

Documentando requisitos de sistemas legados: um estudo de caso utilizando técnicas da Engenharia de Requisitos Orientada a Objetivos

Renato G. Fritola¹, Victor F. A. Santander¹

¹Universidade Estadual do Oeste do Paraná (UNIOESTE)
Caixa Postal 711 — 85.814-110 – Cascavel – PR – Brazil

{renato.fritola,victor.santander}@unioeste.br

Abstract. *Legacy systems are present in many organizations and are often responsible for managing their main and most valuable data. Maintenance is often very expensive and risky due to lack of documentation, unstructured code, and obsolete technology. This work addresses the issue of requirements documentation for these legacy systems via the use of techniques proposed by GORE (Goal Oriented Requirements Engineering), prioritizing the context where the use of source code is not viable. These techniques have the advantage of focusing on the goals and other intentions that stakeholders have in organizational environments, in this case, environments in which legacy systems work. In general, the work seeks to contribute to the dissemination of GORE techniques in the industrial environment, showing how the generated artifacts can contribute to meeting the requirements and showing that it is possible to extract data in situations where the use of code is not feasible.*

Resumo. *Os sistemas legados estão presentes em muitas organizações e muitas vezes são responsáveis por gerir seus principais e mais valiosos dados. Normalmente, a manutenção é muito cara e arriscada devido à falta de documentação, códigos não estruturados e tecnologia obsoleta. Este trabalho aborda a temática de documentação de requisitos desses sistemas legados via utilização de técnicas propostas pela GORE (Goal Oriented Requirements Engineering), priorizando o contexto onde o uso do código-fonte não é viável. Estas técnicas têm a vantagem de focar nos objetivos e outras intenções que stakeholders possuem em ambientes organizacionais, neste caso, ambientes nos quais sistemas legados funcionam. Em geral, o trabalho busca contribuir com a disseminação de técnicas da GORE no ambiente industrial, mostrando como os artefatos gerados podem contribuir no atendimento dos requisitos e mostrar que é possível extrair dados em situações onde a utilização do código não é viável.*

1. Introdução

Atualmente as organizações vêm investindo cada vez mais em tecnologia buscando melhor gerenciamento, ganho de produtividade e competitividade nos negócios. A disseminação do termo Transformação Digital, o surgimento de Tecnologias como IOT, Inteligência Artificial, Blockchain, BigData, serviços em nuvem e de novas normas como a LGPD (Lei Geral de Proteção de Dados) [1] fizeram com que elas buscassem

a atualização do seu ambiente de T.I. Nesta busca, existe um grande complicador, o Sistema Legado. Esses sistemas geralmente estão presentes nas organizações há vários anos, sendo responsáveis pela condução de seus principais processos. Além de obsoletos, são caros, possuem difícil manutenção, não são flexíveis e têm documentação escassa. Por vezes, as regras de negócio estão na mente de pessoas que já não fazem mais parte da organização [2]. É comum encontrar relatos de utilização do código-fonte do sistema legado na reengenharia [3], processo para aprimorar um software existente para que possa ser utilizado como um novo software [4] [5] [6], porém existem alguns complicadores, como o alto custo para a análise do código e até mesmo a indisponibilidade do fonte, o que pode tornar esse processo inviável, fazendo com que devam ser avaliados outros meios para a coleta destes requisitos.

Observando este contexto, percebe-se que o primeiro passo para evoluir ou manter sistemas legados, diminuindo os problemas já mencionados é documentar os requisitos desses sistemas. Para esse fim, técnicas de engenharia de requisitos são recomendadas. Mais especificamente, técnicas da engenharia de requisitos orientada a objetivos (*Goal Oriented Requirements Engineering - GORE*) como i^* [2] e NFR (*Non-functional Requirements*) Framework [7] podem auxiliar a documentar requisitos organizacionais e não funcionais, respectivamente. Essas técnicas têm o foco na representação dos objetivos dos *stakeholders*, o que facilita o processo de elicitação e documentação dos requisitos. Visando mostrar os benefícios do uso dessas técnicas no processo de documentação de sistemas legados, neste trabalho elas são utilizadas em um problema de uma organização real. Relatos de uso dessas técnicas na indústria são importantes para incentivar a utilização das mesmas bem como validar sua utilização por profissionais que enfrentam o desafio de evoluir esses sistemas. Neste sentido, vale ressaltar que um dos autores deste artigo é analista de sistemas na organização em que o estudo de caso foi realizado. Para sistematizar o processo de documentação de sistemas legados usando as técnicas i^* e NFR Framework, alguns passos e diretrizes foram definidos, os quais integram a primeira versão da nossa proposta. Cabe ressaltar na proposta, a identificação, com cores diferentes, de elementos das técnicas visando representar aspectos importantes dos sistemas legados tais como requisitos ainda não implementados ou implementados de forma insatisfatória. Para elaborar os modelos de requisitos, utiliza-se uma adaptação do *template* de entrevista proposto em [8]. Também é importante salientar que no estudo de caso não foi utilizado o código-fonte como base dos requisitos, pois este está muito mal estruturado, sem comentários e existem vários erros que precisam ser corrigidos.

