

Especializando Ações de Transparência para Qualidade no Desenvolvimento de Software no Setor Público

Trabalho Técnico

Fabíola Ferreira de Macedo^{1,2}, Patrícia Vilain²

¹ Assembleia Legislativa do Estado de Santa Catarina (ALESC)
Florianópolis – SC – Brazil

²Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

fabimacedo@alesc.sc.gov.br, vilain@inf.ufsc.br

Abstract. *The desire of the current society for more transparency motivated public companies the need to establish mechanisms that evidence the transparency of information and processes. This article presents definitions for transparency and software transparency, and discusses how the inclusion of quality requirements during the requirements specification can reflect on improving transparency in the public sector. We propose the specialization of a checklist with 51 actions for transparency (checkTrans). As a result, we suggest 19 actions as essential for transparency that were evaluated considering the attending quality requirements.*

Resumo. *O anseio das sociedades atuais por mais transparência gerou às instituições públicas a necessidade de estabelecer mecanismos que demonstrem transparência de suas informações e seus processos. Este artigo apresenta definições para transparência e transparência de software, e aborda como a inclusão de requisitos de qualidade durante a especificação de requisitos pode refletir na melhoria da transparência no setor público. Propõe-se a especialização de um checklist com 51 ações voltadas a transparência (checkTrans). Como resultado sugere-se 19 ações essenciais para transparência que foram avaliadas considerando-se o atendimento a requisitos de qualidade.*

1. Introdução

Atualmente a transparência tem sido enfatizada nas organizações, principalmente nos governos e grandes corporações [Yu-Cheng et al. 2011]. Com o aumento da atenção sobre as questões da transparência na política moderna [Open Forum Foundation 2011], observou-se o surgimento de novas leis no sentido de aprimorar a transparência e a eficiência dos atos da administração pública. No Brasil, a legislação também passou a exigir maior transparência de suas instituições públicas, a exemplo da Lei de Acesso à Informação (LAI).

A LAI gerou nas instituições públicas a necessidade de estabelecer mecanismos que demonstrem a transparência de suas informações e seus processos. Isso requer enfrentar questões de natureza administrativa, técnica e tecnológica, a fim de assegurar a observância do que dispõe esta lei [Brasil 2011a]. Nesse cenário observa-se algumas importantes iniciativas que visam implementar a transparência nas organizações como, por exemplo, documentos do governo federal [W3C 2010], [Brasil 2010]; [Brasil 2011b], o catálogo de transparência [Cappelli 2009], [ER-PUC-Rio 2014] e o *checklist* de ações para transparência (checkTrans) [Macedo 2014].

Destaca-se nesse artigo as ações para transparência apresentadas pelo checkTrans. Ele apresenta como instituições públicas, que terceirizam algumas etapas do desenvolvimento de software, podem executar atividades durante o processo de desenvolvimento de software com o objetivo de atender aos requisitos de transparência exigidos pela legislação. Entretanto, para garantir sua utilização em instituições que estão dando os primeiros passos em busca da transparência, a implementação dos requisitos de transparência deve ser ágil. Portanto, há necessidade de diminuir o checkTrans, priorizando as ações essenciais.

O artigo está organizado da seguinte forma. A seção 2 apresenta os conceitos relacionados com transparência. A seção 3 apresenta o checkTrans, um checklist de ações para transparência. A seção 4 comenta sobre os trabalhos relacionados. Na seção 5 é proposta a priorização das ações do checkTrans, resultando em um checklist reduzido contendo somente ações essenciais. Para avaliar a efetividade deste novo checklist proposto, a seção 6 detalha os resultados obtidos com o seu uso em um estudo de caso. Por fim, a seção 7 apresenta as considerações finais.

2. Transparência

Em alguns domínios, a transparência é definida como “fácil de perceber ou detectar”, em outros, tem um sentido oposto, como “capacidade de funcionar sem que o usuário perceba sua presença” [Oxford 2012]. As definições compatíveis com o foco deste artigo estão relacionadas de acordo com o primeiro significado: “fácil de perceber ou detectar”.

2.1. Transparência de software

Quando aplicada ao software, a transparência pode “incentivar a disponibilização de informações mais completas, objetivas, confiáveis e de qualidade...” [Leal et al. 2011]. Um software é considerado transparente se ele atende a duas condições: transparência de informação (se o software trata com transparência as informações que ele lida); e transparência de processo (se o software informa seu funcionamento, o que faz, porque faz) [Leite e Cappelli 2008].

A transparência de software pode ser tratada como um requisito de qualidade, portanto, como um requisito não funcional (RNF). Assim, para garantir a transparência de software, deve-se lidar com ela já no contexto de especificação de requisitos [Leite e Cappelli 2010]. Logo, a chave para a transparência de software é tornar os requisitos mais legíveis aos interessados em geral, bem como para os desenvolvedores [Leite e Cappelli 2008], [Cysneiros 2009]. Uma das questões a ser tratada dentro da

transparência de software é como garantir a transparência do desenvolvimento de software no setor público [Macedo e Vilain 2012].

