

# Ações de Transparência no Processo de Desenvolvimento de Software no Setor Público

Fabíola Ferreira de Macedo<sup>1,2</sup>, Patrícia Vilain<sup>1</sup>

<sup>1</sup>Universidade Federal de Santa Catarina

<sup>2</sup>Assembleia Legislativa do Estado de Santa Catarina

ffm6325@alesc.sc.gov.br

vilain@inf.ufsc.br

**Abstract.** Impulsionadas pela sociedade e por legislações mais rigorosas, cada vez mais as organizações públicas necessitam demonstrar transparência. Após a promulgação da Lei de Acesso à Informação (LAI), que regulamenta a transparência pública, o desafio está na aplicação desta lei nos processos e informações dos órgãos governamentais, incluindo o processo de desenvolvimento de software. Para isso, são necessários esforços em resolver questões de como cada software irá lidar com os requisitos de transparência ao longo do seu processo de desenvolvimento. Este artigo apresenta como órgãos públicos, que terceirizam algumas etapas do desenvolvimento de software, podem realizar ações de transparência durante o processo de desenvolvimento de software apoiando a implementação de requisitos de transparência.

**Keywords:** Transparência de Software, Transparência Pública, Ações de Transparência, Terceirização de Software, Processo de Software.

## 1 Introdução

As sociedades, cada vez mais exigentes quanto aos atos da administração pública, estimulam iniciativas por mais transparência nos governos. No Brasil, importantes esforços na busca por mais transparência foram traduzidos em leis. A Lei da Transparência (BRASIL, 2009) dispõe sobre a disponibilização, em tempo real, de informações. E, mais recentemente, a Lei de Acesso a Informações – LAI (BRASIL, 2011a) regula o acesso a informações e dispõe sobre os procedimentos a serem observados pela União, Estados, Distrito Federal e Municípios, com o fim de garantir o acesso a informações.

A LAI representa um novo passo na busca pela transparência no país, pois desde a aprovação dessa lei “qualquer cidadão pode solicitar acesso às informações públicas, e não classificadas como sigilosas, conforme procedimentos, prazos e recursos previstos” (BRASIL, 2011a). A LAI também prevê, em seu artigo 6º, que cabe aos órgãos públicos assegurar a gestão transparente, promovendo o amplo acesso e divulgação da informação e garantir a proteção da informação, observando a sua disponibilidade, autenticidade, integridade e eventuais restrições de acesso (BRASIL, 2011a). Essa

nova demanda impõe às instituições públicas a necessidade de estabelecer mecanismos que demonstrem a transparência de suas informações e processos (LEITE e CAPPELLI, 2010), o que inclui, inevitavelmente, o processo de desenvolvimento de software.

A transparência de software relacionada ao processo de desenvolvimento foi definida como um atributo de comunicação que permite aos *stakeholders* responder questões sobre o sistema de software durante o seu ciclo de vida (YU-CHENG et al., 2011). Segundo essa definição, a falta de transparência é um importante fator que contribui para problemas de comunicação que levam a falhas no desenvolvimento de software.

Alguns estudos sugerem, como forma de melhorar a comunicação entre os *stakeholders*, o compartilhamento de dados empíricos do projeto de desenvolvimento de software como, por exemplo, documentos de requisitos, documentos de projeto de software, casos de teste, etc., o que também promoveria a transparência do desenvolvimento de software (TSUNODA et al., 2010). Nesse sentido, modelos que facilitem a comunicação desde a análise de requisitos são de extrema relevância e devem ser utilizados (XAVIER, 2010).

Nesse cenário, pretende-se com essa pesquisa descobrir maneiras de definir, descrever e avaliar os requisitos de transparência de software e identificar em que momento as ações para transparência podem ser realizadas ao longo do processo de desenvolvimento de software.

O artigo está organizado da seguinte maneira, a seção 2 apresenta os objetivos da pesquisa que está sendo desenvolvida para tratar das ações de transparência durante o processo de desenvolvimento de software. A seção 3 mostra as contribuições esperadas para a área de desenvolvimento transparente de software. A seção 4 apresenta os resultados já alcançados por esta pesquisa. E, por fim, a seção 5 apresenta as conclusões.

## **2 Objetivos da Pesquisa**

Nessa pesquisa considera-se que os requisitos de transparência em um sistema de software podem ser especificados por meio de requisitos não-funcionais, como acessibilidade, usabilidade, informatividade, entendimento e auditabilidade (CAPPELLI, 2009; SERRANO e LEITE, 2011). Para mapear essas características não-funcionais em ações, que indicariam como de fato elas poderiam ser concretizadas no desenvolvimento (XAVIER et al., 2010), torna-se necessário saber quais ações as instituições públicas devem realizar durante o processo de desenvolvimento de software.