Este artigo está estruturado conforme segue: a seção 2 consiste em um breve referencial teórico, na seção 3, a proposta de documentação de requisitos é apresentada e na seção 4, a mesma é aplicada a um estudo de caso. Na seção 5 estão descritos os trabalhos relacionados e na seção 6 são realizadas as considerações finais do trabalho.

2. Referencial Teórico

A seguir, apresenta-se um breve resumo dos elementos envolvidos na nossa proposta.

2.1. Sistemas Legados

Sistemas Legados referem-se a sistemas antigos que permanecem em operação na organização. Normalmente, aplicações complexas, de difícil manutenção, com alto grau

de criticidade que continuam ativas [9] e muitas vezes são responsáveis pela condução dos principais processos da organização [10]. Dentre os problemas podemos destacar: pouca ou nenhuma documentação, tecnologia obsoleta e sem suporte, alto custo e dificuldade na manutenção, ligado a rotinas críticas do negócio, alta complexidade de código, problemas com a estruturação do código e *backlog* de solicitações de mudança.

2.2. Entrevista

A técnica de entrevista é uma das principais, mais simples e tradicionais técnicas utilizadas no processo de elicitação de requisitos [11]. Trata-se de uma conversa entre o entrevistador e o(s) entrevistado(s) a fim de coletar dados sobre o objetivo desejado. As entrevistas são uma eficiente maneira para coletar uma grande quantidade de dados sobre o assunto desejado. O sucesso dos resultados da entrevista depende muito da qualidade de interação entre os participantes e da habilidade do entrevistador em conduzi-la [12].

2.3. i* (iStar)

A técnica iStar é uma técnica orientada a objetivos introduzida para preencher a lacuna no espectro das linguagens de modelagem conceitual com foco em dimensão intencional (por quê?), social (quem?) e estratégica (como?) [2]. A técnica propõe dois modelos para descrição dos aspectos, sendo eles: Modelo de Dependências Estratégicas (SD) e o Modelo de Razões Estratégicas (SR). Como proposta para apoio à documentação do sistema legado será utilizado o modelo SR de modo a expressar os objetivos dentro da organização, tarefas, funções e quais são as dependências e relações do sistema com os atores no âmbito da organização. Além de oferecer um ótimo entendimento sobre a relação do sistema e os seus aspectos organizacionais, é possível, utilizando ferramenta denominada JGOOSE [13], realizar a derivação de Casos de Uso a partir desses modelos.

2.4. NFR Framework

O NFR Framework é uma abordagem orientada a processos que trata os Requisitos Não Funcionais (NFRs) como metas (*softgoals*) a serem alcançadas durante o processo de desenvolvimento do sistema. Representa essas metas em um gráfico denominado de *Soft-goal Interdependency Graph (SIG)*. Ao longo do processo, as metas são decompostas, as compensações de design são analisadas, as decisões de design são racionalizadas e o cumprimento das metas é avaliado [2]. O uso do NFR Framework pode trazer benefícios como: visualização dos relacionamentos e interdependências entre requisitos; é possível visualizar o impacto que um NFR gera sobre outro; auxilia na identificação de conflitos; auxilia na representação, organização e análise dos NFRs; permite decompor os requisitos facilitando a identificação das prioridades e necessidades dos *stakeholders*.

Para a proposta deste artigo, o NFR Framework é utilizado para apoiar o resgate dos requisitos não funcionais que foram ou deveriam ter sido atendidos com base nos relatos do entrevistado. Além disso, irá apoiar a descoberta de NFRs, que para a época em que o sistema foi desenvolvido, não foram identificados e são de grande importância para uma futura evolução ou migração do software. Na seção 3, apresenta-se na figura 3 o modelo gerado para o estudo de caso realizado.

