

**XIX** encontro nacional  
de pesquisa em  
ENANCIB ciência da informação

// SUJEITO INFORMACIONAL E AS  
PERSPECTIVAS ATUAIS EM CIÊNCIA  
DA INFORMAÇÃO. //

**22-26**  
**OUTUBRO**  
**2018**  
LONDRINA/PR



## XIX ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2018

### GT-4 – Gestão da Informação e do Conhecimento - Pôster

#### CONHECIMENTO EXPLÍCITO, DESAFIOS PRODUTIVOS E DESENVOLVIMENTO DE SOFTWARE

Leonardo Pereira Pinheiro de Souza (Universidade Estadual Paulista, UNESP)

Cássia Regina Bassan de Moraes (Universidade Estadual Paulista, UNESP)

*EXPLICIT KNOWLEDGE, PRODUCTIVITY CHALLENGES AND SOFTWARE DEVELOPMENT*

#### Modalidade da Apresentação: Pôster

**Resumo:** Desenvolver *softwares* exige conhecimentos técnicos, de negócio e trabalho em equipe. Explicitar estes conhecimentos é necessário para a comunicação e para facilitar a análise do código. Devido às pressões mercadológicas, empresas de *software* necessitam otimizar custos e tempo de produção. O Manifesto Ágil trouxe desburocratização e aumento da rapidez no trabalho. Contudo, há evidências na literatura de que a documentação de *software* é comumente negligenciada. O objetivo da presente pesquisa é verificar o papel da informação e do conhecimento em uma empresa de desenvolvimento de *software*, enfatizando a documentação de projetos. A presente pesquisa é um estudo de caso com abordagem qualiquantitativa, realizado em uma empresa do município de Marília, São Paulo, onde se situa um Arranjo Produtivo Local de Tecnologia da Informação. O questionário é utilizado para coleta de dados, além das técnicas de entrevista e observação do ambiente de trabalho, verificando valores e práticas referentes à documentação. Como resultados, há uma tendência de privilegiar a comunicação face a face, com menor ênfase na documentação, visando a produtividade. Esta tendência pode afetar a manutenção de sistemas, visto a relevância desta documentação para o compartilhamento e internalização de conhecimento explícito e informação, o que contribui para a aprendizagem organizacional.

**Palavras-Chave:** Conhecimento; Documentação; Engenharia de Software.

**Abstract:** Software development requires technical and business knowledge and also team work. To make this knowledge explicit is necessary for communication and to facilitate code analysis. Due to market pressures, software companies need to optimize costs and production time. The Agile Manifesto brought less bureaucracy and increased speed at work. However, evidence from the literature demonstrates that software documentation is commonly neglected. The purpose of this

research is to verify the role of information and knowledge in a software development company, emphasizing project documentation. This research is a case study with qualitative and quantitative approaches, executed in a company located in Marília city, São Paulo, where the Arranjo Produtivo Local de Tecnologia da Informação is located. Questionnaires are used for data collection, as well as interviews and work environment observation, verifying values and practices related to documentation. As a result, there is a tendency to favor face-to-face communication, with less emphasis on documentation, aiming at productivity. This trend may affect systems maintenance, considering the relevance of documentation for the sharing and internalization of explicit knowledge and information, which contributes to organizational learning.

**Keywords:** Knowledge; Documentation; Software Engineering.

## 1 INTRODUÇÃO

A elaboração de *softwares* comerciais requer sólidos conhecimentos técnicos e de negócios. Estes *softwares* são complexos, exigindo trabalho em equipe e informações consistentes, que devem ser documentadas. A documentação é necessária para: definir funcionalidades, comunicar ideias, realizar manutenções e guiar a inspeção do sistema (PARNAS, 2008). Alguns requisitos de qualidade são imprescindíveis para a documentação. Para Garousi et al. (2015) estes critérios incluem: completude, consistência e acessibilidade. Para Briand (2003), nas empresas de *software* a rapidez e lucratividade são priorizados, devendo-se equilibrar os custos de criação e manutenção da documentação. O fator tempo é tão crítico que surgiu o Manifesto ágil, para desburocratizar e intensificar a velocidade de produção. Segundo Dingsoyr et al. (2012) este manifesto motivou a minimização de trabalhos considerados desnecessários, restringindo a documentação.

