

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/334163476>

Padrão de Requisitos no Ciclo de Vida de Software: Um Mapeamento Sistemático

Conference Paper · April 2019

CITATIONS

2

READS

1,283

4 authors:



Taciana Kudo

Universidade Federal de Goiás

18 PUBLICATIONS 43 CITATIONS

[SEE PROFILE](#)



Renato Bulcão Neto

Universidade Federal de Goiás

89 PUBLICATIONS 332 CITATIONS

[SEE PROFILE](#)



Alessandra Alaniz Macedo

University of São Paulo

88 PUBLICATIONS 400 CITATIONS

[SEE PROFILE](#)



Auri Marcelo Rizzo Vincenzi

Universidade Federal de São Carlos

117 PUBLICATIONS 990 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Semantic-Enabled Services for Context-Aware Computing: Studies in Healthcare [View project](#)



Software Requirement Patterns and the Software Development Life Cycle [View project](#)

Padrão de Requisitos no Ciclo de Vida de Software: Um Mapeamento Sistemático

Taciana N. Kudo^{1,3}, Renato F. Bulcão-Neto¹, Alessandra A. Macedo², and
Auri M. R. Vincenzi³

¹ Instituto de Informática, Universidade Federal de Goiás, Goiânia-GO, Brasil
{taciana,renato}@inf.ufg.br

² Depto. de Computação e Matemática, FFCLRP-USP, Ribeirão Preto-SP, Brasil
ale.alaniz@usp.br

³ Depto. de Computação, UFSCAR, São Carlos-SP, Brasil
auri@dc.ufscar.br

Resumo Pesquisas demonstram que a prática de reúso por meio de padrões de requisitos é uma alternativa eficaz para tratar problemas de qualidade de especificação, com o benefício adicional de economia de tempo. Acredita-se que, em função das interações existentes entre a engenharia de requisitos e demais fases do ciclo de vida de software, esses benefícios se estenderão para todo o processo de desenvolvimento. Este artigo descreve um estudo secundário que identifica e analisa pesquisas que evidenciam tais benefícios com o uso de padrões de requisitos nas fases de projeto, construção, teste e manutenção. Realizou-se um mapeamento sistemático, cujo protocolo inclui busca automática em 4 bases de dados e a aplicação de critérios de inclusão e exclusão sobre 218 estudos, dos quais selecionaram-se 9 estudos primários para análise e síntese. Identificou-se uma carência de estudos relacionados ao objeto desta pesquisa, sendo a maioria relacionada a projeto de software, e nenhum a manutenção. Além disso, apenas um dos estudos analisados caracteriza-se como pesquisa de validação com estudo de caso, enquanto que os demais são propostas de solução, sem avaliação empírica, ou prática. Portanto, existe um campo aberto para pesquisas que explorem e comprovem, com avaliações empíricas e uso na prática, os benefícios de reúso com padrões de requisitos no ciclo de vida de software.

Keywords: Padrão de requisito · ciclo de vida de software · mapeamento sistemático

1 Introdução

Embora seja consenso que a Engenharia de Requisitos é uma atividade de extrema importância, na qual as funcionalidades e restrições do software devem ser bem identificadas e compreendidas, uma alta porcentagem de projetos de software não cumpre prazos e orçamento planejados, devido à documentação incompleta, com requisitos mal interpretados, conflitantes ou omitidos [21,17].

O reúso por meio de padrão de requisitos é uma abordagem que tem sido bastante explorada com o objetivo de melhorar a qualidade da documentação de requisitos de software [7,17]. Um padrão de requisito é uma abstração que reúne comportamentos e serviços de aplicações com características semelhantes, os quais podem ser replicados em documentações futuras. Um padrão de requisito serve também como um modelo para especificação de um novo requisito [24].

Por exemplo, para escrever um requisito de autenticação de usuário, é possível utilizar um padrão de requisito para esse fim, fazendo as devidas adaptações no requisito, se necessário. Na literatura são encontradas várias propostas de padrões de requisitos, por exemplo, para sistemas embarcados [14], sistemas gerenciadores de conteúdo [18] e sistemas de computação em nuvem [3]. Dentre os benefícios obtidos com a adoção de padrões de requisitos, têm-se: maior eficiência na elicitação, pois os requisitos não são identificados do zero; melhoria na qualidade e consistência no documento de especificação; e melhoria no gerenciamento dos requisitos [24].

Devido à interação inerente entre a engenharia de requisitos e as demais fases do ciclo de vida de software, pressupõe-se que os benefícios advindos do uso de padrões de requisitos podem alcançar as outras atividades de desenvolvimento. Embora existam estudos secundários sobre engenharia de software [19], engenharia de requisitos [15] e padrões de requisitos [7,2], não existem evidências em estudos dessa natureza que analisem o uso de padrões de requisitos nas demais fases do processo de desenvolvimento de software. Em suma, os estudos secundários existentes restringem-se a analisar a adoção de padrões de requisitos particularmente na fase de engenharia de requisitos.