3. Proposta

Muitas vezes na criação de um sistema, a pressão por cumprimento de prazos e redução de custos faz com que a documentação seja deixada em segundo plano. Tem-se a pre-

ferência por produzir o sistema e se houver tempo no projeto, documentá-lo. Ocorre que, em decorrência das operações de rotina e outros projetos, nunca existe tempo para documentação e assim o sistema segue evoluindo deixando-a para trás. A necessidade da documentação começa a surgir a partir do momento em que é preciso realizar uma manutenção e o responsável pela manutenção não participou da criação do software. A documentação não deve ser vista com um esforço desnecessário no projeto e sim deve fazer parte de suas etapas, pois é essencial para a boa construção de um sistema e sua existência pode contribuir diretamente no sucesso do sistema [14]. Particularmente, a documentação de requisitos deve incluir informações essenciais de modo que um novo colaborador consiga entender e dar sequência na evolução do sistema [15]. Essa documentação bem elaborada e atualizada, auxilia não somente a equipe técnica, mas também usuários proporcionando melhor compreensão de uso do sistema. Além disso, traz outros benefícios como facilidade de evolução, redução de retrabalho e erros após o desenvolvimento, proporciona um ambiente colaborativo, maior conformidade com os requisitos do usuário e aumento da qualidade do software. Desta forma, as três técnicas de engenharia de requisitos descritas na seção 2, podem contribuir para o início da confecção da documentação de requisitos de um sistema legado, entendendo os seus objetivos na organização assim como suas funcionalidades existentes e possíveis vulnerabilidades devido à falta de evolução. A figura 1 apresenta um resumo do processo proposto para documentação de sistemas legados.

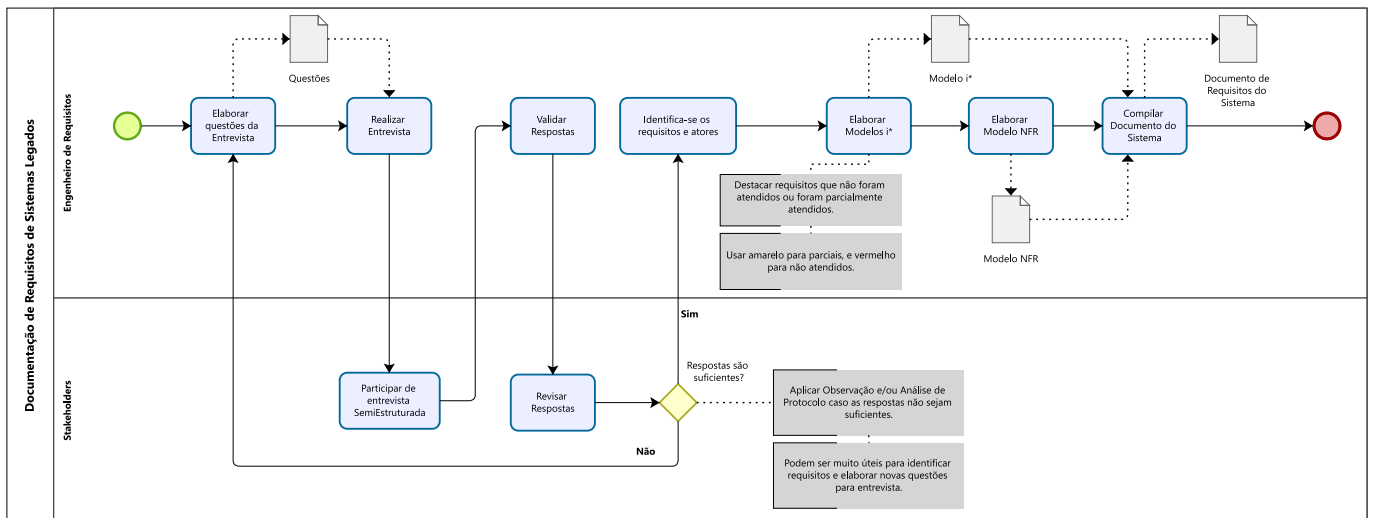


Figura 1. Passos da Proposta

Nesta figura, podemos observar que a atividade inicial do processo é a elaboração das questões da entrevista adaptadas de [8]. Na sequência, a entrevista é realizada bem como se permite que o entrevistado expresse livremente os aspectos do sistema considerados importantes. Vale ressaltar que a fim de ter questões e respostas mais objetivas, outras técnicas de elicitación como Observação e/ou Análise de Protocolo podem ser adotadas. Em seguida, a elicitación dos requisitos é realizada baseada nos resultados da entrevista. Após identificação dos requisitos e atores, inicia-se a documentação desses requisitos via técnicas i* e NFR Framework. Neste sentido, cabe destacar que são utilizadas propostas de extensões ao framework i*, as quais permitem a identificação visual através de coloração diferenciada de requisitos de sistemas legados atendidos parcialmente, através

da coloração amarela ou não foram atendidos, através da coloração vermelha. O processo se finaliza com a compilação de um documento de requisitos do sistema, unindo as informações obtidas nos passos anteriores.