Portanto, colocam-se os seguintes problemas: como se configuram as reais práticas de documentação na rotina das empresas de *software*? Como o enfoque na rapidez e produtividade afeta a consistência da documentação? O objetivo da presente pesquisa é verificar o papel da informação e do conhecimento em uma empresa de desenvolvimento de *software*, enfatizando a documentação de projetos.

## 2 PROJETO DE SOFTWARE, DOCUMENTAÇÃO, INFORMAÇÃO E CONHECIMENTO

O processo de desenvolvimento de *software* inicia-se antes da programação e prossegue por toda a sua vida útil. Para Blaha e Rumbaugh (2006) este processo consiste em: concepção, considerando as necessidades do negócio; elicitação dos requisitos do sistema; projeto do sistema, estratégia de alto nível para atingir o objetivo da aplicação; projeto de classes, representações do mundo real a serem implantadas no computador;

elaboração do código fonte; testes; treinamento dos usuários; implantação; manutenção, viabilizando o sistema a longo prazo. Estas atividades requerem diferentes tipos de documentos.

Dois conceitos fundamentais de um *software* são a classe e o objeto. Para Sommerville (2007), classes são representações de elementos reais, constituídas de atributos e operações, servindo para a instanciação de objetos na memória do computador. Uma classe ‘contato’ poderia ter atributos ‘nome’ e ‘telefone’ e uma operação ‘adicionar’. Para o autor supracitado (2007) sistemas orientados a objetos são mais fáceis de alterar devido à independência desses objetos, possibilitando reutilizá-los em outros projetos.

Para auxiliar a programação, foi criado um padrão de documentação orientado a objetos, abrangendo vários níveis de abstração. Segundo Marshall (2000), a *Unified Modeling Language* (UML), Linguagem de Modelagem Unificada, é composta de diagramas que explicam a estrutura e execução dos *softwares*. Para Blaha e Rumbaugh (2006), esta linguagem inclui documentos como os diagramas: de casos de uso, definindo os modos de interação de atores com o sistema; de classes, definindo a estrutura dos objetos e seus relacionamentos; de estados, descrevendo a sequência das operações; de interações, mostrando como os objetos interagem criando os comportamentos requeridos do sistema.

Para Parnas (2008), documentar é laborioso, mas imprescindível para criar *softwares* de boa qualidade. Os registros servem também como uma memória do projeto. Segundo Garousi et al. (2015), documentar é necessário para a manutenção do sistema, visto que o profissional que efetua alterações pode não ser quem o produziu, sendo que tentar compreender o código sem a documentação pode consumir muito tempo e esforço. Para Lethbridge, Singer e Forward (2003) programadores preferem documentações simples e significativas, que possam ser facilmente atualizadas. Há ainda outros requisitos que a documentação deve seguir. Segundo Parnas (2008), os fatores de usabilidade da documentação são: ser concisa e clara; abstrair detalhes desnecessários; seguir regras precisas de arquivamento, para que o profissional saiba onde encontrar o que precisa.

Embora crucial, a documentação é comumente negligenciada, sendo inconsistente, incompleta, e desatualizada (BRIAND, 2003; LETHBRIDGE; SINGER; FORWARD, 2003; PARNAS, 2008, GAROUSI et al., 2015). O enfoque maior nas empresas de *software* é a produtividade e a documentação, em certo sentido, pode não ser priorizada.

Segundo Dingsoyr et al. (2012), em 2001, surgiu o Manifesto ágil, tendo como princípios: rapidez no desenvolvimento, flexibilidade para acomodar mudanças no projeto, trabalho em equipe, participação do cliente no processo e minimização de tarefas desnecessárias. No *site* oficial deste manifesto é afirmado ser preferível *software* funcionando que uma documentação abrangente<sup>1</sup>.