Sendo assim, o objetivo deste artigo consiste em apresentar um estudo secundário na forma de um mapeamento sistemático, que identifica e analisa estudos primários que evidenciam o uso de padrões de requisitos nas fases de projeto, construção, teste e manutenção de software⁴.

A metodologia definida apóia-se em trabalhos clássicos de estudos secundários em Engenharia de Software [12,19,8] com suporte da ferramenta *StArt*⁵ para o planejamento e a condução do protocolo do mapeamento sistemático. Buscas automáticas foram realizadas em 4 bases de dados [9] e critérios de inclusão e exclusão foram aplicados para identificar os estudos relevantes.

Como resultado da aplicação do protocolo, foram identificados e analisados 9 estudos relevantes, classificados segundo o tipo de pesquisa realizado e o tipo de contribuição do padrão de requisito nas fases citadas, resultando no mapeamento de lacunas relacionadas a padrões de requisitos no ciclo de vida de software.

Este artigo está organizado como segue: a Seção 2 detalha o protocolo do mapeamento sistemático; a Seção 3 reporta a extração de dados dos estudos relevantes. As respostas às questões de pesquisa deste estudo e as lacunas de pesquisa estão sintetizadas na Seção 4. Por fim, a Seção 5 expõe as ameaças à validade deste estudo sistemático, bem como nossas considerações finais.

⁴ Neste artigo, adotar-se-á a terminologia das fases do ciclo de vida de software proposta no SWEBOK (*Software Engineering Body of Knowledge*) [4].

⁵ Disponível para download em http://lapes.dc.ufscar.br/tools/start_tool.

2 Protocolo do Mapeamento Sistemático

Um processo de estudo sistemático pode ser dividido em três fases principais [9]: planejamento, condução e publicação dos resultados. No planejamento, define-se um protocolo de modo que o estudo possa ser replicado futuramente. São integrantes desse protocolo: questões de pesquisa, estratégia de busca, *string* de busca, fontes de pesquisa e critérios de inclusão e exclusão dos estudos.

Na fase de condução, faz-se a identificação e seleção dos estudos relevantes, segundo os critérios definidos no protocolo. Desses estudos são extraídos dados, a partir dos quais faz-se uma síntese para responder às questões de pesquisa do protocolo. O mapeamento sistemático aqui apresentado seguiu essas fases.

2.1 Questões de Pesquisa e Palavras-chave

O principal objetivo deste mapeamento sistemático é identificar propostas de uso de padrões de requisitos e o seu impacto nas demais fases do ciclo de vida de software, além da engenharia de requisitos. Para atender a esse objetivo, as questões de pesquisa (**QP**) que se deseja responder são:

- QP1.** Em quais fases do ciclo de vida de software são usados padrões de requisitos: projeto, construção, teste e/ou manutenção?
- QP2.** Há evidências de uso prático de padrões de requisitos nessas fases do ciclo de vida de software?
- QP3.** Há benefícios relatados do uso de padrão de requisitos nessas fases? Se sim, quais métricas são utilizadas para medir esses benefícios?

Para responder adequadamente às questões de pesquisa, é necessário seguir uma estratégia de busca apropriada [9]. Para fundamentar a definição de termos padronizados da Engenharia de Software, os termos de pesquisa foram definidos de acordo com o SEVOCAB⁶. Como resultado, chegou-se ao seguinte conjunto de palavras-chave em inglês: *requirement pattern*, *development process*, *software development*, *life cycle*, *design*, *construction*, *coding*, *implementation*, *test*, *integration* e *maintenance*.

2.2 Estratégia de Busca

Com base nas palavras-chave, e em busca de equilíbrio entre abrangência⁷ e relevância dos resultados, foi definida a seguinte *string* de busca:

(“*requirement pattern*” OR “*requirement patterns*” OR
“*requirements pattern*” OR “*requirements patterns*”)
AND ((“*software development*” OR “*development process*”) OR
 (“*life cycle*” OR *design* OR *construction* OR *coding* OR
implementation OR *test* OR *integration* OR *maintenance*))

⁶ *Software and Systems Engineering Vocabulary* é uma iniciativa da ISO/IEEE de padronizar os termos utilizados na área de Engenharia de Software [11]

⁷ Variações em inglês do termo “padrão de requisito” foram usadas em função das capacidades das máquinas de busca das fontes de pesquisa escolhidas.

Foram escolhidas 4 fontes de pesquisa relevantes (*ACM DL*, *Engineering Village*, *IEEE Xplorer* e *Scopus*) sobre as quais foram realizadas buscas sobre os metadados dos estudos. Como resultado, foram identificados 218 estudos⁸, assim distribuídos: *ACM Digital Library* (24), *Engineering Village* (100), *IEEE Xplorer* (23) e *Scopus* (71). No caso da *ACM DL*, a busca foi expandida para *The ACM Guide to Computing Literature* por incluir uma coleção maior de artigos.

2.3 Identificação e Seleção dos Estudos Primários

Esta seção descreve o método utilizado para selecionar, dentre os 218 estudos identificados, quais são relevantes para responder às questões de pesquisa deste mapeamento sistemático. Um conjunto de 7 (sete) critérios de exclusão foi definido e aplicado a esses estudos. Os critérios de exclusão (CE) são:

- CE1** - Não é um estudo primário.
- CE2** - Não é um artigo (p.ex. prefácio/sumário de periódicos/anais de eventos).
- CE3** - Não é relacionado com padrões de requisitos de software.
- CE4** - Aborda padrões de requisitos apenas na fase de engenharia de requisitos.
- CE5** - O texto completo do estudo não está em inglês.
- CE6** - Não foi possível acessar o texto completo do estudo.
- CE7** - É uma versão preliminar ou reduzida de outro estudo (não adiciona informação relevante).

Quando um estudo se encaixa em pelo menos um critério de exclusão, ele está automaticamente excluído do mapeamento. Caso contrário, o estudo deverá ser categorizado em um critério de inclusão (CI):

- CI1** - Aborda padrões de requisitos na fase de projeto de software.
- CI2** - Aborda padrões de requisitos na fase de construção de software.
- CI3** - Aborda padrões de requisitos na fase de teste de software.
- CI4** - Aborda padrões de requisitos na fase de manutenção de software.

O processo completo de seleção dos estudos primários pode ser visualizado na Figura 1, na qual são descritas também as quantidades de estudos selecionados em cada uma das atividades da fase de condução deste mapeamento sistemático.

Após a atividade de busca sobre as fontes de pesquisa, foi realizada a identificação e eliminação de 111 estudos duplicados. Em seguida, iniciou-se a leitura do título, resumo e palavras-chave de cada um dos 117 estudos restantes, sobre os quais foram aplicados os critérios de exclusão e inclusão. Como resultado, foram selecionados 39 estudos potencialmente relevantes, pois esta seleção baseou-se apenas na leitura e interpretação do conteúdo dos respectivos metadados.

Na atividade de extração dos dados, descrita na próxima seção, foi realizada a leitura na íntegra de cada um desses 39 estudos e constatou-se que apenas 9 deles realmente investigam o uso de padrões de requisitos em projeto, construção, teste ou manutenção de software.

⁸ A atividade de busca foi realizada no período de 24 de abril a 5 de maio de 2018.

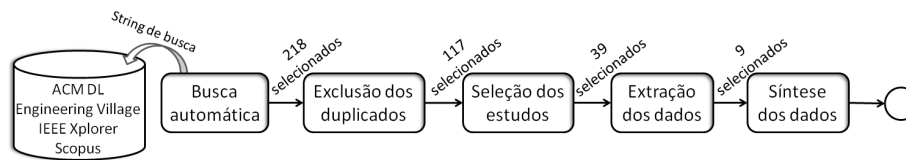


Figura 1. Atividades da fase de condução do Mapeamento Sistemático.

A lista a seguir consiste dos 9 estudos relevantes para extração e síntese (vide Seções 3 e 4), segundo o protocolo definido para este mapeamento sistemático. Esses estudos serão identificados ao longo deste artigo como A1 a A9:

- A1** - Adaptive requirement-driven architecture for integrated healthcare systems [25]
- A2** - Analysing security requirements patterns based on problems decomposition and composition [23]
- A3** - An architectural framework of the integrated transportation information service system [5]
- A4** - Application of ontologies in identifying requirements patterns in use cases [6]
- A5** - Effective security impact analysis with patterns for software enhancement [16]
- A6** - From requirement to design patterns for ubiquitous computing applications [13]
- A7** - Modeling design patterns with description logics: A case study [1]
- A8** - Mutation patterns for temporal requirements of reactive systems [22]
- A9** - SACS: A pattern language for Safe Adaptive Control Software [10]

Quanto à aplicação dos critérios de exclusão, dos 117 artigos não-duplicados, 108 foram excluídos da seguinte maneira: 48% dos artigos (52) foram excluídos por abordarem padrões de requisitos apenas na engenharia de requisitos (**CE4**); 31,5% deles (34) por tratarem de padrões de requisitos, porém não relacionados a software (**CE3**); e os 20,5% restantes (22) pelos demais critérios.

3 Extração dos Dados

Esta seção descreve o processo de extração de dados baseado na leitura completa dos 9 artigos relevantes (A1 a A9) deste estudo. Apresenta-se uma análise comparativa dos tipos de contribuição, bem como o seguinte conteúdo extraído de cada um desses artigos:

1. o tipo de pesquisa apresentado no estudo;
2. o tipo de requisito para o qual o padrão de requisito é proposto;
3. a fase do ciclo de vida a qual o padrão de requisito está relacionado; e
4. o tipo de contribuição apresentada no estudo.

Dado o primeiro critério, os 9 estudos foram classificados pelo tipo de pesquisa desenvolvida, usando para este critério a definição de Petersen et al. [20], em que um conjunto de condições atendidas por uma pesquisa conduzem a uma classificação da mesma. Por exemplo, um artigo de opinião reporta o ponto

de vista do autor sobre um assunto, porém sem nenhuma descrição de uso na prática, nem avaliação empírica, relato de experiência do autor, ou proposta de arcabouço conceitual ou de uma nova solução.

Conforme apresenta a Tabela 1, a maioria dos artigos (8 de 9) enquadra-se, pelos critérios de Petersen et al. [20], como “proposta de solução”, pois não apresentam avaliação empírica da solução proposta em suas pesquisas. Dessas 8 propostas de solução, 3 apresentam uma prova de conceito informal como validação, sem descrever um estudo de caso, nem um experimento; os outros 5 não validam a solução proposta. Por fim, dentre os 9 artigos relevantes, apenas um foi classificado como “pesquisa de validação” (A7), por apresentar um estudo de caso, embora realizado em ambiente acadêmico.

Tabela 1. Tipos de pesquisa e validação dos artigos relevantes.

Tipo da pesquisa	Tipo de Validação
Proposta de solução	Prova de conceito informal: A2, A5, A9
	Sem validação: A1, A3, A4, A6, A8
Pesquisa de validação	Estudo de caso: A7

Em seguida, foi analisado se havia algum tipo específico de requisito tratado pelos padrões de requisitos propostos. Dos 9 artigos relevantes, 4 deles definem padrão para requisitos de adaptabilidade, 4 para requisitos de segurança e 1 define padrão para requisitos de propósito geral. A Tabela 2 mostra os artigos de acordo com os tipos de requisitos.

Tabela 2. Tipo de requisito considerado no padrão proposto.

Tipo do requisito	Artigos
Requisito de adaptabilidade	A1, A3, A6, A8
Requisito de segurança	A2, A5, A7, A9
Requisito de propósito geral	A4

Foi realizada uma análise comparativa detalhada entre as contribuições propostas em cada um dos trabalhos, a partir da qual pode-se notar algumas similaridades apresentadas a seguir.

Os artigos A1 e A3 propõem uma *arquitetura conceitual* semelhante para sistemas desenvolvidos a partir de padrões de requisitos. A comparação realizada entre essas propostas está representada na Figura 2: as linhas tracejadas A, B, C e D demonstram os pontos em comum entre as arquiteturas propostas por A1 (à esquerda) e A3 (à direita). Nas duas arquiteturas, a camada de requisitos (A) identifica, analisa e modela os possíveis requisitos como padrões de requisitos de

usuário (URP). A camada de serviço (B) interage com a camada de requisitos e fornece serviços para satisfazer o URP. O mecanismo de segurança e compartilhamento de informação (C) garante um processo de troca de informação confiável entre os sistemas do mesmo domínio. A base de conhecimento (D) agrupa padrões, normas e ontologias do domínio do sistema. A motivação de ambas as pesquisas é a necessidade de compartilhamento de informações entre sistemas de um mesmo domínio: A1, para sistemas da área médica, e A3, para sistemas de transporte.

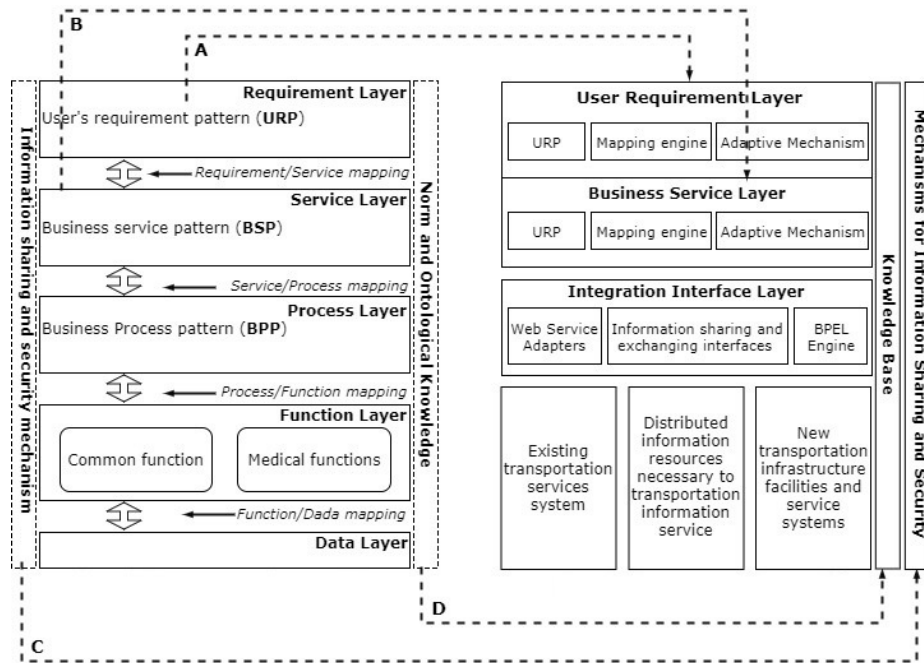


Figura 2. Análise comparativa entre as arquiteturas baseadas em padrões de requisitos propostas em A1 (à esquerda) e A3 (à direita).

Com relação ao objeto de estudo deste mapeamento sistemático, observa-se que os trabalhos apresentados em A1 e A3 estão diretamente relacionados à fase de *projeto de software*, pois os padrões de requisitos do usuário (URP) da camada de requisitos servem de base para a seleção eficiente dos serviços (BSP) na camada de serviços. Os URPs são os principais elementos na camada de requisitos que correspondem aos requisitos do usuário e orientam o funcionamento de todo o sistema.

Semelhanças também foram identificadas nos artigos A2 e A5 sobre como *representar requisitos de segurança na forma de padrão de requisito*. Foi possível observar que ambas as pesquisas tratam os mesmos conceitos de segurança (i.e.

contexto, ativos e ameaças) como padrões de requisitos, e as medidas de proteção são tratadas nos padrões de projeto. As semelhanças identificadas entre os estudos de A2 (à esquerda) e A5 (à direita) estão ilustradas na Figura 3 como linhas tracejadas. Os passos descritos na abordagem do estudo A2 — a identificação de interessados e objetivos, de ativos essenciais de informação, e de fontes de ameaças usando padrões — correspondem, respectivamente, aos seguintes itens do padrão de requisito de segurança do estudo A5: a definição do formato do padrão (contexto, problema, solução e estrutura), dos ativos, e das ameaças. A atividade “adicionar medidas de proteção no projeto do sistema” do estudo A2 é tratada como padrão de projeto de segurança no estudo A5.

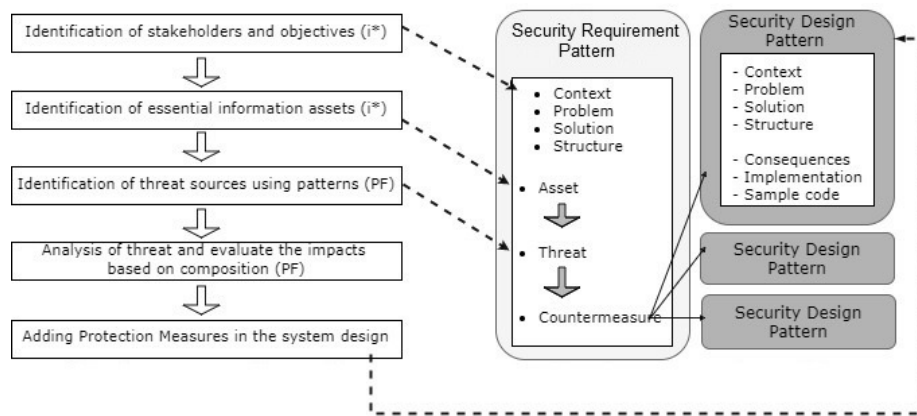


Figura 3. Análise comparativa entre as abordagens de segurança apresentadas em A2 (à esquerda) e A5 (à direita).

Conclui-se que os trabalhos A2 e A5 também estão relacionados com a fase de *projeto de software*, uma vez que ambos definem padrões de requisitos de segurança, de forma a impactar diretamente na definição de medidas de proteção por meio de padrões de projeto.

Foi também identificado que os artigos A8 e A9 apresentam propostas relacionadas ao formato de representação de padrões de requisitos. No artigo A8, associa-se uma fórmula em linguagem temporal linear para cada requisito escrito em linguagem natural, utilizando mutações para amenizar os problemas que podem surgir nessa associação. Para cada tipo de padrão de requisito, as falhas potenciais são identificadas e mutações apropriadas são introduzidas. As fórmulas que acompanham os mutantes podem ser usadas para a geração de testes, a adequação de conjuntos de testes, ou a construção automática de monitores para a verificação em tempo de execução do comportamento do sistema. Sendo assim, as mutações incluídas na transformação dos padrões de requisitos contribuem para a *fase de teste* no ciclo de vida de software. Já no artigo A9 é apresentada uma proposta de padrão de software que integra três tipos de padrão

(de requisito, projeto e segurança), o qual nomeia como padrão composto. Baseado na ideia de *problem frames*, o padrão composto utiliza parâmetros extraídos do seu padrão de requisito. Cada conjunto de funções do padrão de requisito corresponde a soluções no padrão de projeto e a elementos contextuais no padrão de segurança. Portanto, essa proposta está também relacionada à *fase de projeto* de software.

Os artigos A4 e A7 utilizam o formalismo da lógica de descrição (LD) de ontologias para representar padrões de requisitos. No estudo A4, requisitos escritos em LD permitem a geração automática de código fonte; portanto, refere-se à *fase de construção* de software. Na proposta A7, foi implementado um mecanismo que, dado um padrão de requisito de segurança descrito em LD, consegue encontrar uma solução correspondente na forma de padrão de projeto. Daí também relaciona-se à *fase de projeto* de software.

Por fim, o estudo A6 tem como objetivo mapear a dependência entre padrões de projeto e padrões de requisitos de aplicações de computação ubíqua, cuja característica recorrente é a adaptabilidade. Uma abordagem que integra esses padrões tem como objetivo transpor as lacunas da fase inicial do desenvolvimento de software, em que os requisitos recorrentes exigem soluções semelhantes. Em suma, a principal contribuição do artigo A6 é para a fase de *projeto de software*.

As informações extraídas e a análise comparativa realizada na fase de extração desse mapeamento estão sintetizadas na Tabela 3 de acordo com 4 tipos de contribuições identificadas nos artigos: arquitetura conceitual para sistemas que usam padrão de requisito; processo para descoberta e uso de padrão de requisito; definição de formato de representação de padrão de requisito; e proposta de catálogo de padrões de requisitos.

Tabela 3. Informações extraídas dos artigos relevantes.

Tipo de contribuição do estudo	Fase do ciclo de vida	Tipo de requisito	Artigos
Arquitetura conceitual baseada em padrão de requisito	Projeto	Adaptabilidade	A1, A3
Formato de representação de padrão de requisito	Projeto	Segurança	A2, A5, A9
	Teste	Adaptabilidade	A8
Processo para descoberta e uso de padrão de requisito	Projeto	Segurança	A7
	Construção	Propósito Geral	A4
Catálogo de padrão de requisito	Projeto	Adaptabilidade	A6

4 Síntese dos dados

Esta seção apresenta uma síntese dos dados obtidos na fase de extração com o objetivo de responder às questões de pesquisa deste estudo secundário.

4.1 Sobre a Questão de Pesquisa 1

Em resposta à questão de pesquisa “*Em quais fases do ciclo de vida de software são usados padrões de requisitos: projeto, construção, teste e/ou manutenção?*”, 7 estudos utilizam padrões de requisitos na fase de projeto de software, 1 na construção, 1 em testes de software e não houve nenhum para manutenção.

Dentre os 7 estudos que abordam padrões de requisitos em projeto de software (A1–A3, A5–A7 e A9), não existem autores de mais de um estudo, logo não se trata de uma iniciativa de um ou mais grupos de pesquisa. Duas hipóteses podem ser pensadas para a alta concentração dos estudos relacionados à fase de projeto: primeiro é o fato desta ser subsequente à engenharia de requisitos no processo de desenvolvimento tradicional, segundo a disseminação crescente do uso de padrões de projeto no desenvolvimento de software.

Além disso, embora tenham sido identificados 9 estudos que relacionam padrões de requisitos a alguma fase do ciclo de vida, 52 artigos foram excluídos na atividade de seleção pelo critério **CE4**, i.e. tratavam de padrão de requisito apenas na engenharia de requisitos. Essa diferença na quantidade de pesquisas relacionadas à engenharia de requisitos e aquelas que expandem padrões de requisitos para as demais fases do ciclo de vida deixa evidente uma oportunidade de pesquisa sobre padrão de requisito nas fases de construção, teste e manutenção.

Outra evidência é a inexistência de pesquisas sobre o uso de padrão de requisito que permeie o processo de desenvolvimento ao longo do ciclo de vida, i.e. partindo da engenharia de requisitos, passando pelo projeto, construção, teste, e sendo refinado na manutenção de software. Um desafio de estudo é pensar em como propor o uso de padrões de requisitos de forma que estes impactem na melhoria do processo de desenvolvimento como um todo, além dos conhecidos benefícios de qualidade de documentação e de economia de tempo.

4.2 Sobre a Questão de Pesquisa 2

Com relação à questão de pesquisa: “*Há evidências de uso prático de padrões de requisitos nessas fases do ciclo de vida de software?*”, em nenhum dos artigos foi reportada evidência do uso de padrão de requisitos em ambiente de produção de software. Um total de 8 dos 9 artigos encontra-se no estágio de propostas de solução, sem validação, e apenas 1 artigo (A7) é validado com estudo de caso. Esta análise sugere que futuros trabalhos no tema em questão atenham-se cada vez mais à aplicação prática de padrão de requisito na indústria de software em todo o ciclo de desenvolvimento.

4.3 Sobre a Questão de Pesquisa 3

Referente à questão de pesquisa “*Há benefícios relatados do uso de padrão de requisitos nessas fases? Se sim, quais métricas são utilizadas para medir esses benefícios?*”, identificou-se que nenhum dos artigos relatou explicitamente qualquer benefício oriundo da prática de padrão de requisito.

Essa ausência de preocupação com métricas para analisar os benefícios do uso de padrões de requisitos se deve ao fato da maioria dos artigos estar no nível de propostas, sem uso prático dos mesmos.

4.4 Discussão

Ao interligar as informações da fase do ciclo de vida, do tipo de padrão de requisito proposto e o tipo de pesquisa de cada artigo, foi possível elaborar o gráfico de bolhas da Figura 4. Ao analisar o gráfico, observa-se que 4 estudos (A2, A5, A7 e A9) propõem padrão de requisito de *segurança* com uso na fase de *projeto*. Supomos que usar padrão de requisito para segurança deve-se ao fato desta característica ser recorrente em muitos sistemas, contando também com o apoio de padrões internacionais, como a família de padrões ISO/IEC 27000. Entretanto, esses estudos com segurança ainda carecem de maior validação com avaliações empíricas e uso na indústria de software.

Dos 9 estudos relevantes, 4 estudos (A1, A3, A6 e A8) utilizam padrão de requisito para *adaptabilidade*, sendo o último na fase de *testes* e os demais na fase de *projeto*. Além disso, nenhum desses 4 estudos apresenta validação de seus padrões, i.e., caracterizam-se como pesquisas de proposta conceitual de solução. Por fim, o estudo A4 é voltado para padrão de requisito de *propósito geral* com uso na fase de *construção* de software, mas também não apresentou validação.

Considerando ainda a Figura 4, tão importante quanto mapear as propostas de pesquisa é analisar as lacunas existentes:

1. existe uma carência geral de pesquisas de adoção de padrão de requisito em outras fases do ciclo de vida (9 estudos), excetuando a engenharia de requisitos (52 estudos).
2. ao analisar o lado esquerdo da figura, conclui-se que requisitos não funcionais (adaptabilidade e segurança) são os mais estudados quando do uso de padrão de requisito para as fases de projeto e testes. Porém, outros tipos de requisitos não funcionais podem ser explorados com padrão de requisito envolvendo outras fases do ciclo de vida como, por exemplo, aspectos de usabilidade com a geração automática de código e de casos de teste.
3. ao considerar o lado direito da figura, identificamos uma lacuna quanto à aplicação, na indústria de software, dos resultados das pesquisas de padrões de requisitos nas fases de projeto, construção, testes e manutenção — quase todas as propostas encontram-se em estágio de prova de conceito.

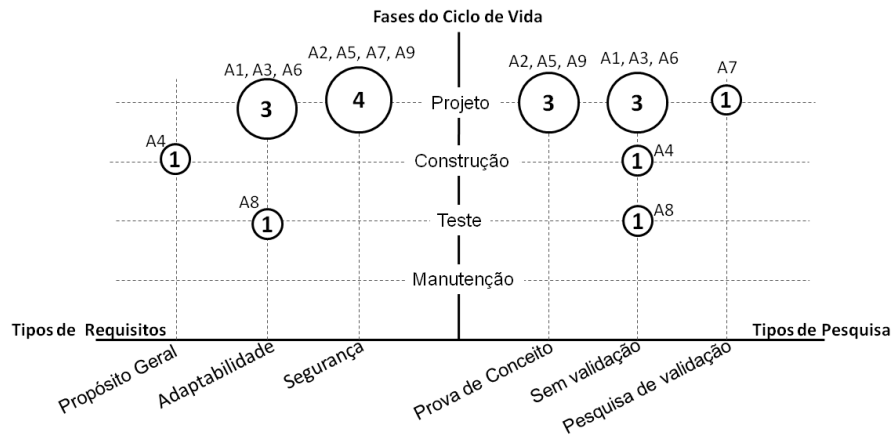


Figura 4. Mapeamento dos tipos de requisitos e de pesquisas sobre padrões de requisitos mapeados nas fases do ciclo de vida de software.

5 Considerações Finais

Os maiores problemas enfrentados em estudos sistemáticos são encontrar todos os estudos relevantes sobre um tema e, a partir dos estudos relevantes, selecionar evidências de qualidade. Por isso, ameaças foram mitigadas com 2 ações ao longo do planejamento e da condução deste mapeamento sistemático.

Primeiro, adotou-se uma estratégia de busca automática que combina fontes de informação consideradas relevantes para a Engenharia de Software com termos da *string* de busca baseados no vocabulário padrão SEVOCAB. Busca na literatura cinzenta não fez parte do protocolo, mais especificamente teses e relatórios técnicos, pois supõe-se que esse tipo de literatura, quando de boa qualidade, está publicada como artigo em periódicos ou conferências.

