

Tornando a Legislação Transparente para o Engenheiro de Requisitos

Priscila Engiel, Julio Cesar Sampaio do Prado Leite

¹ Departamento de Informática – PUC-Rio
R. Marquês de São Vicente, 225 – Gávea, 22451-900, Rio de Janeiro – RJ – Brasil
{pengiel, julio}@inf.puc-rio.br

Abstract. *Laws and regulations impact the design of software systems, as they may introduce additional requirements and possible conflicts. Laws are inherently voluminous, consisting of hundreds of pages of text and written in legal language¹, being difficult to the requirement engineering to understand what is written. This research proposes a process that aims to help to turn transparent the regulatory requirements to the requirement engineering. With more transparent legal requirements, it will be easy to develop complaint software.*

Resumo. *Leis e regularizações impactam o projeto de sistemas de informação, uma vez que estas podem introduzir novos requisitos e possíveis conflitos. Leis são volumosas, consistem em centenas de páginas de texto escritas em linguagem legal, sendo difícil para o engenheiro de requisitos o seu entendimento. Este trabalho propõe um processo que ajuda a tornar os requisitos regulatórios mais transparentes para o engenheiro de requisitos contribuindo para o desenvolvimento de um software em conformidade com a legislação.*

1. Introdução

Grandes empresas, principalmente bancos e instituições públicas, estão descobrindo constantemente violações relacionadas a *requisitos regulatórios*, o que resulta em penalidades, clientes não satisfeitos e em alguns casos multas legais (Otto and Anton, 2007). Antigamente, conformidade legal era responsabilidade dos auditores. Com o crescimento do uso de produtos de software para apoiar a maior parte das atividades, a conformidade legal passa a impactar o desenvolvimento de software.

Um processo para conformidade legal elícita, analisa e especifica *requisitos regulatórios*, garantindo consistência com os demais requisitos do software. Na prática, esse processo pode ser extremamente difícil e árduo (Nekvi and Madhavji, 2015): os requisitos regulatórios podem ser encontrados em diferentes fontes de informação (mais de uma lei pode influenciar o software, além de existirem diferentes níveis de leis); além disso, leis são escritas de uma forma abstrata e com uma linguagem mais robusta sendo necessário um especialista interpretar a lei para que possa ser traduzida em requisitos;

¹ https://en.wikipedia.org/wiki/Legal_English

por fim leis são atualizadas constantemente, tornando difícil para as organizações manterem o rastro dos requisitos regulatórios.

O primeiro desafio é: como descobrir os requisitos regulatórios que impactam o software em textos volumosos? Resolvido esse desafio, como comparar esses requisitos com os requisitos do software para verificar a conformidade se eles estão escritos em linguagens completamente diferentes? Por fim, para aumentar o desafio, o software pode evoluir e a legislação pode ser alterada (ou novas legislações que impactam o software podem ser criadas). Como garantir que os requisitos continuam em conformidade?

2. Objetivos da Pesquisa

O objetivo dessa pesquisa é propor um processo contínuo que garanta a conformidade entre a lei e o software. A proposta é ancorada no pensamento de que o processo de conformidade legal é uma tarefa para engenheiros de requisitos e que deve ser tratada no nível dos requisitos. Propomos uma forma de extrair os requisitos da lei usando uma ferramenta de modelagem (NomosT) e uma ferramenta de requisitos (Desiree) para o refinamento e análise dos requisitos. Acreditamos que dessa forma diminuimos a complexidade do entendimento do texto legal tornando os *requisitos regulatórios* mais transparentes assim como os conflitos gerados. O objetivo deste trabalho é criar uma ponte entre os engenheiros de requisitos e a legislação.

3. Contribuições Esperadas

A principal contribuição desse trabalho é o processo que tem o apoio de ferramentas para ajudar a extração dos requisitos legais (NomosT) e realizar a comparação com os requisitos de software verificando os possíveis conflitos (Desiree). Neste processo, o objetivo que a extração do texto da lei já seja em um formato de requisitos facilitando assim a leitura e a comparação com os requisitos de software. Ao final do processo é esperada uma nova lista de requisitos onde: (a) alguns requisitos continuaram na lista, (b) alguns requisitos de software serão excluídos devido aos conflitos com os requisitos regulatórios e (c) novos requisitos serão inseridos. O processo contribui para uma melhor lista de requisitos.

4. Resultados já alcançados

Foi proposto um processo (Figura 1) que ajuda a extrair os requisitos da lei, compará-los com os requisitos de software e resolver os possíveis conflitos, resultando em uma especificação formal, consistente e conforme. Algumas atividades no processo são suportadas por ferramentas (Busca dos requisitos, comparação dos requisitos e resolução de conflitos) e o processo tem como objetivo planejar mudanças no software para que este seja conforme com a legislação. Além disso, tem como objetivo deixar transparente a influência da legislação no software em questão.

Para *buscar requisitos na lei*, é utilizada a ferramenta NomosT (Zeni et al, 2016) que suporta a geração semi-automática de modelos da lei expressado na linguagem de modelagem Nomos (Ingolfo et al, 2013). São utilizadas palavras chaves e o modelo conceitual do Nomos, baseado em direitos e obrigações, para realizar a extração dos

requisitos. A ferramenta minera o texto e retorna um texto XML com anotações sintáticas.