Sendo assim, as ações para transparência consideradas nessa pesquisa são aquelas apresentadas pelo *checklist* desenvolvido a partir de um projeto de pesquisa iniciado em 2012 (MACEDO, 2012) e em processo de evolução desde então, e que está sendo aplicado em um projeto na Assembleia Legislativa do Estado de Santa Catarina (ALESC). O *checklist* de ações para transparência foi proposto considerando-se a legislação relacionada à transparência pública (BRASIL, 1998; BRASIL, 2009; BRASIL, 2011a), o Catálogo de Transparência de Software apresentado pelo Grupo

ER – PUC-Rio (ER – PUC-Rio, 2013), e as iniciativas para concretizar características de transparência do Governo Federal, como o Manual dos Dados Abertos (W3C, 2010), a Cartilha de Usabilidade (BRASIL, 2010) e o Modelo de Acessibilidade (BRASIL, 2011b).

O *checklist* de ações consiste em uma adaptação do Catálogo de Transparência proposto pelo Grupo ER – PUC-Rio, voltado a atender as necessidades do setor público, por meio de ações que podem ajudar na transparência do desenvolvimento de software, mesmo quando parte dele é terceirizada. Entretanto, além do uso desse *checklist*, verifica-se a necessidade de encontrar maneiras de definir e avaliar o atendimento a transparência, além de identificar a ocasião em que as ações de transparência, propostas pelo *checklist*, podem ser realizadas ao longo do processo de desenvolvimento de software.

Sendo assim, o objetivo geral dessa pesquisa é (i) definir em que momento do processo de desenvolvimento de software os requisitos de transparência, identificados a partir das ações de transparência apresentadas pelo *checklist*, devem ser definidos e validados; (ii) sugerir uma forma de como o atendimento à transparência pode ser avaliado.

### **3 Contribuições Esperadas**

A falta de entendimento dos objetivos do sistema e a deficiência na comunicação podem levar ao fracasso do projeto (XAVIER, 2010). Portanto, como uma primeira contribuição, pretende-se determinar quando os requisitos de transparência, identificados a partir das ações de transparência sugeridas pelo *checklist*, devem ser definidos e validados durante o processo de desenvolvimento de software. Isso ajudará a esclareceras verdadeiras necessidades do projeto em relação à transparência e demais requisitos.

Para alcançar a transparência, num sentido completo, todas as características de transparência (acessibilidade, usabilidade, informatividade, entendimento e auditabilidade) devem estar presentes na organização (ARAÚJO et al., 2010). Porém, como estas são características de qualidade, podem existir diferentes níveis de entendimento e percepção dependendo do interessado (LEITE e CAPPELLI, 2010). Essa questão se configura numa segunda contribuição esperada para essa pesquisa, ou seja, propor uma forma de como o atendimento à transparência pode ser avaliado considerando as necessidades de cada instituição.

### **4 Resultados Alcançados**

Nas instituições públicas, devido à alta demanda por sistemas de informação, algumas etapas do desenvolvimento de software são terceirizadas. Normalmente, nesse processo as atividades que vão conduzir a aquisição de um software pela organização pública (contratante) têm início desde o estabelecimento das necessidades (requisitos), passando pela elaboração do contrato, indo até a aceitação do sistema entregue (SOFTEX, 2011). Neste cenário, como primeiro resultado esperado, propõe-se que os

requisitos de transparência, identificados por meio do *checklist*, sejam definidos na fase de levantamento de requisitos e descritos na forma de testes de aceitação.

Os testes de aceitação podem ser definidos como testes escritos pela colaboração entre *stakeholders* e programadores a fim de definir quando um requisito é realizado (MARTIN, 2011). Tradicionalmente, eles são executados na etapa final de testes, antes da liberação para o cliente. Entretanto, nem sempre é conveniente esperar tanto tempo para defini-los, recomendando-se especificar os testes de aceitação antes da implementação começar (CONNOLLY, 2009). Essa recomendação, de especificar os testes de aceitação o quanto antes, está incluída nesta proposta de pesquisa e está sendo aplicada em um projeto realizado na ALESC. Entendeu-se que a necessidade de delegar etapas de desenvolvimento a empresas externas exige uma representação mais completa dos requisitos desejados em estágios iniciais do desenvolvimento, para que estes possam ser incluídos nos contratos (editais) licitatórios e exigidos na entrega do sistema de software.