2.2. Transparência no setor público

Num primeiro momento, as iniciativas por mais transparência na administração pública Brasileira ganharam força com a promulgação da Constituição Federal [Brasil 1988] que, em seu artigo 37 deliberou sobre a transparência na forma do princípio constitucional de publicidade.

Posteriormente, para atender a crescente demanda dos cidadãos pelo conhecimento dos atos administrativos, as iniciativas foram no sentido de divulgação da informação como, por exemplo, a Lei da Transparência [Brasil 2009], que dispõe sobre a disponibilização, em tempo real, de informações. E, mais recentemente, representando um novo passo na busca pela transparência no país, a Lei de Acesso a Informações (LAI) [Brasil 2011c], que “consolida e define o marco regulatório sobre o acesso à informação pública sob a guarda do Estado” [Brasil 2011a]. Para a Controladoria Geral da União, órgão do Governo Federal responsável pelo incremento da transparência da gestão pública, o desafio passa a ser a garantia da implementação da LAI.

Logo, essas questões atingem também o contexto do desenvolvimento de software. Assim, para implementar a transparência, a sociedade necessita abordar como o software lidará com este conceito [Leite e Cappelli 2010]. As instituições públicas, devido à alta demanda por sistemas de informação, frequentemente, terceirizam algumas etapas do desenvolvimento de software, o que pode aumentar a dificuldade em garantir os requisitos de transparência exigidos pela legislação, exigindo esforços administrativos e tecnológicos para o efetivo alcance dessa transparência.

Nesse cenário, a recorrente opção por terceirização de parte do desenvolvimento de software demanda ações a serem realizadas pelas empresas públicas contratantes ainda na fase de especificação de requisitos [Macedo 2014]. Uma ferramenta que visa apoiar a definição dos requisitos de transparência desejados pela instituição e a incluí-los nos contratos licitatórios é o *checklist* de transparência (checkTrans).

3. Checklist de Transparência - CheckTrans

O checkTrans é um *checklist* de ações para transparência que apresenta um conjunto de ações e exemplos que visam à definição de requisitos de transparência no desenvolvimento de software em empresas públicas [Macedo 2014]. O checkTrans é uma especialização do catálogo de transparência proposto pelo Grupo Engenharia de Software da PUC – Rio [ER – PUC - Rio 2014], e é composto por 51 ações voltadas a transparência que devem ser realizadas a partir da especificação de requisitos. Estas ações estão agrupadas de acordo com a característica de qualidade (transparência) a ser atendida: acessibilidade, usabilidade, informatividade, entendimento e auditabilidade, conforme detalhado em [Macedo 2014].

As ações sugeridas no checkTrans devem ser identificadas ainda na fase de especificação de requisitos, e podem auxiliar no acompanhamento da evolução do processo de desenvolvimento de software e na validação dos requisitos. A proposta associada ao checkTrans é que os requisitos de transparência identificados a partir das ações sugeridas sejam definidos na fase de especificação de requisitos e descritos na

forma de testes de aceitação. Essa recomendação de especificar os testes de aceitação antes de a implementação começar deve-se à necessidade de delegar etapas de desenvolvimento a empresas externas, uma vez que isso 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 [Macedo 2014].

Em suma, o checkTrans sugere que os testes de aceitação sejam especificados da seguinte forma: os requisitos, incluindo os de transparência, são descritos como “critérios de aceitação” pelos *stakeholders* e incluídos no contrato licitatório (edital). A empresa terceirizada deve, então, implementar um sistema que atenda a todas as exigências contidas nos “critérios de aceitação”, pois eles serão usados na validação do sistema entregue.

4. Trabalhos Relacionados

Das iniciativas que visam apoiar o aumento da transparência, destacam-se o Catálogo de Transparência [Cappelli 2009], a *wiki* do Grupo de Pesquisa de Engenharia de Requisitos da PUC-Rio [ER-PUC-Rio 2014], os documentos do Governo Federal [W3C 2010], [Brasil 2010], [Brasil 2011b] e o próprio checkTrans (apresentado na seção 3). Essas iniciativas podem ajudar a atender aos requisitos obrigatórios impostos pela legislação e a concretizar as características de transparência de acessibilidade, usabilidade, informativo, entendimento e auditabilidade.

O Catálogo de Transparência apresenta uma importante contribuição na forma de um catálogo de requisitos não funcionais (NFR). É uma estrutura que auxilia na organização das características de qualidade (acessibilidade, usabilidade, informativo, entendimento e auditabilidade), apresentando a priorização e as dependências entre essas características [Cappelli 2009]. Também apresenta as operacionalizações, que indicam como, de fato, os requisitos não funcionais deveriam ser concretizados, entretanto está mais voltado aos processos organizacionais.

A *wiki* do Grupo de Pesquisa de Engenharia de Requisitos da PUC-Rio dispõe também de uma estrutura baseada no Catálogo de Transparência. No Catálogo de Transparência disponibilizado pelo Grupo ER – PUC-Rio as características são complementadas com uma série de questões (algo em torno de 320 questões), as quais podem ser levantadas durante o desenvolvimento de um sistema de software, visando o atendimento de cada característica de transparência [ER – PUC-Rio 2014]. Porém, este catálogo não ultrapassa o nível das questões, ou seja, não chega ao nível das operacionalizações que são ações que podem ajudar na efetiva aplicação da transparência.

O Governo Federal sugere maneiras de melhorar a disponibilização da informação e comunicação por meio de documentos como, por exemplo, o Manual dos Dados Abertos [W3C 2010], a Cartilha de Usabilidade [Brasil 2010] e o Modelo de Acessibilidade [Brasil 2011b]. Estes documentos tratam mais da transparência na divulgação da informação.

5. Proposta

O checkTrans, apresentado na seção 3, elenca a execução de 51 ações durante o processo de especificação de requisitos. Porém, entende-se que poderá haver um aumento de custos no desenvolvimento com a utilização do checkTrans, isso porque as ações sugeridas, provavelmente, acarretarão em atividades extras a serem realizadas pela instituição e, nesse caso, aumentará o número de horas trabalhadas adicionando-se custos ao projeto.

Dessa forma, pensando-se na viabilidade dos projetos e no engajamento de instituições que estão dando os primeiros passos em direção à transparência, é proposta a priorização das ações do checkTrans. Para isso elencou-se as ações mínimas de transparência, ou seja, foram identificadas, dentre as 51 ações do checkTrans, quais são de extrema importância e não devem deixar de ser executadas. A seleção dessas ações mínimas foi estabelecida a partir dos seguintes critérios: (i) ações com elementos recorrentes em mais de uma questão do catálogo de transparência, e (ii) ações para atender requisitos explícitos na legislação.

No primeiro caso, o de ações para atender elementos recorrentes, percebe-se que os itens presentes em mais de uma questão do catálogo de transparência, que serviu como base para o checkTrans completo, indicam que eles têm uma maior importância, e, portanto, não poderiam deixar de ser observados. Na segunda situação, o de ações explícitas na legislação, entende-se que é de extrema importância atender aos requisitos de transparência exigidos explicitamente pela legislação vigente, e, portanto, devem ser preferencialmente realizadas. O resultado da priorização das ações do checkTrans resultou em 19 ações essenciais para transparência que foram organizadas também em um checklist denominado: checkTrans-E (checkTrans-Essencial). As ações do checkTrans-E são elencadas na Tabela 1 a seguir.

Tabela 1. CheckTrans-E: Ações essenciais para transparência

Atributos		Ação sugerida	Exemplos
Acessibilidade	Portabilidade	Especificar as diferentes plataformas atualmente utilizadas que devem ser compatíveis com o <i>software</i> .	- Hardware (PC's, tablets, smartphones). - <i>Software</i> (windows, MAC, Linux, android) [Google 2013].
		Definir o formato aberto a ser utilizado.	- Formatos reconhecíveis por máquinas (RDF, JSON e XML). - Disponibilização dos dados em <i>Web services</i> [W3C 2010].
		Especificar as diferentes tecnologias onde o <i>software</i> poderá ser utilizado.	- Navegadores Iexplorer e Chrome [Google 2013]. - Padrões de desenvolvimento W3C. [Brasil 2011b].

	Disponibilidade	Definir os recursos e técnicas a serem utilizados para garantia da disponibilidade dos <i>sites</i> .	- Links de dados com SLA's. - Servidor de Aplicação. - Servidor dedicado, replicação, balanceamento de carga. [Tanenbaum 2003].
		Definir as tecnologias de proteção da informação	- Algoritmo de Criptografia. [Tanenbaum 2003].
		Definir as facilidades de navegação, inclusive, para pessoas com deficiência.	- Acesso ao computador sem mouse. - Padrões de acessibilidade (e-MAG). [Brasil 2011b].
		Definir e classificar as informações sigilosas ou públicas.	- Política de sigilo de informação.
Informativo	Corretude	Definir um plano de testes adequado ao domínio e a cobertura desejada.	- Testes de Aceitação [Connolly 2009]. - Testes de sistema [Sommerville 2011].
		Definir e usar a representação da linguagem do ambiente a ser desenvolvido.	- Vocabulário mínimo [Brasil 2010]. - Técnica Léxico Ampliado da Linguagem. [C&L 2013].
	Clareza	Descrever a arquitetura do <i>software</i> (definições gerais da arquitetura).	- Diagrama de Blocos [Sommerville 2011]. - Diagrama de Pacotes [Silva 2007].
	Integridade	Garantir a integridade das informações disponíveis para acesso.	- Assinatura digital de documentos.
	Atualidade	Definir a política de atualização do <i>software</i> .	- Mecanismos de atualização <i>online</i> - Mecanismos de informação sobre a disponibilidade de novas versões (Metadados). - Especificação da nova versão. [ER – PUC-Rio 2014].
	Completeza	Identificar e descrever as necessidades (requisitos funcionais e não funcionais) dos diversos pontos de vista.	- Documentos diferenciados para cada grupo de interessados [ER – PUC-Rio 2014]. - Histórias do usuário, casos de uso, cenários [Sommerville 2011].
	Comparabilidade	Identificar, aplicar e avaliar critérios de comparação entre os valores de referência e o <i>software</i> apresentado.	- Valores de referência para avaliar qualidade. [ER – PUC-Rio 2014].

Usabilidade	Validade	Avaliar os artefatos de <i>software</i> produzidos no processo de requisitos.	- Inspeção [Sayao 2007]. - Consultar os interessados [ER – PUC-Rio 2014].
	Uniformidade	Padronizar ambientes.	Padronizar interfaces, rótulos, terminologias, hierarquia e menus de forma lógica e intuitiva. [Brasil 2010].
Usabilidade	Amigabilidade	Utilizar uma linguagem clara e usual.	Diretrizes de redação, esquema consistente de cores e fontes [Brasil 2010].
		Definir os elementos de informação que são apresentados em cada contexto.	Ferramenta de busca presente em todas as páginas, mapa do sítio ou sistema, índice de palavras [Brasil 2010].
		Estabelecer políticas de acesso. Otimizar o tamanho dos elementos disponibilizados.	Políticas de segurança. Estimativas do tempo de carregamento dos elementos (imagens, gráficos, downloads) com base na velocidade de conexão do público-alvo [Brasil 2010].

Assim, mantendo-se parte da proposta do checkTrans completo, sugere-se que na fase de especificação de requisitos sejam definidos os requisitos de transparência, identificando-os por meio do checkTrans-E (Tabela 1). Em seguida, os requisitos de transparência devem ser descritos na forma de testes de aceitação (critérios de aceitação) para incluí-los nos contratos. Por fim, para aceitação do sistema entregue, deve ser verificado o seu atendimento aos critérios de aceitação descritos.

6. Avaliação da Proposta

Nesta seção é descrita a avaliação conduzida para verificar a efetividade do uso do checkTrans-E (Tabela 1) que contém as ações essenciais para auxiliar a definição dos requisitos de transparência.

A avaliação da proposta foi realizada no setor de projetos e desenvolvimento de software de uma instituição pública estadual. Normalmente, o processo de atividades que conduz a aquisição de um software terceirizado pela organização pública (contratante) tem início desde o estabelecimento das necessidades (especificação de requisitos), passando pela elaboração do contrato até a aceitação do sistema entregue [SOFTEX 2011].

Assim, de acordo com a proposta deste artigo, na fase de especificação de requisitos, definiram-se os requisitos de transparência, identificados por meio do checkTrans-E (Tabela 1). Em seguida, descreveram-se os requisitos de transparência na forma de testes de aceitação (critérios de aceitação) para incluí-los em contrato. Por fim, para aceitação do sistema entregue foi verificado o atendimento aos critérios de aceitação. Os papéis dos envolvidos foram simulados para criar uma situação semelhante à real de contratação de software terceirizado. O papel de empresa contratante foi realizado por um dos autores desse trabalho e o papel da empresa

terceirizada foi delegado a dois desenvolvedores efetivos da instituição pública em questão. A avaliação da proposta foi conduzida em dois casos, o primeiro sem o uso do checkTrans-E, ou seja, sem a preocupação explícita com os requisitos de transparência. No segundo caso, utilizou-se as ações sugeridas no checkTrans-E para definir os requisitos de transparência, explicitando essa necessidade. Em seguida, uma análise comparativa dos resultados dos dois casos foi realizada em relação aos itens de transparência atendidos.

O sistema desenvolvido nos dois casos foi o “Sistema de Recursos Humanos”, para os módulos de benefícios: Licença-Prêmio, Licença Saúde e Triênio. O objetivo desses módulos é possibilitar o cálculo, a inclusão e o controle de concessão de benefícios aos servidores da instituição em questão.

6.1. Caso 1 - Desenvolvimento Sem o Uso do CheckTrans-E

No primeiro caso, a empresa pública contratante (representada por um dos autores) não utilizou o checkTrans-E durante a especificação de requisitos do sistema. Os requisitos foram descritos na forma de critérios de aceitação, porém os requisitos de transparência desejados pela instituição foram descritos de maneira geral.

A empresa terceirizada (representada pelo desenvolvedor 1) foi orientada a implementar um sistema que atendessem a todas as exigências contidas nos critérios de aceitação pois eles seriam usados na validação do sistema entregue. Como parte dos requisitos gerais, foi dada a orientação que se desejava à transparência de software, listando os atributos de acessibilidade, usabilidade, entendimento, informativo e auditabilidade e explicando suas definições e características. Entretanto, não se explicitou como alcançar os requisitos de transparência de software esperados.

A validação do sistema entregue foi realizada observando se, mesmo sem terem sido explicitados, os requisitos mínimos de transparência haviam sido contemplados. Para isso, as características do sistema foram confrontadas com as diretrizes apresentadas pelas “ações essenciais de transparência” descritas no checkTrans-E (Tabela 1). Observou-se que o sistema entregue apresentou relevantes problemas relacionados ao não atendimento de requisitos mínimos de transparência e considerando-se cada problema como sendo uma ação mínima sugerida (Tabela 1) que não foi realizada, tem-se o descumprimento de aproximadamente 58% de ações voltadas à transparência. Entende-se que todas as faltas apresentadas poderiam ter sido evitadas com a realização das ações mínimas de transparência ainda na fase de especificação de requisitos, conforme sugerido na proposta desse artigo.

Para possibilitar um comparativo com o caso 2 as horas utilizadas para as fases de especificação de requisitos e validação foram anotadas pela contratante, e aquelas necessárias para o projeto e implementação foram informadas pela terceirizada. É importante frisar que os requisitos funcionais, apesar de não terem sido o foco dessa avaliação, também foram validados e atenderam aos critérios de aceitação especificados.

Na próxima seção são descritas as atividades desenvolvidas para o segundo caso desta avaliação da proposta.

6.2. Caso 2 - Desenvolvimento Com o Uso do CheckTrans-E

No segundo caso de desenvolvimento, a empresa contratante, representada por um dos autores, utilizou o checkTrans-E (Tabela 1) durante a especificação dos requisitos do sistema. Os requisitos do sistema, da mesma maneira que no primeiro caso, foram descritos na forma de “critérios de aceitação”, porém incluíram-se explicitamente os requisitos de transparência desejados pela instituição durante a especificação de requisitos.

No desenvolvimento da segunda aplicação, a empresa terceirizada foi representada por outro desenvolvedor (denominado desenvolvedor 2). A contratação de um desenvolvedor diferente foi importante para poder observar a relação da quantidade de horas necessárias para implementação dos requisitos de transparência nos dois casos. Assim, o desenvolvedor 2 foi orientado a implementar um sistema que atendesse a todas as exigências contidas nos critérios de aceitação descritos, pois eles seriam usados na validação do sistema entregue. É importante lembrar que a diferença entre o caso 1 e o caso 2 é que no segundo o checkTrans-E foi utilizado para definir os requisitos de transparência desejados. Nesse segundo caso, realizou-se a especificação de requisitos utilizando-se do checkTrans-E (Tabela 1) para identificar e definir os requisitos de transparência. Após definirem-se os requisitos de transparência, identificados por meio do checkTrans-E, eles foram descritos na forma de testes de aceitação (critérios de aceitação) conforme pode ser visto na Tabela 2.

Tabela 2. Critérios de aceitação descritos na avaliação da proposta

Atributos		Ação sugerida no checkTrans-E	Critério de Aceitação descrito para Sistema
Acessibilidade	Portabilidade	Especificar as diferentes plataformas atualmente utilizadas que devem ser compatíveis com o <i>software</i> .	- O <i>software</i> deve ser compatível com computadores do tipo <i>desktop</i> e <i>tablets</i> . - O <i>software</i> deve ser compatível com sistema operacional <i>Windows</i> (versões XP, Vista, W7 e W8/ 32 e 64bits).
		Definir o formato aberto a ser utilizado.	- Deve ser possível exportar os relatórios em formato RDF ou TXT.
		Especificar as diferentes tecnologias nas quais o <i>software</i> poderá ser utilizado.	- Deve ser possível executar a tarefa completa em navegadores <i>IEExplore</i> e <i>Chrome</i> .
	Disponibilidade	Definir os recursos e técnicas a serem utilizados para garantia da disponibilidade dos <i>sites</i> .	- O sistema deve ser disponibilizado primeiramente em ambiente de testes. - O sistema em ambiente de produção deve ter um servidor de aplicação

			dedicado.
		Definir as tecnologias de proteção da informação.	- O sistema deve ser acessível apenas em ambiente de intranet.
		Definir as facilidades de navegação, inclusive, para pessoas com deficiência.	- O sistema deve apresentar barra de acessibilidade para aumentar/diminuir fonte e alteração de contraste.
		Definir e classificar as informações sigilosas ou públicas.	- O sistema não deve apresentar em relatórios de consultas as informações de CPF, conta bancária, data nascimento, telefone e endereço do servidor.
Informativo	Corretude	Definir um plano de testes adequado ao domínio e a cobertura desejada.	- A contratada deve realizar testes de integração e testes de sistema.
		Definir e usar a representação da linguagem do ambiente a ser desenvolvido.	- O sistema deve conter a representação da linguagem descrita no vocabulário mínimo (será verificado por inspeção).
	Clareza	Descrever a arquitetura do <i>software</i> (definições gerais da arquitetura).	- Fornecer o diagrama de pacotes do sistema implementado.
	Integridade	Garantir a integridade das informações disponíveis para acesso.	- O sistema deve possibilitar a assinatura digital dos documentos gerados.
	Atualidade	Definir a política de atualização do <i>software</i> .	- O sistema deve apresentar a versão atual do <i>software</i> .
	Completeza	Identificar e descrever as necessidades (requisitos funcionais e não funcionais) dos diversos pontos de vista.	- O sistema deve atender aos requisitos funcionais e não funcionais descritos nas histórias de usuário.
	Comparabilidade	Identificar, aplicar e avaliar critérios de comparação entre os valores de referência e o <i>software</i> apresentado.	- O sistema deve permitir 100 acessos simultâneos (mínimo). - O sistema deve estar disponível 24 h por dia x 7 dias na semana.

Auditabilidade	Validade	Avaliar os artefatos de <i>software</i> produzidos no processo de requisitos.	Ação para contratante, portanto, não descrita como critério de aceitação do sistema.
	Usabilidade	Uniformidade	Padronizar ambientes.
Amigabilidade		Utilizar uma linguagem clara e usual.	- O sistema deve apresentar diretrizes de redação de acordo com o tipo de usuário final.
		Definir os elementos de informação que são apresentados em cada contexto.	- O sistema deve apresentar ferramentas de busca em todas as páginas.
		Estabelecer políticas de acesso. Otimizar o tamanho dos elementos disponibilizados.	- Os acessos ao sistema deverão ser via <i>Active Directory</i> . - Deve ser possível visualizar os relatórios em, no máximo, 2 segundos

De acordo com a proposta desse artigo a validação do sistema foi realizada por meio dos critérios de aceitação descritos para o sistema (Tabela 2) verificando se os requisitos de transparência haviam sido atendidos pelo sistema entregue pela contratante. Para cada critério de aceitação descrito para o sistema, a contratante realizou testes para confirmar ou não o cumprimento do requisito de transparência. No decorrer desse processo, os critérios de aceitação foram anotados com o tipo: atendido, parcialmente atendido e não atendido. Um critério de aceitação foi considerado “atendido” quando o requisito descrito foi atendido de forma satisfatória, “parcialmente atendido”, quando o requisito foi atendido, mas não de maneira satisfatória, e “não atendido” quando o requisito não foi atendido.

Como resultado da validação, verificou-se que dos 28 critérios de aceitação voltados à transparência, 25 foram atendidos. Esse número representa, aproximadamente, 90% de atendimento satisfatório aos requisitos de transparência definidos pela contratante. A Tabela 3 apresenta o resultado consolidado da validação dos requisitos de transparência.

Tabela 3. Resultado consolidado da validação

Critérios de Aceitação	Qnt	%
Atendido	25	89,29%
Parcialmente Atendido	2	7,14%
Não atendido	1	3,57%
Total=>	28	100%

Aponta-se que os critérios de aceitação anotados como “parcialmente atendido” foram: (i) “O software deve ser compatível com computadores do tipo desktop e tablets”, pois foi testado apenas em equipamentos do tipo desktop, devido à falta de disponibilidade de tablets na instituição e, (ii) “Fornecer o diagrama de pacotes do sistema implementado.”, uma vez que o diagrama entregue estava desatualizado (mostrando apenas uma versão anterior). O único critério de aceitação anotado como “não atendido” foi “O sistema de conter a representação da linguagem descrita no vocabulário mínimo”, devido a novos termos descobertos durante o desenvolvimento do sistema que não foram incluídos no documento de glossário.

As horas utilizadas para as fases de especificação de requisitos e validação e aquelas necessárias para o projeto e implementação também foram anotadas para fins de estudos comparativos.

6.3. Comparativos entre esforços de desenvolvimento

A partir dos dois casos de desenvolvimento realizadas nas seções anteriores (6.1 e 6.2) foi realizado um estudo comparativo para o desenvolvimento do sistema nas casos 1 e 2. Primeiramente, em relação aos esforços destinados para desenvolvimento o resultado comparativo pode ser observado na Tabela 4.

Tabela 4. Comparativo entre esforços de desenvolvimento

Fase	Caso 1 (sem uso do checkTrans-E)	Caso 2 (com uso do checkTrans-E)	Comparativo
	Qnt Homem/Hora	Qnt Homem/Hora	% Aumento Qnt Homem/Hora
Especificação de Requisitos	35	75	114%
Projeto e Implementação	50	85	70%
Validação	20	45	125%
Total	105	205	95%

Extrai-se desse comparativo que houve um aumento significativo de esforços para realizar o caso 2, principalmente, para as fases de especificação de requisitos e validação. Esse aumento total dos esforços de desenvolvimento se deve ao fato de no caso 2, diferentemente do caso 1, terem sido realizadas as ações sugeridas no checkTrans-E, o que resultou em um incremento de requisitos.

Observou-se que a maioria dos problemas de transparência relacionados no caso 1 como, por exemplo, publicidade indevida de informações sigilosas, inexistência de elementos de informação de contexto, tempo excessivo de carregamento dos relatórios, foram sanados na realização do caso 2. Assim, como um segundo resultado comparativo pode-se analisar o atendimento aos requisitos mínimos de transparência nos dois casos. No caso 1, esse atendimento foi de apenas 42%, ou seja, 53% menor em relação ao caso 2, que apresentou 90% de atendimento aos requisitos de transparência

especificados. A Tabela 5 mostra o resultado consolidado dessa segunda análise comparativa.

Tabela 5. Comparativo entre esforços de desenvolvimento

	Caso 1 (sem uso do checkTrans-E)	Caso 2 (com uso do checkTrans-E)	Comparativo
			% Aumento
Atendimento aos requisitos de transparência	42%	90%	114%

Como terceira análise, apresentam-se os resultados comparativos entre os esforços demandados para o desenvolvimento do sistema, com e sem o uso do checkTrans-E, e a efetividade em atender aos requisitos de transparência. Com base nas análises anteriores, observou-se que houve um aumento de 95% no total de esforços necessários para realizar o desenvolvimento de software comparando-se os dois casos realizados. Em compensação, também houve um aumento no atendimento às exigências mínimas de transparência, algo em torno de 114%, conforme apresentado na Figura 1, melhorando significativamente a qualidade em relação à transparência.

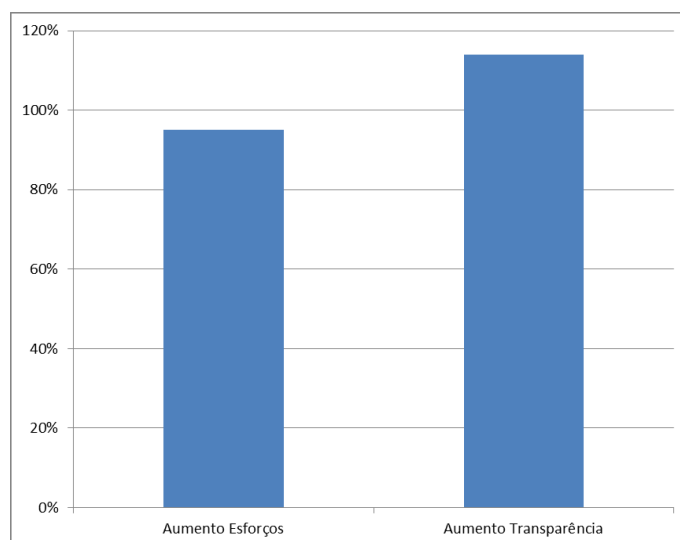


Figura 1. Comparativo de esforços de desenvolvimento versus o atendimento à transparência

Logo, comparando-se o aumento de esforços de desenvolvimento versus o aumento de atendimento à transparência observados nesta avaliação, tem-se uma evidência de que a proposta de um *checklist* com ações essenciais é válida.

7. Conclusões

As sociedades cada vez mais exigentes e a conseqüente regulamentação da transparência pública geraram uma nova demanda às instituições governamentais. Essa nova realidade exige esforços para enfrentar questões de natureza administrativa, técnica e tecnológica, atingindo também o desenvolvimento de software no setor público.

Nesse contexto, tendo em vista o desenvolvimento de software mais ágil, este artigo apresentou uma proposta de especialização de um checklist com ações para transparência. Como resultado sugere-se um checklist reduzido (checkTrans-E) com 19 ações essenciais para transparência que devem ser realizadas por instituições públicas ainda na fase de especificação de requisitos, visando à definição dos requisitos de transparência exigidos e/ou desejados pela instituição.

Para verificar a efetividade da proposta avaliou-se a inclusão de requisitos de transparência a partir do desenvolvimento de dois sistemas de software em uma empresa pública. Verificou-se que o uso do checkTrans-E aumentou em torno de 114% o atendimento às exigências mínimas de transparência, apesar do aumento de 95% no total de esforços necessários para realizar o desenvolvimento de software. Observou-se neste estudo de caso que o aumento dos requisitos de qualidade de transparência requereu um esforço no desenvolvimento quase na mesma proporção.

Referências

- Brasil. (1988) “Constituição da República Federativa do Brasil: promulgada em 5 de outubro de 1988”. Brasília, DF, Senado, 2004.
- Brasil. (2009) “Lei n. 131, de 27 de maio de 2009. Dispõe sobre disponibilização em tempo real de informações”. <https://www.planalto.gov.br/>
- Brasil. (2010) “Cartilha de Usabilidade”, Ministério do Planejamento, Orçamento e Gestão, Padrões Brasil e-Gov: Brasília: MP, SLTI, 2010. 50 p.
- Brasil. (2011a) “Acesso à Informação Pública: Uma introdução à Lei 12.527, de 18 de novembro de 2011”, Controladoria Geral da União. Brasília. <http://www.acessoinformacao.gov.br>.
- Brasil. (2011b) “Modelo de Acessibilidade em Governo Eletrônico: e-MAG”, Ministério do Planejamento, Orçamento e Gestão, Brasília: MP, SLTI.
- Brasil. (2011c) “Lei n. 12.527, de 28 de novembro de 2011. Regula o acesso a informações”. <https://www.planalto.gov.br/>
- Cappelli, C. (2009) “Uma Abordagem para Transparência em Processos Organizacionais Utilizando Aspectos”, 328 p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- C&L. (2013) “Cenários e Léxicos”, PUC-Rio. <http://transparencia.les.inf.puc-rio.br/>.
- Cysneiros, L. M. e Werneck, V. M. B. (2009) “An Initial Analysis on How Software Transparency and Trust Influence each other”. 12th Workshop on Requirements Eng., pp. 27-32.

- Connolly, D., Keenan, F. e Mccaffery, F. (2009) “Developing acceptance tests from existing documentation using annotations: An experiment. In Automation of Software Test”, AST '09. ICSE Workshop on (18-19 2009), pp. 123 –129.
- ER – PUC-Rio. (2014) “Catálogo de Transparência”, Grupo de Pesquisas em Engenharia de Requisitos da PUC-RIO. <http://www.er.les.inf.puc-rio.br/~wiki/index.php>
- Google. (2013) “A Evolução da Web”. <http://www.evolutionoftheweb.com/>
- Leal, A. C., Sousa, H. P., Leite, J. C. S. P. e Braga, J.L. (2011) “Transparência Aplicada a Modelo de Negócios”, Rio de Janeiro. <http://wer.inf.puc-rio.br/WERpapers/>
- Leite, J. C. S. P. e Cappelli, C. (2008) “Exploring i* Characteristics that Support Software Transparency”, In Proceedings of the 3rd International i* Workshop, CEUR Workshop Proceedings, Vol. 322, 2008, pp. 51-54 (<http://CEUR-WS.org/Vol-322/>).
- Leite, J. C. S. P. e Cappelli, C. (2010) “Software Transparency. Business & Information Systems Engineering”, Springer, 01 jun. 2010. Volume: 2, N: 3, p. 127-139.
- Macedo, F. F. de e Vilain, P. (2012), “Transparência no Processo de Software como Apoio à Publicidade da Administração Pública”, VII Simpósio Brasileiro de Sistemas de Informação. São Paulo, Brasil, pp.303-323.
- Macedo, F. F. de. (2014) “Transparência de Software como Apoio à Publicidade da Administração Pública”, Florianópolis, 2014. 148 p. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. PPGCC da Computação.
- Open Forum Foudation. (2011) “A Guide to Owning Transparency: How Federal Agencies Can Implement and Benefit from Transparency”; Open Forum Foudation.
- Oxford. (2012) “Oxford Dictionaries on line”. <http://oxforddictionaries.com>.
- Sayão, M. (2007) “Verificação e Validação em Requisitos: Processamento de Linguagem Natural e Agentes”, 205 p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- Silva, R. P. (2007) “UML – Modelagem Orientada a Objetos”, Florianópolis: Visual Books, 2007. ISBN 85-7502-205-9.
- SOFTEX. (2011) “MPS. BR - Guia de Aquisição”. SOFTEX, Associação para Promoção da Excelência do Software Brasileiro.
- Tanenbaum, A. S. (2003) “Redes de computadores.” 1ª.ed, Rio de Janeiro (RJ): Elsevier, 2003. 945p. ISBN 788535211856.
- W3C. (2010) “Manual dos dados abertos: governo”, 2010. (Trad. Adapt. [openmanualdata.org](http://www.w3c.br/pub/Materiais/PublicacoesW3C)), <http://www.w3c.br/pub/Materiais/PublicacoesW3C>.
- Yu-Cheng, T., Thomborson, C. e Tempero, E. (2011) “Illusions and Perceptions of Transparency in Software Engineering”, Software Engineering Conference (APSEC), 18th Asia Pacific, vol., no., pp.365-372, 5-8 Dec. 2011.doi: 10.1109/APSEC.2011.42.