Após a extração dos requisitos é necessário *compará-los com os requisitos* de software, porém é necessário ter uma linguagem comum para realizar essa comparação (*Classificar requisitos e Escrever requisitos*). Para isso foi utilizado o Desiree Framework (Li et al, 2015) que transforma os requisitos dos stakeholders em um conjunto mais robusto de requisitos. O framework possui uma ontologia para classificação dos requisitos e um padrão de escrita para cada elemento da ontologia. Neste momento os requisitos do software e da lei são escritos em um mesmo padrão, contribuindo para a *uniformidade*. Além disso, a linguagem utilizada é uma linguagem mais amigável para o engenheiro de requisitos do que a linguagem legal, sendo mais fácil o *entendimento* das restrições impostas pela lei e deixando mais *claro* os requisitos legais.

O ponto central do framework é um conjunto de operadores que ajudam a enfraquecer ou fortalecer requisitos, reduzindo a falta de informações, removendo ambigüidades e contribuindo para *resolver os conflitos*. Na resolução de conflitos, novos requisitos podem ser adicionados, requisitos de software podem ser excluídos ou requisitos de software podem ser modificados para ficarem conformes com a legislação. Todo o *rastro* é mantido na ferramenta de modelagem Desiree.

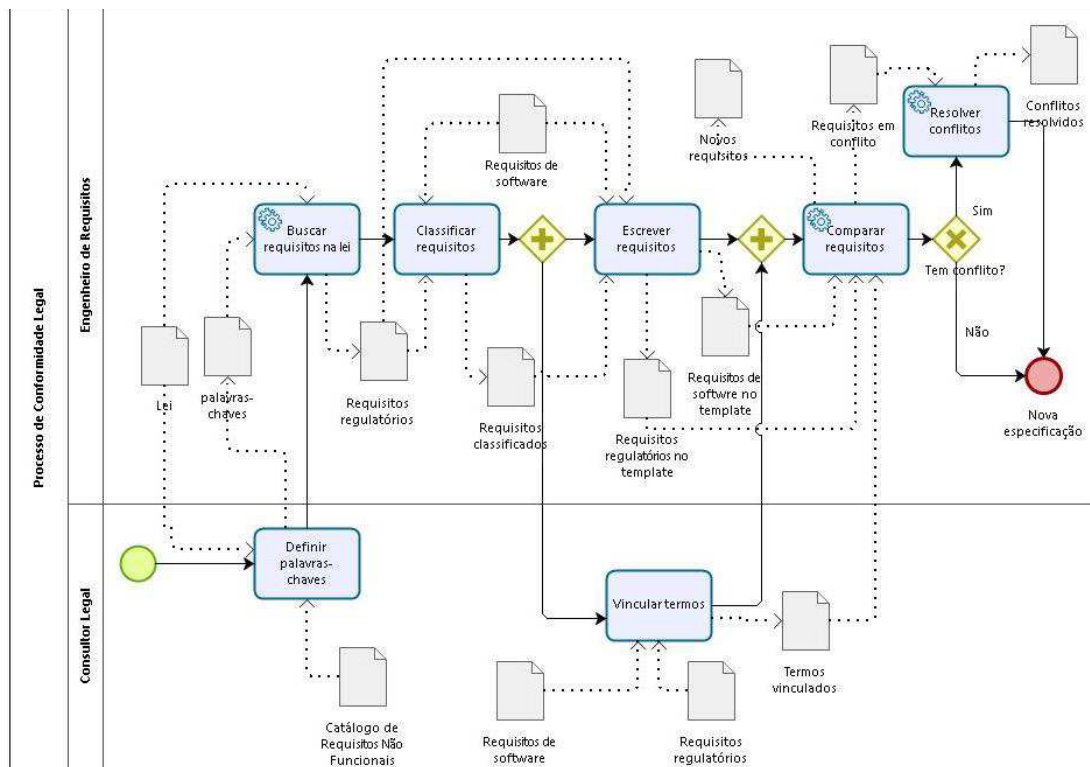


Figura 1 Processo de Conformidade Legal

Um estudo de caso foi conduzido para avaliar o processo proposto, comparando ele com um processo ad-hoc. Para realizar essa avaliação, foi mensurado o tempo gasto para

realizar as tarefas e a qualidade dos requisitos encontrados e da resolução de conflitos comparando com o *gold-standard*.

No final do estudo, os participantes responderam a um questionário que tinha como objetivo verificar os pontos fortes e fracos do método, as impressões, sugestões e as principais dificuldades. Os dois grupos disseram que a maior dificuldade encontrada era em entender o texto da lei, já que eles não eram acostumados com esse tipo de linguagem. O participante 5 respondeu quando perguntado que a maior dificuldade encontrada foi “Ler a lei, a linguagem da lei é bem diferente da linguagem que estou acostumado”. O grupo que utilizou o método, disse que a comparação e a extração eram fáceis, porém, sentiram dificuldade em colocar o texto da lei no template dos requisitos. Mesmo isso sendo uma parte difícil, não impediu que eles completassem a tarefa e encontrassem um número maior de requisitos que o grupo sem método.