Nesta pesquisa, os testes de aceitação são especificados da seguinte forma, os requisitos, incluindo os de transparência, são descritos em “critérios de aceitação” pelos *stakeholders*, e incluídos no contrato licitatório (edital). A empresa terceirizada deve implementar um sistema que atenda a todas as exigências contidas nos “critérios de aceitação”, pois eles serão usados na validação do sistema entregue. A tabela 1 apresenta alguns exemplos de “critérios de aceitação” especificados no projeto em desenvolvimento na ALESC para a funcionalidade “Listar Licença - Prêmio”, representada através de uma estória de usuário. As ações de transparência sugeridas a partir do *checklist* e selecionadas para essa estória do usuário estão grifadas.

**Tabela 1.** Exemplos de Critérios de Aceitação

<b>Estória de Usuário 01: Listar Licença-Prêmio</b> Usuário insere um determinado período e o sistema lista o nome e a matrícula dos servidores que tem direito ao benefício <b>licença-prêmio</b> baseado nos critérios da Lei 6.745.
<b>Critérios para Aceitação:</b> <ul style="list-style-type: none"><li>- O usuário deve obrigatoriamente informar o período. Caso o campo estiver em branco, o sistema deve mostrar uma mensagem de alerta.</li><li>- Quando o sistema não encontrar servidores que atendam aos critérios, deve informar que nenhum servidor foi encontrado.</li><li>- <u>Deve ser possível salvar as informações em formato aberto (RDF).</u></li><li>- <u>Deve ser possível executar a tarefa completa em navegadores IExplore e Chrome.</u></li><li>- <u>O sistema deve apresentar informações sobre o conjunto de dados (Catálogo de dados).</u></li></ul>

Nesse exemplo, observa-se que foram incluídos, além dos requisitos de sistema, alguns requisitos e ações para alcance da transparência, como a possibilidade de salvar as informações em formato aberto, a definição das diferentes plataformas atualmente utilizadas com que o software deve ser compatível e a necessidade de apresentar informações sobre o conjunto de dados.

Assim, considerando as quatro atividades fundamentais para a engenharia de software: especificação, projeto e implementação, validação e evolução (SOMMERVILLE, 2011), propõe-se que as ações de transparência constantes no

*checklist* devem ser selecionadas pela empresa contratante nas atividades de **especificação** e descritas em testes de aceitação, por meio de critérios de aceitação. As ações constantes nos critérios de aceitação serão realizadas pela empresa terceirizada no **projeto e implementação** de software. E, finalmente, os critérios de aceitação serão averiguados na **validação** de software pela empresa pública contratante. Porém, ainda não foi detalhado como estas ações serão realizadas dentro destas atividades.

Como segundo resultado dessa pesquisa, propõe-se o uso dos critérios de aceitação para realizar a conferência do percentual de ações atendidas. Sendo assim, a própria instituição pode determinar um percentual mínimo de ações concretizadas para considerar o software transparente. No projeto desenvolvido na ALESC, por exemplo, pretende-se estabelecer um mínimo de 90% (noventa por cento) de realização (em relação aos critérios de aceitação como, por exemplo, aqueles grifados na tabela 1).

Alguns trabalhos pesquisados apontam para o uso de modelo de níveis de transparência, usando estágios (CAPPELLI e LEITE, 2008; CAPPELLI, 2009; ARAÚJO et al., 2010). Também o “Projeto de Construção do Modelo de Maturidade em Transparência Organizacional”, da Universidade Federal do Rio de Janeiro (UNIRIO, 2013), sugere o uso de um modelo de níveis para avaliar o atendimento a transparência. Entretanto, apesar de ainda não ter sido analisado como será a validação de um sistema quando um modelo de níveis de transparência é usado, acredita-se que os critérios de aceitação também sejam adequados neste tipo de abordagem.

## 5 Conclusão

As sociedades cada vez mais exigentes e a regulamentação da transparência pública geraram uma nova demanda às instituições governamentais. Para garantir a aplicação desta Lei de Acesso à Informação (LAI) nos processos e informações dos órgãos governamentais, são necessários esforços para resolver questões de como software irá lidar com os requisitos de transparência ao longo do processo de desenvolvimento.

O objetivo dessa pesquisa foi esclarecer em que momento do desenvolvimento de software as instituições públicas, que terceirizam parte desse desenvolvimento, podem definir e validar os requisitos de transparência desejados. E como o atendimento a essa transparência pode ser avaliado, contribuindo assim para a garantia da transparência necessária. Essa proposta de pesquisa está sendo aplicada no projeto de Recursos Humanos (módulo de benefícios) na ALESC, em que as etapas de especificação de requisitos e validação serão realizadas pela instituição pública e as etapas de projeto e implementação serão delegadas a uma empresa contratada.

## Referências

1. ARAÚJO, R.M.,CAPPELLI, C. A., LEITE, J.C.S.P.. A importância de um Modelo de Estágios para avaliar Transparência. Revista TCMRJ, setembro, 2010. Número: 45, p. 97.
2. BRASIL. Constituição (1988). Constituição da República Federativa do Brasil: promulgada em 5 de outubro de 1988. Brasília, DF, Senado, 2004.

3. BRASIL. Lei n. 131, de 27 de maio de 2009. Dispõe sobre disponibilização em tempo real de informações (Lei da Transparência).
4. BRASIL. Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação. Padrões Brasil e-Gov: Cartilha de Usabilidade. - Brasília: MP, SLTI, 2010. 50 p.
5. BRASIL. Lei n. 12.527, de 28 de novembro de 2011. Regula o acesso a informações. 2011<sup>a</sup> (Lei de Acesso à Informação).
6. BRASIL. Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação. e-MAG Modelo de Acessibilidade em Governo Eletrônico. Ministério da Educação, - Brasília : MP, SLTI, 2011b.
7. CAPPELLI, C., LEITE, J. C. S. P. Transparência de Processos Organizacionais. Universidade Federal Fluminense, LATEC. II Simp. Inter. de Transparência nos Negócios. 2008.
8. CAPELLI, C. A.. Uma Abordagem para Transparência em Processos Organizacionais Utilizando Aspectos. 2009. 328p. Tese de Doutorado, Departamento de Informática, PUC-Rio, Rio Janeiro. 2009.
9. CONNOLLY, D.; KEENAN, F.; MCCAFFERY, F. . Developing acceptance tests from existing documentation using annotations: An experiment. In Automation of Software Test, 2009. AST '09. ICSE Workshop on (18-19 2009), pp. 123 –129.
10. ER – PUC-RIO. Grupo de Pesquisas em Engenharia de Requisitos da PUC-RIO. <Disponível em: <http://www.er.les.inf.puc-rio.br/~wiki/index.php/>><Acesso em: 16/07/2013>
11. MACEDO, F. F. de; VILAIN, P.. Transparência no Processo de Software como Apoio à Informação da Administração Pública. In.: VII Simpósio Brasileiro de Sistemas de Informação (SBSI). 2012.São Paulo- SP, Brasil, pp 303-323.
12. LEITE, J.C.S.P., CAPPELLI, C. A. Software Transparency. Business & Information Systems Engineering, Springer, 01 jun. 2010. Volume: 2, Número: 3, p. 127-139.
13. MARTIN, R.C..The clean coder: A Code of Conduct for Professional Programmers.Prentice Hall.2011. ISBN 0-13-708107-3.
14. YU-CHENG, T.; THOMBORSON, C.; TEMPERO, E. Illusions and Perceptions of Transparency in Software Engineering. Software Engineering Conference (APSEC), 2011, 18th Asia Pacific, vol., no., pp.365-372, 5-8 Dec. 2011.doi: 10.1109/APSEC.2011.42.
15. SERRANO, M.; LEITE, J.C.S.P. Capturing Transparency-Related Requirements Patterns through Argumentation. Requirements Patterns (RePa).First International Workshop on, p.32-41, Aug. 2011. DOI: 10.1109/RePa.2011.6046723 2011.
16. SOFTEX, ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO - SOFTEX. MPS.BR - Guia de Aquisição:2011, junho 2011.
17. SOMMERVILLE, I..Engenharia de Software. 9ª ed. S.P.: Pearson Prentice Hall, 2011.
18. TSUNODA, M.; et al. . Standardizing the Software Tag in Japan for Transparency of Development. Book Title: Product-Focused Software Process Improvement. Book Series Title: Lecture Notes in Computer Science. 2010. Publisher: Springer Berlin / Heidelberg.
19. UNIRIO. Modelo de Maturidade em Transparência Organizacional. <Disponível em: [http://www2.unirio.br/unirio/ccet/unirio-cgu. /](http://www2.unirio.br/unirio/ccet/unirio-cgu./)><Acesso em: 20/07/2013>.
20. XAVIER, L.; ALENCAR, F.; CASTRO, J.; PIMENTEL, J.. Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN and NFR. In: 13th Workshop em Engenharia de Requisitos (WER), Cuenca, Ecuador, p. 29-40.2010.
21. W3C. Brasil. Manual dos dados abertos: governo. 2010. (Trad. Adapt. openmanualdata.org). Disponível em:< <http://www.w3c.br/>>. Acessado em: 10/07/2013.