4. Estudo de Caso

Para o estudo de caso, foi selecionado um sistema legado responsável pela disponibilização e registro de execução de provas de processos seletivos de estágio de uma organização no ramo da indústria farmacêutica (o nome não pode ser citado devido às políticas de segurança da organização). Trata-se de um sistema desenvolvido em 2016 usando linguagem Java versão 1.6 com Java EE 6, JSF 2.1 e Primefaces 3.4. Assim como muitos sistemas legados, esse sistema foi concebido sem nenhuma documentação prévia. Com o tempo foram efetuadas melhorias e manutenções ainda sem a existência de documentação tornando o sistema mais complexo e difícil de manter. O Sistema atende seu principal objetivo que é computar a nota das provas realizadas no processo seletivo, porém com muitas atividades e funções engessadas, outras não utilizadas e operações manuais sendo realizadas durante o processo. Além disso, devido à versão antiga da tecnologia, algumas versões de navegadores não o suportam mais, fazendo com que exista uma demanda considerável de chamados durante a realização de processos de avaliação. Para obter informações sobre objetivo, funções e problemas atuais do sistema, foi realizada uma entrevista com o usuário chave. Este usuário coordena todo o processo de disponibilização e registro de execução de provas de processos seletivos de estágio. Desta forma, foram primeiramente realizadas quatro questões para direcionamento e após isso ficou em aberto para que o usuário pudesse fazer mais considerações sobre o sistema legado. As questões do questionário foram adaptadas da proposta apresentada em [8]. Por aspectos de espaço, esse questionário está disponível em [16].

Com base nos dados obtidos na entrevista, os próximos passos foram gerar o modelo *i** (iStar) e o NFR Framework. Para ambos os modelos, foram utilizadas ferramentas do pacote DSM3-goals [17]. Após a entrevista foi possível gerar o modelo *i** identificando os objetivos do sistema conforme figura 2. Foram identificados sete atores principais nesse processo, sendo a Organização, Analista de TI e Avaliador (especialização do ator Colaborador), Sistema de Estágios, Sistemas de Banco de Talentos e o Candidato. Os requisitos macro atendidos atualmente pelo sistema e mapeados através dos objetivos do modelo foram: Manter programa de estágio, Manter fases do programa, Manter provas do programa, Manter questões da prova, Manter alternativas de questões, Configurar tempos de execução das questões, Manter Candidatos, Gerenciar Candidatos, Registrar Execução das Provas. Além disso, conforme a figura 2, com esse levantamento pode-se identificar requisitos solicitados na concepção do sistema, porém por falta de definição ou não serem tratados como uma prioridade na época, não foram atendidos ou foram atendidos de maneira ineficiente. Por exemplo, foram identificadas em tonalidade amarela na figura 2, as tarefas realizadas manualmente hoje devido à satisfação ineficiente desses requisitos. O uso de cores para essas tarefas bem como outros elementos nos modelos *i** são uma adaptação do trabalho apresentado em [15].

Segue uma descrição destas tarefas.

- Migrar Currículos: Conforme relato, o analista de TI da empresa é responsável por fazer a transferência dos dados entre o sistema de maneira totalmente manual,