A Tabela 1 apresenta os resultados do estudo. O grupo A não teve apoio do método. Podemos observar que em relação ao tempo, o grupo que não utilizou o método teve um resultado melhor. Porém, devemos observar que no estudo de caso utilizamos apenas um fragmento de 7 páginas da lei. Além disso quando observamos os resultados com relação a descoberta de novos requisitos verificamos que este grupo foi bem inferior ao grupo que utilizou o método,. Com relação a descoberta dos requisitos em conflitos, não vemos uma grande diferença do uso do método. Observando os novos requisitos gerados sem o método, não encontramos requisito não funcional algum e os novos requisitos normalmente foram encontrados no momento da resolução de conflitos, sendo requisitos que precisavam ser incluídos para complementar os requisitos de software garantindo a conformidade. Outra questão foi na resolução de conflitos, usando o método requisitos de software foram excluídos e modificados; sem o método requisitos de software foram excluídos e novos requisitos foram criados, não havendo alteração nos requisitos existentes.

tabela 1 – Resultados estudo de caso

Participante	Suportado pelo Processo	Novos requisitos	Requisitos com Conflito	Tempo (em minutos)
Participate1 (grupo A)	Não	4/14	2/2	68
Participate2(grupo A)	Não	7/14	1/2	61
Participate3(grupo B)	Sim	11/14	2/2	73
Participate4(grupo B)	Sim	12/14	2/2	84
Participate 5(grupo B)	Sim	14/14	2/2	69

O processo ainda precisa de melhorias, faltam padrões que ajudem o mapeamento entre o texto da lei e a notação do Desiree. Este estudo comprovou a dificuldade no entendimento das leis por engenheiros de requisitos, porém mostrou que o método já auxilia a tornar esses requisitos mais transparentes. Ainda é necessário um melhor ferramental para extrair os requisitos da lei. Uma possível solução é procurar por patterns (Suppakul 2010) ao invés do uso de palavras chaves. Uma melhor extração e o

mapeamento automático farão com que os requisitos se tornem ainda mais transparentes para o engenheiro de requisitos, pois ele não deixará de lidar com a linguagem legal em nenhuma parte do método.

6. Conclusão

Este artigo apresentou uma forma de tornar a lei mais transparente para os engenheiros de requisitos. Relacionando com o SIG de transparência (Cappelli, 2009), o método propõe que a lei seja escrita de uma forma mais *amigável* para o engenheiro de requisitos, usando um formato *simples* para a escrita dos requisitos. O uso desse formato também contribui para a *uniformidade*, uma vez que os requisitos da lei e do software serão escritos da mesma forma, tornando assim a lei mais *clara* e apresentando de maneira *inteligível* os requisitos legais. A ideia é que a especificação final permita uma *rastreabilidade* na influência que os requisitos regulatórios possuem sobre a especificação do software. A inclusão dos requisitos regulatórios na especificação permite que esta seja mais *completa*. Ainda são necessárias melhorias no método, assim como avaliações em outros ambientes com o texto completo das leis.

Da mesma forma que podemos “traduzir” a lei para uma linguagem que pode ser decodificada por engenheiros de requisitos, poderíamos pensar em uma forma de traduzi-la para os cidadãos. Atualmente, as leis são escritas em uma linguagem formal e específica, não sendo transparentes para a sociedade. Uma vez que todos pudessem entender seus direitos e obrigações, provavelmente, seria mais fácil possuir uma sociedade mais participativa e democrática.

Referências

- Nekvi, M.R.I., Madhavji, N.H. (2015) Impediments to regulatory compliance of requirements in contractual systems engineering projects: a case study. ACM Trans. Manag. Inf. Syst. 5(3),15:1–15:35
- Otto P.N. and Antón A. I (2007) Addressing legal requirements in requirements engineering. In Proceedings of the IEEE International Requirements Engineering Conference (RE'07). IEEE, Los Alamitos, CA, 5–14.DOI:10.1109/ RE.2007.65
- Zeni, N. ; Seid, E. ; Engiel, P. ; Ingolfo, S. ; Mylopoulos, J. (2016). Building Large Models of Law with NomosT. In: The 35th International Conference on Conceptual Modeling
- Ingolfo S. , Siena A., Jureta I., Susi A., Perini A., and Mylopoulos J. (2013) Choosing Compliance solutions through stakeholder preferences. In REFSQ'13. pp 206-220. Springer
- Li F., Horkoff J., Borgida A., Guizzardi G., Liu L.,and Mylopoulos J (2015) :From Stakeholder Requirements to Formal Specifications Through Refinement. In REFSQ'15, pp 164–180. Springer
- Supakkul S., Hill T., Chung L., Tun T.T., Prado Leite J.C.S. (2010)An NFR pattern approach to dealing with NFRs, In18th IEEE International Requirements Engineering Conference, 179-188

Cappelli, C. (2009) “Uma Abordagem para Transparência em Processos Organizacionais Utilizando Aspectos”. Rio de Janeiro, 2009. 328 p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro