TCET*, pages 767–770.
- Ferreira, E. (2010). Uso de mineração de dados para análise do processo de teste. Technical report, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos - SP. Acesso em maio 2011.
- Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *In Formal Ontology in Conceptual Analysis and Knowledge Representation*.
- Lamas, E. A. (2010). Uma estrutura de maturidade operacional para gestão de teste de software aplicada a um projeto de monitoramento hidrológico. Master's thesis, Dissertação de mestrado do curso de Engenharia Eletrônica e Computação - ITA, São José dos Campos - SP.
- MPS.BR (2005). Melhoria de processo do software brasileiro. Acesso em abril 2011.
- MYERS, G. J. (2004). *The art of software testing*. John Wiley and Sons, Canada, 2 edition.
- O'Leary, D. and Studer, R. (2001). Knowledge management: An interdisciplinary approach. *IEEE Intelligent Systems*, 16, No. 1.
- Wu, Y. L. J. and Xuemei, L. G. G. (2009). Investigation of knowledge management methods in software testing process. *International Conference on Information Technology and Computer Science*, 2:90–94.