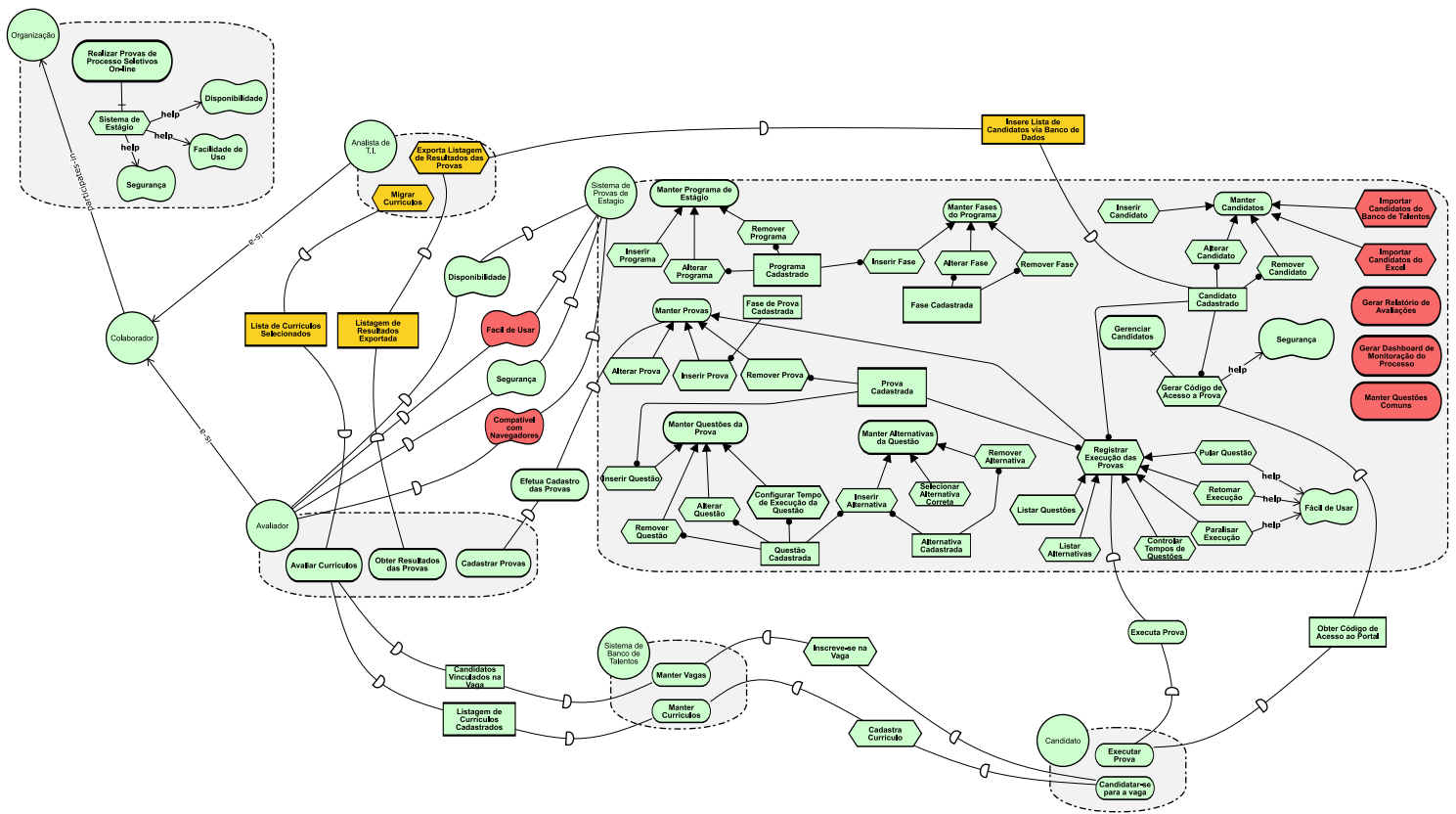


Figura 2. Modelo i* (iStar) gerado com base nos requisitos obtidos na entrevista realizada

selecionando dados de um sistema, validando com o usuário avaliador e depois fazendo a inserção desses dados validados em outro.

- Exportar Listagem de Resultados das Provas: Durante o processo de execução das avaliações o analista de TI precisa a cada dia gerar a relação dos candidatos que realizaram a prova e enviar para os avaliadores.

Em vermelho, destacam-se requisitos importantes para o negócio que não são atendidos na versão atual: Importar Candidatos do Banco de Talentos, Importar Candidatos do Excel, Gerar Relatório de Avaliações, Gerar Dashboard de Monitoração do Processo, Manter Questões Comuns. O atendimento destes requisitos eliminaria o trabalho manual listado acima. Além disso, dois requisitos de qualidade que eram atendidos na época, por falta de atualização e manutenção, já não são mais atendidos de maneira satisfatória, são eles: Fácil de Usar e Compatível com Navegadores. Para identificar como os requisitos não funcionais (as metas de qualidade identificadas no iStar) são atendidos, foi gerado o modelo SIG do NFR Framework, conforme figura 3. No modelo foram identificados os seguintes NFR's:

- Segurança, composto pelos 3 NFR's [7]: Disponibilidade (Atendido através de uma VM (virtual machine) exclusiva); Integridade (Atendido através do uso de banco de dados relacional); Confidencialidade (Atendido através de autenticação de usuário).
- Usabilidade: Atendido através de utilização de framework JSF e visão separada para administrador e candidato. Assim como no i*, em vermelho por estar defasado.
- Compatibilidade: Atendido através de utilização de framework JSF e linguagem Java. Em vermelho, pois perdeu compatibilidade por falta de atualização das tecnologias utilizadas.
- Performance: Não houve operacionalização direta para atendimento. Existem

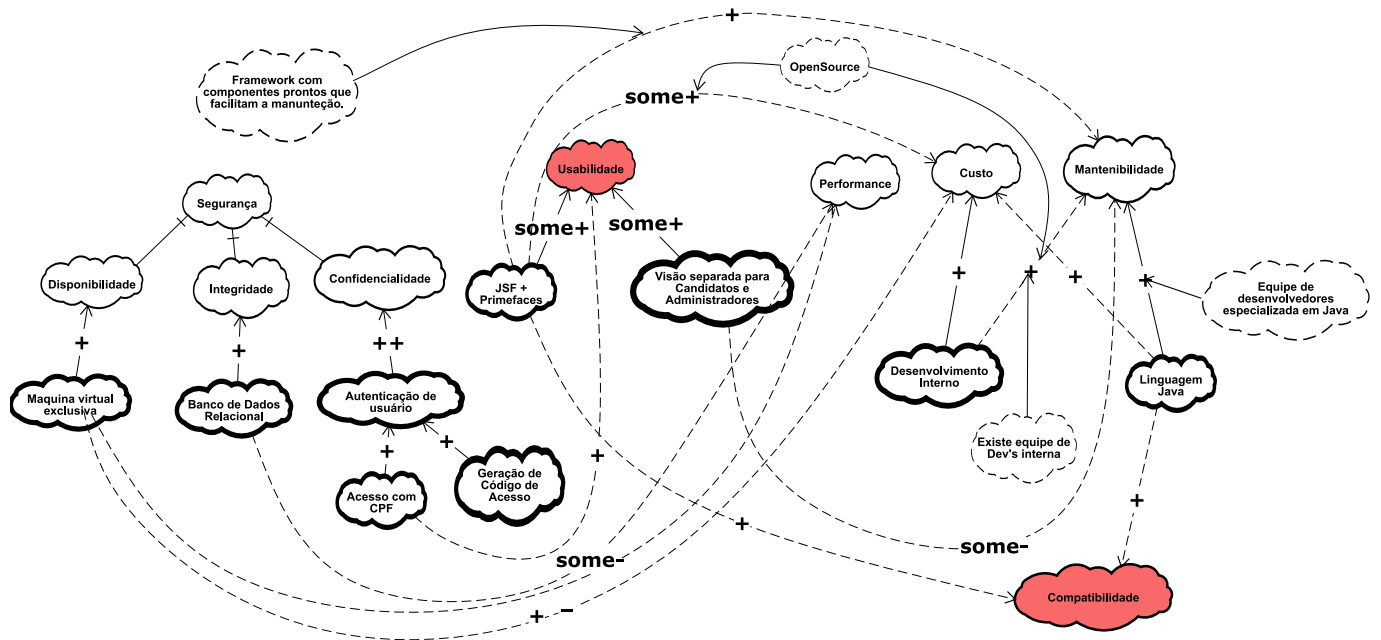


Figura 3. SIG do NFR Framework para o sistema de Provas

duas interdependências sendo uma positiva em relação à VM exclusiva e uma negativa (com menor impacto some-) em relação à utilização de banco de dados relacional.

- Custo: Operacionalizado positivamente através do desenvolvimento interno, reduzindo custos de contratação de terceiros. É impactado negativamente pelo uso de VM exclusiva que aumenta custo para manutenção e positivamente pela utilização de framework de desenvolvimento Opensource.
- Manutenibilidade: Operacionalizado pela escolha da linguagem Java, que possui impacto positivo em relação à equipe de desenvolvedores especializada. Existem também interdependências com este requisito sendo positivas: desenvolvimento interno e utilização do framework JSF com Primefaces e negativa: visão separada para administrador e candidato, pois se geraram aí dois pontos de manutenção

Com o modelo SIG gerado fica mais fácil incluir requisitos não funcionais que atualmente são necessários de acordo com a evolução tecnológica e leis como LGPD [1]. Por exemplo, em uma futura migração fica mais fácil identificar que os tratamentos para Segurança são fracos e precisam ser revistos com prioridade. Alguns exemplos de operacionalizações para serem inclusas em segurança seriam: Utilização de Certificado SSL, Recaptcha, Criptografia de Dados entre outros. O mesmo pode ser aplicado à Usabilidade e Compatibilidade. A tecnologia obsoleta e layout desatualizado fez com que esses requisitos deixassem de ser atendidos por completo.

5. Trabalhos Relacionados

Em [18] propõe-se o uso da técnica i^* para apoiar o processo de evolução de sistemas legados. Os autores baseiam-se na documentação de software existente, mesmo que incompleta, e apresentam diretrizes de derivação de modelos i^* a partir dos DFD's (Diagramas de Fluxo de Dados) disponíveis como documentação do sistema legado. Em [19], os autores apresentam uma proposta de gamificação para as atividades de levantamento de requisitos de substituição de sistemas legados do setor público. Defendem que o uso da gamificação para substituição de legado, tem o potencial de promover a inovação e encorajar a criatividade do profissional durante a análise de requisitos, desde que seja introduzida de forma adequada. Em [20], os autores recuperam as metas a partir dos comentários do código legado e modelam usando o método KAOS. Os próprios autores informam que o método não funciona bem se não existirem comentários bem pensados. Em

