



## ELICITAÇÃO DE REQUISITOS DE SOFTWARE NO SETOR PÚBLICO: LIÇÕES APRENDIDAS E RECOMENDAÇÕES PARA MITIGAÇÃO DE RISCOS

### RESUMO

A atividade de elicitação de requisitos é reconhecida como uma das mais difíceis do processo de engenharia de software, é executada desde o início do projeto, tem um grande impacto nas atividades subsequentes e é essencialmente humana e subjetiva. Além disso, no setor público há características peculiares que devem ser observadas, tais como as influências das políticas de governo e Estado. Antecipar as necessidades e identificar adequadamente os requisitos nas etapas iniciais de um projeto contribuem para reduzir a possibilidade de mudanças ao longo do ciclo de vida e principalmente nas fases finais, onde são muito mais onerosas e de maior impacto, podendo até inviabilizar a solução. Este artigo apresenta um mapeamento sistemático da literatura na base Scopus sobre projetos de desenvolvimento de software no setor público com o objetivo de identificar as lições aprendidas que impactam na atividade de elicitação. Técnicas e recomendações que podem auxiliar para a mitigação dos riscos nesta atividade são abordadas, além de indicar que gerenciamento dos riscos de requisitos é crítico para atingir a melhor performance do projeto. A aplicação dessas recomendações e a observação das características inerentes ao setor público podem contribuir para um processo mais eficiente e auxiliar a evitar o desperdício do dinheiro público.

**PALAVRAS-CHAVES:** Engenharia de Software; Elicitação; Requisito; Setor Público.

## REQUIREMENTS ELICITATION OF SOFTWARE IN THE PUBLIC SECTOR: LESSONS LEARNED AND RECOMMENDATIONS FOR RISK MITIGATION

### ABSTRACT

The requirements elicitation activity is recognized as one of the most difficult in software engineering process, is performed since the beginning of the project, has a major impact on subsequent activities, and is essentially human and subjective. Moreover, the public sector's unique characteristics that must be observed, such as the influences of government policies and state. Anticipate needs and properly identify the requirements in the early stages of a project contribute to reduce the possibility of changes over the life cycle and especially during the final stages, where they are much more expensive, have a greater impact, and could even make impracticable the solution. This article presents a systematic mapping of literature in Scopus on software development projects in the public sector in order to identify lessons learned that impact the elicitation activity. Techniques and recommendations that can help to mitigate the risks in this activity are addressed, besides indicating that risk management requirements is critical to achieve the best performance of the project. The implementation of these recommendations and observations of the characteristics inherent in the public sector can contribute to a more efficient process and help avoid the waste of public money.

**KEYWORDS:** Software Engineering; Elicitation; Requirement; Public Sector.

*Revista Brasileira de Administração Científica, Aquidabã, v.3, n.2, Ago 2012.*

ISSN 2179-684X

SEÇÃO: Artigos

TEMA: **Sistemas de Informação**

*Anais do Simpósio Brasileiro de Tecnologia da Informação (SBTI 2012)*



DOI: 10.6008/ESS2179-684X.2012.002.0014

**Luciano Antonio Costa**

*Universidade Federal de Santa Catarina, Brasil*

<http://lattes.cnpq.br/9324084297796198>  
[lacosta@eqc.ufsc.br](mailto:lacosta@eqc.ufsc.br)

**Alessandra Casses Zoucas**

*Universidade Federal de Santa Catarina, Brasil*

<http://lattes.cnpq.br/2007011625196648>  
[alessandrazoucas@gmail.com](mailto:alessandrazoucas@gmail.com)

**João Bosco da Mota Alves**

*Universidade Federal de Santa Catarina, Brasil*

<http://lattes.cnpq.br/4046984728286369>  
[jbosco@inf.ufsc.br](mailto:jbosco@inf.ufsc.br)

Recebido: 10/06/2012

Aprovado: 15/08/2012

Avaliado anonimamente em processo de pares cegas.

*Referenciar assim:*

*COSTA, L. A.; ZOUCAS, A. C.; ALVES, J. B. M.. Elicitação de requisitos de software no setor público: lições aprendidas e recomendações para mitigação de riscos. Revista Brasileira de Administração Científica, Aquidabã, v.3, n.2, p.214-226, 2012.*

## INTRODUÇÃO

A implementação de sistemas computacionais por um governo, seja para a automação de uma atividade (ex. folha de pagamento) ou para uma solução de governo eletrônico (ex. eleições), é reconhecida como uma tarefa complexa, uma vez que comumente é de interesse de mais de um setor da sociedade ou de um grupo político e envolve mais de uma área da própria administração.

Essa característica torna ainda mais difícil a tarefa da engenharia de *software* e, portanto, demanda experiência e conhecimento dos agentes envolvidos para que a solução seja implementada e entregue da forma correta e para o cliente correto.

No processo da engenharia de *software* as fases iniciais de concepção e elaboração são críticas para a definição da solução. Essas fases exigem aproximadamente 25% do esforço total em 40% do tempo do projeto, onde 40% deste esforço é dedicado a disciplina de requisitos (MARTINS, 2007, p. 232, 242).

É nesse momento que a equipe de desenvolvimento constrói a compreensão do problema do cliente e define todas as etapas seguintes. As mudanças ocorridas no início do projeto são muito menos onerosas e de menor impacto que as ocorridas em etapas finais de desenvolvimento, onde podem até inviabilizar a solução construída.

Os objetivos deste artigo são, com ênfase no processo de elicitação de requisitos, apresentar a diferença entre políticas de governo e Estado, apresentar um mapeamento sistemático da literatura sobre estudos de caso de desenvolvimento de *software* para o setor público e recomendar técnicas para mitigação dos riscos identificados.

O público-alvo potencial restringe-se aos engenheiros de *software*, a quem compete construir o conhecimento sobre o problema, mas se direciona principalmente aos gestores públicos, já que estes poderão facilitar o processo ao indicar as melhores fontes de requisitos e a evitar obstáculos no desenvolvimento.

## REVISÃO TEÓRICA

A definição de requisito de *software*, segundo o SWEBOK (ABRAN et al., 2004, p.33-36), é de uma propriedade que deve ser exibida ou adaptada na condição de resolver algum problema particular no mundo real. Desta forma, os requisitos de um dado *software* são tipicamente uma combinação complexa desses próprios, de diferentes pessoas, em diferentes níveis de uma organização e do ambiente em que este *software* irá operar.

Por sua vez, o processo de elicitação de requisitos tem por objetivo identificar de onde provêm os requisitos do *software* e como coletá-los. Cabe àquele processo construir uma compreensão do problema que o *software* se propõe a resolver. É uma atividade essencialmente humana, tipicamente realizada em conjunto com o cliente e com os usuários, pois inicialmente é preciso entender o problema para então propor uma possível solução (PFLEEGER; ATLEE, 2010).

Durante este processo são identificados os *stakeholders* e estabelecidos os relacionamentos entre a equipe de desenvolvimento e o cliente.

Por suas características é reconhecida como uma atividade difícil, uma vez que o engenheiro de *software* deve ser capaz de perceber que os usuários podem apresentar dificuldades em descrever suas tarefas, podem não fornecer informações importantes, podem não estar motivados ou mesmo se recusarem a cooperar, além do que a informação pode ser mal compreendida.

Os principais pontos que deverão ser cobertos pelo elicitador são: as metas do *software*, ou seja, os objetivos de mais alto nível que a solução deverá alcançar; o domínio do conhecimento necessário para a aplicação; a identificação, gerenciamento e representação dos pontos de vista dos *stakeholders* e seus grupos; o ambiente operacional e organizacional ao qual será submetido o *software*.

Quando a elicitación envolve o setor público, a distinção entre as políticas de governo e de Estado contribui para a identificação correta dos requisitos. Contudo, no cotidiano é comum observar o uso incorreto dos termos governo e Estado como sinônimos, o que pode criar problemas durante o desenvolvimento ou reduzir a vida útil do *software*.

O Estado é definido por sua “posse soberana de um território e seu povo e é a expressão do controle do governo sobre estes” (FLINT, 2006, p. 105). Esta definição corrobora com a de Azambuja (2008, p. 34) que considera que “os três elementos essenciais e suficientes que compõe o Estado são: uma população, um território e um governo independente, ou quase, dos demais Estados”. Também pode ser compreendido como o conjunto de instituições permanentes que possibilitam a ação do governo (HÖFLING, 2001).

Já o governo se refere ao grupo de pessoas que estão encarregadas de administrar o Estado em um tempo particular. Em um Estado democrático, onde as sucessões destes grupos se dão através de eleições (BEALEY, 1999, p. 147), parte da comunidade propõe um conjunto de programas e projetos para a sociedade como um todo, conforme sua orientação política (HÖFLING, 2001).

Estas propostas podem se tornar políticas de Estado ou de governo, sendo que para este último, a vigência tende a ser menor, uma vez que foi criada pelo grupo que naquele momento está no poder para atendimento de sua agenda política.

Considera-se então, segundo Oliveira (2011), que as políticas provenientes do Executivo através de um processo elementar visando responder as demandas da agenda política interna são de governo. Já as de Estado são aquelas que envolvem mais de uma agência do Estado e tem a incidência em setores mais amplos da sociedade.

Um exemplo é a criação do Plano de Desenvolvimento da Educação, que reuniu de dezenas de programas procurando dar direção à política educacional no Brasil, isto é, a implementação de uma política de Estado através da integração de diversos programas de governo. Esta visão permeia a condução da gestão pública.

Em verdade, a administração pública moderna pode ser considerada como um ecossistema, segundo Bresciani *et al.* (2003), onde existe a interação das diversas entidades presentes nos setores a fim de alcançar seus objetivos. Por sua vez, tais entidades gozam geralmente de certo nível de independência política e econômica para sua atuação, o que as permite fazer suas próprias escolhas em termos tecnológicos, organizacionais e estratégicos.

A inserção de atores do governo nas estruturas cognitivas, culturais, sociais e institucionais influenciam o *design*, as percepções e os usos das tecnologias de informação. Portanto, o sucesso de investimentos no setor público para iniciativas de governo eletrônico tem como maior desafio e mais relevante instrumento o planejamento de atividades a fim de prover sustentabilidade e agregar valor a solução (VISCUSI *et al.*, 2010, p. 4-5).

## METODOLOGIA

O mapeamento sistemático foi o método utilizado nesta pesquisa, onde se definiram três etapas: planejamento, execução e avaliação. A pesquisa foi efetuada na base Scopus<sup>1</sup>, por ser referência em bases indexadas e ter uma quantidade representativa de informações das publicações da área de interesse.

A fase de planejamento construiu o protocolo de busca que relaciona as informações utilizadas na pesquisa, permitindo sua reutilização futura e oferecendo oportunidade para acompanhar a evolução da literatura sobre o tema estudado de forma padronizada.

A definição do protocolo teve por objetivo selecionar publicações que apresentassem estudos de caso de projetos de desenvolvimento de *software* para o setor público e que contivessem ocorrências com relação à atividade de elicitação de requisitos.

Foram objetos da pesquisa artigos redigidos em inglês, que compõem a maior parte da base, ou português, para contemplar a situação local brasileira. Os termos da consulta final para a pesquisa definidos após o refinamento foram que:

- Contivesse as palavras-chaves (*keywords*) “setor público” (“*public sector*”) ou “governo” (“*government*”), já que ambas são utilizadas como sinônimos e fornecem a restrição para a área objeto deste artigo;
- Contivesse em seu título, resumo (*abstract*) ou palavras-chaves os termos “estudo de caso” (“*case study*”), “sistema de informação” (“*information system*”) e “projeto” (“*project*”), pois o interesse está em situações reais de desenvolvimento de *software* já executadas ou em execução. O termo “sistema de informação” foi utilizado ao invés de “*software*” por trazer uma busca mais precisa, já que ao se utilizar este último, sempre há retorno excessivo de textos que contemplam apenas sua utilização;
- Não contivesse em seu título, resumo (*abstract*) ou palavras-chaves os termos “abordagem” (“*approach*”) ou “proposta” (“*proposal*”), visto que o interesse está em relatos de casos já realizados;
- Tivessem sido publicados entre 2007 e 2012, ou seja, são relevantes para a pesquisa apenas os últimos 6 anos.

---

<sup>1</sup><http://www.scopus.com>

A seleção dos artigos seguiu os critérios de inclusão e exclusão conforme apresentados no quadro 1. Com base nos termos arrolados, as consultas mostradas no quadro 2, abaixo, foram executadas e a primeira retornou trinta documentos, sendo que somente cinco estavam disponíveis no formato completo, ver quadro 3, a seguir, e a segunda não retornou resultado algum. Os artigos disponíveis foram então analisados.

**Quadro 1:** Critérios de Inclusão e Exclusão de Artigos de Pesquisa.

TIPO DO CRITÉRIO	CRITÉRIO
Inclusão	Artigos em inglês ou português que apresentem estudos de caso do processo de desenvolvimento de <i>software</i> no setor público.
Exclusão	Artigos inacessíveis ou indisponíveis para pesquisa.
	Artigos que não apresentem estudos de caso do processo de desenvolvimento de <i>software</i> no setor público.
	Artigos que não estejam redigidos em inglês ou português.

**Quadro 2:** Componentes da Busca Realizada (*queries* estasbelecidas).

MÉTODO	BIBLIOTECAS	IDIOMA	TERMOS DA CONSULTA
Busca em publicações indexadas por bibliotecas digitais conveniadas com o Portal da CAPES	Scopus	Inglês, idioma padrão utilizado nas bibliotecas digitais	( KEY( "public sector" OR government ) AND TITLE-ABS-KEY( "case study" AND "information system" AND project ) AND NOT TITLE-ABS-KEY( approach OR proposal ) ) AND PUBYEAR > 2006
			( KEY( "setor público" OR governo ) AND TITLE-ABS-KEY( "estudo de caso" AND "sistema de informação" AND projeto ) AND NOT TITLE-ABS-KEY( abordagem OR proposta ) ) AND PUBYEAR > 2006

**Quadro 3:** Artigos Resgatáveis na Pesquisa Efetuada na Base Scopus.

AUTORES	TÍTULO	CONSIDERAÇÕES
Gauld (2007)	<i>Public sector information system project failures: lessons from a New Zealand hospital organization</i>	Estudo de caso com as lições aprendidas do projeto, apresentando as escolhas entre os possíveis sistemas.
Sirajul Islam e Grönlund (2010)	<i>The Bangladesh national biometric database: a transferable success?</i>	Estudo de caso da criação da base, com breve revisão da literatura e considerações sobre um projeto anterior fracassado.
Hara <i>et al.</i> (2010)	<i>Current organic waste recycling and the potential for local recycling through urban agriculture in Metro Manila</i>	Não é um estudo de caso de desenvolvimento, mas sim do processo de reciclagem, onde uma das etapas tem auxílio de uma aplicação.
Li <i>et al.</i> (2011)	<i>Locating new coal-fired power plants with carbon capture ready design – a GIS case study of Guangdong province in China</i>	Não é um estudo de caso de desenvolvimento, mas sim do uso de um software de GIS para planejamento da localização de construções de CCR ( <i>Carbon Capture Ready</i> )
Abraham <i>et al.</i> (2011)	<i>Transforming healthcare with information technology in Japan: a review of policy, people, and progress</i>	Apresenta uma revisão do impacto das políticas de governo no Japão sobre o uso de sistemas de informação para saúde e um estudo de caso.

Com o objetivo de agregar informações sobre os projetos brasileiros, uma busca exploratória foi realizada nas publicações disponíveis de órgãos públicos, da qual retornaram

alguns relatórios técnicos do Tribunal de Contas da União (TCU), sendo que destes enquadravam-se nos critérios de seleção apenas dois; um sobre o Sistema Nacional de Integração de Informações em Justiça e Segurança Pública, Infoseg (BRASIL, 2007), e outro sobre o Sistema Informatizado de Controle de Óbitos, Sisobi (BRASIL, 2010). O relato sobre o desenvolvimento do CRAMS (*Case Recording and Management System*) contido no relatório do Escritório Parlamentar de Ciência e Tecnologia do Reino Unido (REINO UNIDO, 2003), que não consta nas bases indexadas, também foi agregado ao material por ser relevante para o objeto da pesquisa.

## DISCUSSÃO

Um estudo empírico (HAN; HUANG, 2007) considerou a informação de risco de 115 projetos de *software* e sob a ótica da análise MANOVA revelou que a probabilidade de ocorrência e a composição do impacto têm diferenças significantes em seis dimensões de risco. Em especial, a dimensão que representa os requisitos foi a que obteve a maior significância independente da performance do projeto. Os resultados sugerem que o gerenciamento dos riscos de requisitos é crítico para atingir a melhor performance do projeto porque mesmo os projetos de alta performance ainda tem um alto grau de risco nessa dimensão.

O gerenciamento dos riscos ligados a requisitos é uma atividade necessária e ganha ainda importância quando o projeto está em áreas de governo ou possui ligações com estas. O profissional deve ser capaz de discernir quais são os reais *stakeholders* do projeto e suas intenções, visto que a composição da estrutura de governo é feita por correntes políticas, por vezes conflitantes, e a falta desta distinção pode levar o projeto a falha devido a uma transição em um setor.

Na fase de análise dos artigos selecionados para a pesquisa foram identificados os fatores críticos e problemas relacionados com a elicitação de requisitos, os quais são exibidos no quadro 4. Destaque-se tal análise já incorpora a ampliação da base de dados, com as adições aludidas, e destaca os sistemas associados aos projetos de *software* incorporados ou resultantes nas sequências de seus desenvolvimentos (ou aparição).

**Quadro 4:** Relação dos Fatores Críticos e Problemas Identificados.

PUBLICAÇÃO	SISTEMA	PONTOS RELEVANTES
Gauld (2007)	Suíte SMS	<ul style="list-style-type: none"> <li>• Divergências entre os grupos na escolha</li> <li>• Requisitos ignorados</li> <li>• Necessidade de reengenharia organizacional</li> <li>• Influência da complexidade organizacional e da política</li> <li>• Falta de envolvimento e suporte do usuário</li> </ul>
Sirajul Islam e Grönlund (2010)	1º Sistema	<ul style="list-style-type: none"> <li>• Redução de orçamento</li> <li>• Aspectos culturais ignorados</li> </ul>
	PERP	<ul style="list-style-type: none"> <li>• Comprometimento político</li> <li>• Recursos adequados</li> </ul>

		<ul style="list-style-type: none"> <li>• Foco nos <i>stakeholders</i></li> </ul>
Abraham <i>et al.</i> (2011)	CoMeT	<ul style="list-style-type: none"> <li>• Necessidade de padronização</li> <li>• Divulgação dos casos de sucesso</li> <li>• Ausência de regulamentação</li> <li>• Inclusão dos usuários no processo</li> </ul>
Reino Unido (2003)	CRAMS	<ul style="list-style-type: none"> <li>• Problemas no piloto ignorados</li> <li>• Problemas sérios de usabilidade</li> <li>• Orçamento “estourado”</li> <li>• Riscos técnicos subestimados</li> <li>• Especificação inadequada</li> <li>• Comunicação ineficaz</li> </ul>
Brasil (2007, 2010)	Infoseg	<ul style="list-style-type: none"> <li>• Ausência de definições claras e formais</li> <li>• Ausência de atribuições e responsabilidades</li> <li>• Cooperação informal</li> </ul>
	Sisobi	<ul style="list-style-type: none"> <li>• Ausência de padrões</li> <li>• Processo com falhas e ineficiente</li> </ul>

Priorizar e identificar quais são as políticas de Estado e quais são as de governo que geram requisitos para o *software* é preponderante para reduzir a probabilidade de mudanças de requisitos e escopo ao longo do desenvolvimento do projeto, bem como para ampliar o tempo de vida da solução. Como já visto, as políticas de Estado tendem a ser mais estáveis e duráveis que as de governo, pois foram elaboradas envolvendo mais agentes que não só o governo atual.

Leffingwell e Widrig (2003, p.7-8) explicam que os três fatores mais apontados como desafios para um projeto são: informações do usuário não extraídas (13% dos projetos), requisitos e especificações incompletas (12%); alteração de requisitos e especificações (12%). Em contrapartida, os três fatores primários de sucesso para um projeto são: envolvimento do usuário (16% dos projetos bem sucedidos); suporte da gerência (14%); requisitos claros (12%). Apresenta também uma relação de custos entre as fases do projeto e ressalta que o menor custo para reparar um defeito é no estágio de levantamento de requisitos, sendo que a relação para o estágio de manutenção é de aproximadamente 200:1 (LEFFINGWELL; WIDRIG, 2003, p.10).

Para reduzir os riscos e tornar o processo de elicitação de requisitos mais eficiente existem técnicas que podem ser utilizadas. Contudo, para uma escolha assertiva é importante também conhecer as três síndromes que são inerentes a essa atividade: “Sim, mas”, “Ruínas não Descobertas” e “Usuário e Desenvolvedor” (LEFFINGWELL; WIDRIG, 2003, p.89).

A síndrome “Sim, mas” é parte da natureza humana e, portanto, integra de forma inerente o processo de desenvolvimento. Ela surge quando usuário vê a implementação pela primeira vez e então percebe que aquilo que foi descrito não é o que foi entregue ou não era exatamente o que necessitava. No caso do Reino Unido (2003) observa-se a consequência nos problemas de usabilidade e Sirajul Islam e Grönlund (2010) também indicam os aspectos culturais ignorados no primeiro sistema, cujo projeto falhou.

A outra síndrome, a “Ruínas não Descobertas” trata da sensação de quanto mais é encontrado, mais existe para ser descoberto. Neste caso nunca se tem o sentimento de ter

encontrado tudo o que era necessário e a questão é determinar quando o que é descoberto é o suficiente para a fase, sendo que o restante poderia ser obtido nas próximas etapas. Os relatórios técnicos do TCU (BRASIL, 2007) (BRASIL, 2010) apontam que os processos envolvidos e os requisitos não estavam definidos e formalizados, o que pode indicar que houve o início de uma etapa sem o conjunto das definições mínimas necessárias.

A terceira é a do “Usuário e Desenvolvedor”, que surge da deficiência de comunicação entre o usuário e o desenvolvedor, onde ambos têm diferentes motivações e objetivos. Neste caso o processo de comunicação entre eles deve ser o mais próximo e efetivo possível. Gauld (2007) cita um conjunto de motivos para a falha do projeto, dentre estes as divergências entre os grupos nas escolhas e a falta de envolvimento e suporte do usuário. No projeto CoMeT (ABRAHAM et al., 2011) também é indicada a necessidade de inclusão de usuários no processo.

Entre as técnicas que se podem empregar para elicitar requisitos, a mais usual é a entrevista, que é simples e direta. Ela pode ser aplicada na maior parte dos casos e é apropriada para buscar requisitos não descobertos e explorar soluções. Aliada a esta técnica também se pode agregar a aplicação de questionários.

Uma das técnicas com grande potencial para elicitação é o *workshop*, onde todos os *stakeholders* chave são reunidos por um curto período para uma reunião intensa e focada. O objetivo é que a comunicação seja clara e eficaz, já que todos os interessados estarão presentes, e que ao término da sessão, todos os assuntos que interferem ao projeto tenham tido uma definição. Uma grande vantagem da técnica é que as decisões são tomadas em um período muito mais curto do que em outras situações e com menor risco de falha ou falta de requisitos, já que os agentes decisores chave estarão presentes.

O *workshop* pode contribuir para construção de um entendimento e comprometimento coletivo entre o *stakeholders*, reduzindo as divergências entre grupos relatada por Gauld (2007), melhorando a comunicação indicada no relatório do Reino Unido (2003) e formalizando a cooperação apontada pelo TCU (BRASIL, 2007). Esta técnica também ser utilizada para auxiliar na inclusão de usuários no processo descrito por Abraham (2011). Entretanto, apesar de poder também contribuir para o entendimento a complexidade organizacional, listada por Gauld (2007), pode não ser suficiente para garantir o comprometimento político, que foi preponderante para o projeto PERP (SIRAJUL ISLAM; GRÖNLUND, 2011).

Quando há necessidade de identificar uma solução para um problema ou novas ideias, o *brainstorming* pode ser adequado. Ele possui duas fases, a de geração de ideias e a de redução destas (aglomeração). Na primeira o objetivo é que os participantes expressem qualquer ideia sobre o tema, sem nenhuma restrição ou crítica, inclusive reutilizando as já sugeridas. Na segunda parte, todas as ideias são analisadas e algumas são combinadas e extraídas. Esta técnica é mais indicada para grupos pequenos e que, preferencialmente, já exista a cooperação entre os participantes. Nos artigos selecionados não foi identificada sua aplicação.



O uso de *storyboards* auxilia a evitar a síndrome “Sim, mas”, já que em estágios muito iniciais o usuário já conhece as interfaces do sistema e contribui a reduzir a “Ruínas não Descobertas” ao receber os *feedbacks*. A utilização destes artefatos poderia ter auxiliado a evitar os problemas de usabilidade do CRAMS (REINO UNIDO, 2003) e os processos com falhas e ineficientes do Sisobi (BRASIL, 2010).

A etnografia, técnica baseada na observação, é indicada para compreender os requisitos sociais e organizacionais (HAGUETTE, 1992). Ela visa promover a familiarização do analista de requisitos com o ambiente onde o sistema será implantado e entender a cultura organizacional. Nesta técnica, o analista observa e toma notas sobre as características das tarefas em execução. Antes do estudo de observação é necessário identificar as áreas a serem observadas. Durante o estudo de observação deve-se observar atentamente como o trabalho é realizado considerando as operações manuais e automatizadas e deve-se ainda acessar os documentos usados em cada atividade observada. As descobertas devem ser documentadas e apresentadas para os envolvidos validarem o entendimento e assim garantir que o trabalho observado foi compreendido e documentado adequadamente. Esta técnica auxilia especialmente na eliciação de requisitos implícitos, que refletem os processos reais. A etnografia pode ser usada para complementar descobertas obtidas por outras técnicas. Sua adoção poderia ter mitigado os riscos materializados e citados por Sirajul Islam e Grönlund (2010) que apontam que aspectos culturais foram ignorados no primeiro sistema, cujo projeto falhou.

Outra técnica que pode apoiar na qualidade dos requisitos elicidados é o JAD (*Joint Application Design*), metodologia criada pela IBM do Canadá em 1977 para mediação de *brainstorming* com o objetivo de facilitar o desenvolvimento de aplicações de Sistemas de Informação (WOOD; SILVER, 1995). O JAD facilita a criação de uma visão compartilhada do que o produto de *software* deve ser. Esta técnica consiste em cinco fases: definição do projeto; pesquisa; preparação para a sessão; sessão; documento final. A coleta de informações é realizada durante as fases 1, 2 e 3. A validação das informações é obtida durante a sessão JAD. Portanto, as pessoas certas devem ser envolvidas nestas sessões, além da presença de um facilitador para manter o foco e minimizar a ocorrência de discussões improdutivas. Desta forma, promove cooperação e entendimento entre as várias partes interessadas no sistema que será desenvolvido. A utilização desta técnica poderia ter auxiliado o projeto apresentado por Gauld (2007) a convergir na escolha e aos auditados pelo TCU a construírem as definições.

Todas as técnicas mencionadas não esgotam as possibilidades, considerando que estas podem ser adaptadas e ainda existirem outras que não foram cobertas por esta pesquisa. Contudo, é altamente recomendável que estas sejam adotadas durante a execução de um processo definido de desenvolvimento de requisitos que adote boas práticas reconhecidas pelo mercado de *software* como as publicadas em normas e modelos nacionais e internacionais CMMI-DEV (SEI, 2011), ISO 12207 (ISO/IEC, 2008) e MPS.BR (SOFTEX, 2012).

O processo de elicitación e análise de requisitos é um processo iterativo, com uma contínua validação de uma atividade para outra. As técnicas apresentadas anteriormente podem ser combinadas e usadas de maneira complementar em cada uma das atividades do processo. Neste sentido, Sommerville (2010) propõe um processo genérico de levantamento e análise de requisitos contendo as seguintes atividades:

- Compreensão do domínio: é a atividade na qual os analistas de requisitos visam desenvolver sua compreensão do domínio da aplicação;
- Coleta de requisitos: é a atividade em que os analistas de requisitos passam a interagir com os *stakeholders* do sistema para elicitare seus requisitos. A compreensão do domínio acaba por ser refinada durante essa atividade;
- Classificação: é a atividade onde o analista de requisitos irá organizar os requisitos elicitados em grupos coerentes;
- Resolução de conflitos: em casos onde são envolvidos muitos *stakeholders* pode haver requisitos conflitantes e esta atividade visa remover os conflitos existentes;
- Definição das prioridades: o escopo de um projeto apresenta um conjunto de requisitos onde há uns mais prioritários do que os outros. Nesta atividade o analista de requisitos tem a responsabilidade de interagir com os *stakeholders* para priorizar os requisitos elicitados para o escopo do projeto;
- Verificação de requisitos: os analistas verificam os requisitos para garantir a qualidade antes de entrarem para a fábrica de software e se estão de acordo com o que os *stakeholders* esperam do sistema.