Contudo, o atingimento dos objetivos do projeto e a sinergia da equipe de desenvolvimento dependem também do registro e compartilhamento de informações e conhecimentos. Para Forward (2002) e Boer e Vliet (2009), a documentação é o registro do conhecimento produzido durante o projeto, sendo sua principal função a de comunicá-lo aos envolvidos. O registro é necessário, pois, conforme Garousi et al. (2015), embora muita informação possa ser transmitida pessoalmente, a memória humana é transitória, e esta maneira informal pode não ser eficaz para a comunicação com toda a equipe.

Nonaka e Takeuchi (1997) classificam o conhecimento em tácito, adquirido com a experiência, difícil de expressar, e explícito, estruturado e mais facilmente compartilhado, sendo que a criação de novo conhecimento depende da interação entre estes dois elementos. Para estes autores (1997), a documentação auxilia os indivíduos a internalizarem o conhecimento explícito, permitindo vivenciar indiretamente a experiência registrada, criando seu próprio saber tácito. Saberes tácitos e explícitos configuram distintos fluxos informacionais. Para Valentim (2010) a organização se vale de fluxos informacionais para propiciar a interação entre os indivíduos e efetivar suas ações, fluxos estes tanto estruturados, documentados, quanto não-estruturados, não registrados.

Para a autora supracitada (2010), os fluxos não-estruturados resultam da aprendizagem organizacional. Verifica-se que os fluxos estruturados contribuem para a aprendizagem organizacional, e esta contribui para os fluxos não-estruturados. Para Yeung et al. (1999), a aprendizagem organizacional evita a perda de conhecimentos, disseminando ideias além de fronteiras de espaço e tempo, o que envolve seu registro. A documentação é relevante para o aprendizado organizacional e formação de uma memória, visando o futuro.

### 3 PROCEDIMENTOS METODOLÓGICOS

A presente pesquisa descritiva-exploratória, caracteriza-se como um estudo de caso, com abordagem quali-quantitativa. Para a pesquisa foi selecionada uma empresa de *software* do

---

<sup>1</sup> Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 27 jun. 2018.

município de Marília, São Paulo, membro da Associação de Empresas de Serviços de Tecnologia da Informação (ASSERTI)<sup>2</sup>. O município referido destaca-se pelo seu Arranjo Produtivo Local de Tecnologia da Informação (APL TI)<sup>3</sup>, reconhecido pelo Governo estadual. A empresa selecionada, denominada doravante ‘empresa A’, atua há quase duas décadas, tendo noventa colaboradores, dos quais vinte e seis atuam com a elaboração de *software*.

Para verificar a relação dos profissionais de desenvolvimento de *software* com a informação, o conhecimento e a documentação, foram elaborados questionários com a escala de likert. Foi também realizada uma entrevista com o diretor de desenvolvimento, que coordena a programação, verificando os valores organizacionais que influenciam na prática de documentação. Foi efetuada ainda uma observação do ambiente, verificando como suas configurações contribuem para o compartilhamento de informações. A coleta de dados deu-se de maio a junho de 2018. A análise dos questionários foi quantitativa, computando as porcentagens das respostas. A entrevista foi analisada qualitativamente, extraindo elementos que pudessem esclarecer as tendências observadas nos questionários.

#### 4 RESULTADOS E DISCUSSÃO

Inicialmente, será examinado o perfil dos respondentes do questionário. Do total de 26 profissionais de desenvolvimento de *software* da empresa A, 07 ou 26,9%, responderam aos questionários. Quanto ao tempo de empresa: 03 respondentes trabalham nela há menos de um ano; 01 respondente de três a cinco anos; 02 respondentes de sete a nove anos; 01 respondente de dez a quatorze anos. Quanto aos cargos, são 05 programadores, 01 supervisor de desenvolvimento e 01 consultor de Engenharia de *Software*.

Quanto à aquisição da informação, na afirmação “adquiro informações para o meu trabalho principalmente junto à documentação de *software*”: 01 pessoa, 14,3%, concordou plenamente; 05 pessoas, 71,4%, concordaram parcialmente; 01 pessoa, 14,4%, discordou plenamente. Evidencia-se que a documentação é uma importante fonte informacional, ainda que não seja a mais utilizada.

Todavia, documentar pode ser visto como uma atividade pouco eficiente. Na afirmativa “é preferível me concentrar em implementar o *software* do que gastar tempo elaborando documentação”: 02 pessoas, ou 28,6%, concordam plenamente; 02 pessoas

---

<sup>2</sup> Disponível em: <<http://asserti.org/>>. Acesso em: 26 jun. 2018.

<sup>3</sup> Disponível em: <<http://www.apltimarilia.org.br/historia-do-apl-ti-marilia-arranjo-produtivo-local-de-tecnologia-da-informacao-de-marilia/>>. Acesso em: 26 jun. 2018.

concordam parcialmente; 03 pessoas, ou 42,9% discordam parcialmente; nenhuma discordou plenamente. A soma dos que de algum modo concordam com a afirmação perfaz 57,2%, sobrepujando os discordantes. Estes dados são exibidos no Gráfico 1.

**Gráfico 1: elaboração de documentação versus produtividade**

É preferível me concentrar em implementar o software do que gastar tempo elaborando documentação.



**Fonte: elaborado pelos autores (2018).**

Contudo, dar menor enfoque à documentação pode impactar na manutenção do sistema. Na sentença “normalmente, quando tenho que trabalhar com o código que outra pessoa escreveu, fico perdido no começo, porque tenho que analisar os métodos para descobrir exatamente como eles funcionam”, 03 indivíduos, 42,9% concordaram plenamente, outros 03 concordaram parcialmente, e apenas uma pessoa, 14,3% afirmou que não concorda nem discorda. Ninguém discordou plena ou parcialmente. Segundo Parnas (2008), embora muitos profissionais defendam que o código fonte possui toda a informação necessária, a documentação é essencial para a compreensão deste código. Para Garousi et al. (2015), tentar entender o código sem a documentação demanda muito tempo e esforço.

Na entrevista com o diretor de desenvolvimento, verificou-se que se reconhece a importância da documentação, mas que não há uma cobrança acentuada para sua elaboração. Verificou-se que é priorizada a produtividade, sendo utilizada uma metodologia que segue o Manifesto ágil. Segundo o diretor, há duas equipes, uma que dá manutenção em um sistema que é o principal produto da empresa, e outra, que desenvolve produtos novos, sendo que esta última gera mais documentos. A documentação concentra-se mais em especificações de negócio e registro de correções de erros. O diretor observa que há um grupo de programadores que elabora documentação mais rigorosamente que os demais.

Os resultados corroboram a conclusão de Lethbridge, Singer e Forward (2003) que afirmam que os profissionais dão maior atenção à documentação de testes e controle de qualidade, que é útil para executar tarefas mais imediatas. Estes autores (2003) sugerem

que se deve elaborar documentação simples e significativa, sendo desnecessário atualizar frequentemente documentação de alto nível de abstração, que continua útil, mesmo com alguns detalhes defasados. Para Forward (2002) criar documentação ‘boa o suficiente’ implica equilibrar forças favoráveis e desfavoráveis à documentação, considerando futuros mantenedores do código, restrições orçamentárias, de tempo, entre outros fatores.

Contudo, apurou-se que a empresa enfatiza os fluxos não-estruturados de informação e o compartilhamento de conhecimento tácito, com espaços sem divisórias e um local de descanso, com máquina de café e jogos, utilizado para o intercâmbio de ideias. É premente elaborar registros simples e significativos, para equilibrar os fluxos informacionais estruturados e não-estruturados, contribuindo para o aprendizado organizacional e internalização de conhecimentos. Para Schein (2007), em empresas ‘jovens’, em que os valores do fundador têm forte influência, a mudança ocorre por não se opor a estes valores e utilizar indivíduos que tenham as características comportamentais desejadas como seus catalisadores. Sugere-se que o grupo de profissionais identificado como os mais diligentes na documentação possa ser utilizado para transmitir esta cultura aos demais.

## 5 CONSIDERAÇÕES FINAIS

Manejar tempo e custos é essencial para desenvolver *softwares* rentáveis. Quanto ao problema de pesquisa proposto, ‘como se configuram as reais práticas de documentação na rotina das empresas de *software*?’, verificou-se que as práticas de documentação formal são inconsistentes, enfatizando-se fluxos informacionais não estruturados. Como resposta ao segundo problema, ‘como o enfoque na rapidez e produtividade afeta a consistência da documentação?’, comprovou-se que a ênfase exacerbada na rapidez e na produtividade pode interferir na elaboração e atualização de documentação, impactando na disponibilidade de conhecimentos explícitos acerca dos projetos e na manutenção dos produtos. Defende-se uma documentação concisa e significativa que, mesmo com as restrições orçamentárias e de tempo, permita o compartilhamento de informações e conhecimento explícito, tornando eficiente o desenvolvimento e manutenção de *software*.

## REFERÊNCIAS

BLAHA, M.; RUMBAUGH, J. **Modelagem e projetos baseados em objetos com UML 2**. 2 ed. Rio de Janeiro: Elsevier: 2006.

- BOER, R. C.; VLIET, H. Writing and reading software documentation: how the development process may affect understanding. In: ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009, Vancouver. **Proceedings...** Whashington: IEEE, 2009. p. 40- 47. Disponível em: <<https://doi.org/10.1109/CHASE.2009.5071409>>. Acesso em: 19 jun. 2018.
- BRIAND, L. C. Software Documentation: how much is enough? In: European Conference on Software Maintenance and Reengineering, 17, 2003, Benevento. **Proceedings...** Washington: IEEE Computer Society, 2003. Disponível em: <<https://doi.org/10.1109/CSMR.2003.1192406>>. Acesso em: 27 jun. 2018.
- DINGSOYR, T. et al. A decade of agile methodologies: towards explaining agile software development. **The Journal of Systems and Software**, v. 85, p. 1213– 1221, 2012. Disponível em: <<https://doi.org/10.1016/j.jss.2012.02.033>>. Acesso em: 22 jun. 2018.
- FORWARD, A. **Software documentation**: mantaining artefacts of communication. 2002. 172 f. Dissertação (Mestrado em Ciência da Computação) – Institute for Computer Science, University of Ottawa, Ottawa, 2002. Disponível em: <[http://www.site.uottawa.ca/~tcl/gradtheses/forward/forward\\_thesis.pdf](http://www.site.uottawa.ca/~tcl/gradtheses/forward/forward_thesis.pdf)>. Acesso em: 19 jun. 2018.
- GAROUSI, G. et al. Usage and usefulness of technical software documentation: an industrial case study. **Information and Software Technology**, v. 57, p. 664-682, jan. 2015. Disponível em: <<https://doi.org/10.1016/j.infsof.2014.08.003>>. Acesso em: 19 jun. 2018.
- LETHBRIDGE, T.C.; SINGER, J.; FORWARD, A. How software engineers use documentation: the state of the practice. **IEEE Software**, v. 20, n. 6, nov./dez. 2003, p. 35 - 39. Disponível em: <<https://doi.org/10.1109/MS.2003.1241364>>. Acesso em: 19 jun. 2018.
- MARSHALL, C. **Enterprise modeling with UML**: designing successful software through business analysis. Reading (MA): Addison Wesley Longman, 2000.
- NONAKA, I.; TAKEUCHI, H. **Criação de conhecimento na empresa**: como as empresas japonesas geram a dinâmica da inovação. Rio de Janeiro: Elsevier, 1997.
- PARNAS, D. L. Document based rational software development. **Knowledge-Based Systems**, v. 22, n. 3, p. 132-141, abr. 2009. Disponível em: <<https://doi.org/10.1016/j.knosys.2008.11.001>>. Acesso em: 19 jun. 2018.
- SCHEIN, E. H. **Guia de sobrevivência da cultura corporativa**. 2 ed. Rio de Janeiro: José Olímpio, 2007.
- SOMMERVILLE, I. **Engenharia de software**. 8.ed. São Paulo: Pearson Education, 2007.
- VALENTIM, M. L. P. **Ambientes e fluxos de informação**. In: \_\_\_\_\_ (Org.). **Ambientes e fluxos de informação**. São Paulo: Cultura Acadêmica, 2010. p. 13- 22.
- YEUNG, A. K. et al. **Organizational learning capability**. Nova York: Oxford University Press, 1999.