[21], propõe-se a extração de requisitos de segurança a partir do código legado, gerando uma extensão do método TROPOS baseada em segurança, o TROPOS *Security*. Neste caso, o foco está apenas nos requisitos de segurança. Em [22], os autores fazem o uso da técnica GBRAM, também orientada a objetivos, para elicitar, revisar e refinar os requisitos de sistema coletados com base na descrição de stakeholders coletadas em entrevistas. Já em [23], utiliza-se o NFR framework para modelar e identificar o deficit associado ao conceito de *trustworthiness* em sistemas legados. Esse conceito remete ao requisito não funcional de Confiabilidade, de forma mais ampla. Também no contexto de uso do código-fonte do sistema legado, o trabalho apresentado em [24] expõe os problemas que podemos encontrar na engenharia reversa de sistemas legados considerando esse artefato como base. Os trabalhos mencionados, diferentemente de nossa proposta, utilizam diferentes fontes para gerar os modelos de requisitos e não consideram o uso complementar de *i** e do NFR Framework. Ainda é possível destacar o trabalho apresentado em [15] que propõe um conjunto de estratégias de visualização para representação da priorização e liberação de requisitos do sistema via técnica *i**. As estratégias de visualização adotadas foram: coloração, tamanho de fonte e tipos de linha diferenciados. Para aplicação da proposta foi desenvolvida uma extensão para a ferramenta piStar. Esta ferramenta e a estratégia de colorização são utilizadas no nosso trabalho.

6. Conclusão

Sistemas legados sustentam importantes processos nas organizações e a negligência em relação à documentação acarreta diversos problemas com manutenção, evolução e até a migração do sistema. A utilização das técnicas apresentadas neste trabalho em um problema real, seguindo o processo proposto, permite apontar alguns benefícios. A entrevista realizada com usuários-chave do sistema traz os requisitos com base no relato desses usuários que acumulam experiências de anos de uso. Além de possibilitar a identificação de requisitos existentes no sistema, a entrevista pode mostrar problemas com requisitos atuais e a falta de requisitos importantes, que passam muitas vezes despercebidos durante o processo, pois durante a rotina, os usuários acabam criando outros meios de atendê-los como, por exemplo, utilizando controles paralelos em planilhas como é o caso do sistema utilizado no estudo de caso. O framework *i** exibe de maneira clara os objetivos e tarefas do sistema e a relação com os atores do processo, facilitando a identificação dos relacionamentos, impactos e dependências entre os requisitos do sistema. O modelo SIG do NFR Framework, assim como o *i**, exibe os requisitos não funcionais e como eles foram atendidos no sistema legado. A relação, contribuição e impacto entre eles fica evidente, tornando possível uma melhor avaliação, identificação e outras operacionalizações que podem ser importantes no momento atual e que não foram consideradas no passado durante a concepção do sistema. Desta forma, pode se concluir que as informações geradas com base nos artefatos apresentados podem servir de base para início da documentação do sistema legado e incorporações de melhorias futuras, possibilitando uma visão geral dos processos executados no domínio do mesmo.

Como limitação da proposta é possível observar que a obtenção de requisitos do estudo de caso foi baseada em relatos de um único usuário que atualmente mantém o sistema e no conhecimento do analista (um dos autores do artigo) em relação ao sistema existente. Entrevistar outros *stakeholders* poderia revelar outros requisitos. Também o uso de outras técnicas de elicitação como observação do sistema atual em operação e até

mesmo aplicar questionários direcionados a outros usuários do sistema, poderia melhorar a completude e validação dos modelos gerados, o que não foi realizado neste artigo.

Como trabalhos futuros podemos destacar o aprimoramento da proposta, considerando os benefícios iniciais percebidos no dia a dia da indústria, validando os artefatos gerados através de questionários aplicados a equipe técnica. Também pretende-se explorar outras propostas de modelagem oriundas da engenharia de requisitos orientada a objetivos como KAOS [25] e TROPOS [26] bem como outras técnicas de elicitação tais como observação, prototipação, entre outras. Adicionalmente, pretende-se incluir no processo de documentação de sistemas legados, casos de uso UML derivados através da ferramenta JGOOSE bem como outros artefatos UML que possam auxiliar nessa documentação. Outro trabalho futuro envolve investigar novas formas de identificação visual para requisitos funcionais e não funcionais críticos, facilitando o tratamento desses em um processo de evolução de sistemas legados.

Referências

- [1] “Lgpd brasil - lei geral de proteção de dados pessoais.” <https://lgpd-brasil.info/>, 2018. Accessed: 2021-10-26.
- [2] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos, *Social Modeling for Requirements Engineering*. The MIT Press, 2011.
- [3] M. Majthoub, M. Qutqut, and Y. Odeh, “Software re-engineering: An overview,” 07 2018.
- [4] V. Fontanette, V. C. Garcia, A. A. Bossonaro, A. B. Perez, and A. F. do Prado, “Reprojeto de sistemas legados baseado em componentes de software,” in *XXVIII Conferencia Latinoamericana de Informática (InfoUYclei)*, 2002.
- [5] L. L. Chaves, “Sistemas legados e a aplicaomo de processos de reengenharia de software,” 2004.
- [6] “Incubando serviços em sistemas legados para migração arquitetônica,” in *11ª Conferência de Engenharia de Software da Ásia-Pacífico*.
- [7] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*, vol. 5. Springer Science & Business Media, 2012.
- [8] C. F. de Souza and V. F. Santander, “Uma proposta de elicitação e análise de requisitos no contexto de médias e pequenas empresas de desenvolvimento de software.,” in *CibSE*, pp. 285–296, 2011.
- [9] B. S. Hussain, SM and M. U. Rasoll, “Sistema legado e formas de sua evolução,” in *[2017] Conferência Internacional sobre Tecnologias de Comunicação (ComTech)*, pp. 56–59, 2017.
- [10] J. Crotty and I. Horrocks, “Managing legacy system costs: A case study of a meta-assessment model to identify solutions in a large financial services company,” *Applied computing and informatics*, vol. 13, no. 2, pp. 175–183, 2017.
- [11] D. Zowghi and C. Coulin, “Requirements elicitation: A survey of techniques, approaches, and tools,” in *Engineering and managing software requirements*, pp. 19–46, Springer, 2005.

- [12] J. Goguen and C. Linde, “Techniques for requirements elicitation,” in *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, pp. 152–164, 1993.
- [13] G. C. L. Geraldino and V. F. A. Santander, “The jgoose tool,” in *iStar@ ER*, 2019.
- [14] A. Galletta, *Mastering the semi-structured interview and beyond*. New York University Press, 2013.
- [15] J. Pimentel and M. Lencastre, “Would you like better visualization for requirements prioritization and release planning?,” in *WER*, 2020.
- [16] “Questionário aplicado e respondido no estudo de caso..” https://drive.google.com/file/d/1K-R_E3JkAMJ8Qz_oHFAz0ZF72maJl1U/view?usp=sharing. Accessed: 2021-10-26.
- [17] C. UFPE, “Dsm3-goals. pacote de ferramentas..” <https://www.cin.ufpe.br/~jhcp/dsm3goals/index.html>, 2020. Accessed: 2021-10-26.
- [18] V. F. A. Santander, A. A. Vicente, F. Koerich, and J. Castro, “Elicitação de requisitos organizacionais, não-funcionais e funcionais em software legado com ênfase na engenharia de requisitos orientada a objetivos,” in *X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software*. Isla Margarita, Venezuela, 2007.
- [19] A. Alexandrova and L. Rapanotti, “Requirements analysis gamification in legacy system replacement projects,” *Requirements Engineering*, vol. 25, no. 2, pp. 131–151, 2020.
- [20] Y. Yu, Y. Wang, J. Mylopoulos, S. Liaskos, A. Lapouchnian, and J. C. S. do Prado Leite, “Reverse engineering goal models from legacy code,” in *13th IEEE International Conference on Requirements Engineering (RE’05)*, pp. 363–372, IEEE, 2005.
- [21] N. Argyropoulos, L. M. Alcañiz, H. Mouratidis, A. Fish, D. G. Rosado, I. G.-R. de Guzmán, and E. Fernández-Medina, “Eliciting security requirements for business processes of legacy systems,” in *IFIP Working Conference on The Practice of Enterprise Modeling*, pp. 91–107, Springer, 2015.
- [22] A. Anton and C. Potts, “The use of goals to surface requirements for evolving systems,” in *Proceedings of the 20th International Conference on Software Engineering*, pp. 157–166, 1998.
- [23] N. Subramanian, S. Drager, and W. McKeever, “Identifying trustworthiness deficit in legacy systems using the nfr approach,” vol. 27, pp. 4–11, 01 2014.
- [24] B. W. Weide, W. D. Heym, and J. E. Hollingsworth, “Reverse engineering of legacy code exposed,” in *1995 17th International Conference on Software Engineering*, pp. 327–327, 1995.
- [25] A. Dardenne, A. Van Lamsweerde, and S. Fickas, “Goal-directed requirements acquisition,” *Science of computer programming*, vol. 20, no. 1-2, pp. 3–50, 1993.
- [26] J. Castro, M. Kolp, and J. Mylopoulos, “Towards requirements-driven information systems engineering: the tropos project,” *Information systems*, vol. 27, no. 6, pp. 365–389, 2002.