Na busca da elicitación dos requisitos a escolha de quais técnicas aplicar e qual a variação destas depende, por exemplo, do tempo disponível, do tipo da aplicação, das características da equipe de desenvolvimento, do cliente e do *stakeholder*, além da escala do problema e experiência do elicitador. É importante destacar que não há um caminho certo e único para se elicitare, mas sim um conjunto de boas escolhas que mitigam os riscos e contribuem para o sucesso do projeto.

## CONCLUSÕES

Para Gauld (2007), o processo de desenvolvimento de um sistema de informação tende a falhar mais que o planejado pelo gestor. Em geral 20% a 30% são um fracasso total ou são abandonados e de 30% a 60% falham parcialmente, extrapolando recursos financeiros ou de tempo.

Segundo dados da *Software Engineering Institute*<sup>2</sup> apenas 35% dos projetos são finalizados no tempo e custo planejados e se se analisar somente a área pública esse número é ainda menor, indicando 16% no Reino Unido (2004). Não é simples melhorar esta estatística, mas ao se aprender as lições de projetos já executados ou que estão em execução e buscar melhorias, pode-se obter a maturidade para se elevar esse percentual.

Já nas etapas iniciais deve-se ter ciência que alguns fatores que dificultam os projetos no setor público são inerentes a ele e devem ser observados, tais como os relacionados no relatório

<sup>2</sup> Acesso em 02/03/2012: <<http://www.sei.cmu.edu/solutions/softwaredev>>

da Academia Real de Engenharia em conjunto com a Sociedade Britânica de Computação (2004): a alta visibilidade, a cultura de aversão ao risco do serviço público, necessidade de seguir uma agenda política e em muitos casos, a enorme escala e complexidade.

A alocação dos recursos necessários para a elicitação de requisitos também é essencial. O elicitor deve ter capacitação para conseguir capturar as informações necessárias empregando as técnicas adequadas. Para isso deve identificar as características do ambiente em que irá trabalhar e, portanto, compreender que há diferenças significativas em um projeto que envolva entes públicos. Diferenciar e priorizar requisitos provenientes de políticas de Estado, selecionando adequadamente as de políticas de governo, contribui para redução de riscos de mudanças e para um ciclo de vida maior do *software*.

Dentre as técnicas apresentadas por Leffingwell e Widrig (2003), a de *workshops* talvez seja a que mais contribua ao processo. Ao reunir os *stakeholders*, que podem representar correntes políticas distintas, para discutir e elicitar os requisitos o processo de decisão é tomado em conjunto e a comunicação não sofre intermediação. Desta forma, o processo tende a ter mais celeridade e um melhor entendimento entre os participantes.

Através dos relatos dos desenvolvimentos de *software* analisados, é recorrente a dificuldade ou desconhecimento das necessidades dos usuários finais. É comum que somente na etapa de implantação estes tenham conhecimento da ferramenta, o que remete à primeira síndrome, a “Sim, mas”. Identificar claramente quem serão os usuários do produto e o uso da técnica de *storyboard* mitigam esses riscos, que por aparecerem somente ao final do desenvolvimento têm um custo elevado para correção.

Durante o processo de pesquisa para este artigo, foi constatado que há poucas publicações, sejam artigos ou relatórios, sobre as lições aprendidas ou mesmo a execução de um projeto de *software*, em especial quando não é bem sucedido. Esta situação é esperada quando o projeto é de um setor privado, porém não deveria ocorrer no setor público, já que as informações deveriam estar disponíveis, no mínimo, aos demais setores daquele Estado, para que o processo seja melhorado e se evite o desperdício de dinheiro público.

Como trabalhos futuros, a publicação de relatórios de projetos de *software* na área pública, constando seus estados atuais, cronogramas e orçamentos, bem como as lições aprendidas seriam desejáveis. Uma análise posterior sobre estes, estabelecendo comparativos e indicadores, também contribuiria significativamente ao melhor entendimento da temática em discussão.

## REFERÊNCIAS

ABRAHAM, C.; NISHIHARA, E.; AKIYAMA, M.. Transforming healthcare with information technology in Japan: a review of policy, people, and progress. **International Journal of Medical Informatics**, v.80, n.3, p.157-170, 2011. DOI: <http://dx.doi.org/10.1016/j.ijmedinf.2011.01.002>

ABRAN, A.; MOORE, J. W.; BOURQUE, P.; DUPUIS, R.; TRIPP, L. L.. **Guide to the Software Engineering Body of Knowledge** (SWEBOOK). California : IEEE Computer Society. ISO Technical Report ISO/IEC TR 19759, 2004.

AZAMBUJA, D.. **Teoria geral do estado**. 4 ed. São Paulo: Globo, 2008.

BEALEY, F.. **The Blackwell dictionary of political science**: a user's guide to its terms. Oxford: Wiley-Blackwell, 1999.

BRASIL. **Auditoria do sistema nacional de integração de informações em justiça e segurança pública : Infoseg**. Brasília, Relatório Técnico do Tribunal de Contas da União – TCU, Relator Auditor Augusto Sherman Cavalcanti, 2007.

BRASIL. **Auditoria do sistema informatizado de controle de Óbitos (Sisobi)**. Brasília, Relatório Técnico do Tribunal de Contas da União – TCU, Relator Ministro Augusto Nardes, 2010.

BRESCIANI, P.; DONZELLI, P.; FORTE, A.. Requirements engineering for knowledge management in eGovernment. In: 4th IFIP INTERNATIONAL WORKING CONFERENCE ON KNOWLEDGE MANAGEMENT IN ELECTRONIC GOVERNMENT, KMGov'03, **Anais**. Rhodes, Grécia, p.48-59, 2003.

BUTLER, B. R. R.; BLAIR, P. K.; FOX, A. J.; HALL, I. A. M.; HENRY, K. N.; MCDERMID, J. A.; PARNABY, J., SILLEM, H.; RODD, M.. **The challenges of complex IT projects**. Relatório técnico. Londres: The Royal Academy of Engineering, 2004.

FLINT, C.. **Introduction to geopolitics**. Nova Iorque : Routledge, 2006.

GAULD, R.. Public sector information system project failures: Lessons from a New Zealand hospital organization. **Government Information Quarterly**, v.24, n.1, p.102–114, 2007. DOI: <http://dx.doi.org/10.1016/j.giq.2006.02.010>

HAGUETTE, T. M. F.. **Metodologias qualitativas na Sociologia**. 3 ed. Petrópolis: Vozes, 1992.

HAN, W.; HUANG, S.. An empirical analysis of risk components and performance on software projects. **J. Syst. Softw.**, Nova Iorque, v.80, n.1, p.42–50, 2007. DOI: <http://dx.doi.org/10.1016/j.jss.2006.04.030>

HARA, Y.; FURUTANI, T.; MURAKAMI, A.; PALIJON, A. M.; YOKOHARI, M.. Current organic waste recycling and the potential for local recycling through urban agriculture in metro manila. **Waste Management and Research**, v.29, n.11, p.1213-1221, 2010. DOI: <http://dx.doi.org/10.1177/0734242X10386638>

HÖFLING, E. D. M.. Estado e políticas (Públicas) sociais. **Cadernos CEDES**, v.21, p.30-41, 2001. DOI: <http://dx.doi.org/10.1590/S0101-32622001000300003>

ISO/IEC. **Systems and Software Engineering**: Software Life Cycle Processes (ISO/IEC 12207:2008), Gênova : International Organization for Standardization, 2008.

LEFFINGWELL, D.; WIDRIG, D.. **Managing software requirements**: a use case approach. , 2 ed. Boston: Pearson Education, 2003.

LI, J.; COCKERILL, T.; LIANG, X.; GIBBINS, J.. Locating new coal-fired power plants with Carbon Capture Ready Design – a GIS case study of Guangdong province in China. **Energy Procedia**, v.4, p.2824-2830, 2011. DOI: <http://dx.doi.org/10.1016/j.egypro.2011.02.187>

MARTINS, J. C. C.. **Técnicas para o Gerenciamento de Projetos de Software**. Rio de Janeiro: Brasport, 2007.

OLIVEIRA, D. A.. Das políticas de governo à política de estado: Reflexões sobre a atual agenda educacional brasileira. **Educação & Sociedade**, v.32, p.323-337, 2011. DOI: <http://dx.doi.org/10.1590/S0101-73302011000200005>

PFLEEGER, S. L.; ATLEE, J. M.. **Software engineering**: theory and practice. 4 ed., Nova Jersey : Prentice Hall, 2010.

REINO UNIDO. **Government IT projects**: relatório técnico, n.200. Londres: Parliamentary Office of Science and Technology, 2003.

SEI. **CMMI for Development**, Version 1.3 (CMU/SEI-2010-TR-033). Pittsburgh: Software Engineering Institute, 2010.

SIRAJUL ISLAM, M.; GRÖNLUND, Å.. The Bangladesh national biometric database: A transferable success? In: ELECTRONIC GOVERNMENT AND THE INFORMATION SYSTEMS PERSPECTIVE, **Anais**, Bilbao, Espanha : EGOVIS'10, p.189–203, 2010. DOI: [http://dx.doi.org/10.1007/978-3-642-15172-9\\_19](http://dx.doi.org/10.1007/978-3-642-15172-9_19)

SOFTEX. **Melhoria de processos do software brasileiro** (MPS.BR). Campinas: Softex, 2012.

SOMMERVILLE; I.. **Software Engineering**. 9 ed. Boston : Addison-Wesley, 2010.

VISCUSI, G.; MECCELLA, M.; BATINI, C.. **Information systems for egovernment**: a quality-of-Service perspective. Berlim: Springer, 2010. DOI: <http://dx.doi.org/10.1007/978-3-642-13571-2>

WOOD, J.; SILVER, D.. **Joint application development**. 2 ed. Nova Iorque: John Wiley & Sons, Inc., 1995.