Segundo, três pesquisadores elaboraram o protocolo deste mapeamento sistemático e na condução do mapeamento, cada pesquisador exerceu seu papel: o pesquisador A, especialista em Engenharia de Requisitos, realizou a identificação e a extração de dados dos estudos relevantes; o pesquisador B, especialista em Engenharia de Software, verificou os resultados da fase de extração para mitigar a possibilidade de vieses ao longo do processo; e o pesquisador C, líder da equipe e com vasta experiência em Engenharia de Software, juntamente com A e B, realizou a análise, síntese e escrita dos resultados. No caso de divergências, A, B e C resolveram os conflitos em conjunto.

Apesar dos resultados obtidos neste mapeamento sistemático, de poucos estudos sobre o tema e com baixo índice de validação das propostas, os autores reforçam a reconhecida importância de padrões de requisitos no meio acadêmico [17,2]. Para impulsionar o desenvolvimento das pesquisas em padrões de requisitos no processo de desenvolvimento de software, sugerimos que a comunidade acadêmica aproxime-se da indústria de software para identificar as

reais expectativas do setor. A comunidade deve também estabelecer métricas que evidenciem as vantagens do uso de padrões de requisitos em todas as fases do ciclo de vida como, por exemplo, a redução no tempo de projeto e na geração automática de código-fonte, a execução padronizada de testes e a melhoria na qualidade das especificações de software. Complementar a essas ações, a comunidade deve propor novas ferramentas de suporte ao desenvolvimento de aplicações com o uso de padrões de requisitos. Por fim, os pesquisadores devem elaborar metodologias de desenvolvimento que demonstrem como usar padrões de requisitos ao longo das fases do ciclo de vida de forma integrada às demais ações propostas.

Agradecimentos

Taciana Kudo agradece o auxílio financeiro concedido pelo Programa de Apoio à Pós-graduação (PROAP/CAPES). Renato Bulcão-Neto agradece a bolsa concedida pela CAPES/FAPEG (proc. n. 88887.305511/2018-00), vinculada ao estágio pós-doutoral realizado no Depto. de Computação e Matemática da FFCLRP-USP. Alessandra Macedo agradece o apoio financeiro da FAPESP (proc. n. 16/13206-4) e do CNPq (proc. n. 302031/2016-2 e 442533/2016-0).

Referências

1. Asnar, Y., Paja, E., Mylopoulos, J.: Modeling design patterns with description logics: A case study. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 6741 LNCS, pp. 169 – 183. London, United kingdom (2011)
2. Barros-Justo, J.L., Benitti, F.B.V., Leal, A.C.: Software patterns and requirements engineering activities in real-world settings: a systematic mapping study. *Comp. Standards & Interfaces* **58**, 23–42 (2018)
3. Beckers, K., Côté, I., Goeke, L.: A catalog of security requirements patterns for the domain of cloud computing systems. In: Proceedings of the ACM Symposium on Applied Computing. pp. 337–342 (2014)
4. Bourque, P., Fairley, R.E. (eds.): SWEBOK: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, Los Alamitos, CA, version 3.0 edn. (2014)
5. Chang, F., Gan, R.: An architectural framework of the integrated transportation information service system. In: 2009 IEEE International Conference on Grey Systems and Intelligent Services, GSIS 2009. pp. 1342 – 1346. Nanjing, China (2009)
6. Couto, R., Ribeiro, A.N., Campos, J.C.: Application of ontologies in identifying requirements patterns in use cases. In: Electronic Proceedings in Theoretical Computer Science, EPTCS. vol. 147, pp. 62 – 76. Grenoble, France (2014)
7. Da Silva, R., Benitti, F.: Standards writing requirements: A mapping systematic literature [Padrões de escrita de requisitos: Um mapeamento sistemático da literatura]. In: 14th Ibero-American Conference on Software Engineering and 14th Workshop on Requirements Engineering, CIbSE 2011. pp. 259–270 (2011)
8. Fabbri, S.C.P.F., Felizardo, K.R., Ferrari, F.C., Hernandez, E.C.M., Octaviano, F.R., Nakagawa, E.Y., Maldonado, J.C.: Externalising tacit knowledge of the systematic review process. *IET Software* **7**(6), 298–307 (2013)

9. Felizardo, K.R., Nakagawa, E.Y., Fabbri, S.C.P.F., Ferrari, F.C.: Revisão sistemática da literatura em Engenharia de Software: Teoria e Prática. Elsevier, first edn. (2017)
10. Hauge, A.A., Stølen, K.: SACS: A pattern language for safe adaptive control software. In: Proceedings of the 18th Conference on Pattern Languages of Programs. pp. 7:1–7:22. PLoP '11, ACM, New York, NY, USA (2011)
11. ISO/IEC/IEEE: IEEE software and systems engineering vocabulary. IEEE Computer Society pp. 1 – 437 (2016)
12. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - a systematic literature review. *Inf. Softw. Technol.* **51**(1), 7–15 (Jan 2009)
13. Knotte, R., Baraki, H., Söllner, M., Geihs, K., Leimeister, J.M.: From requirement to design patterns for ubiquitous computing applications. In: Proceedings of the 21st European Conference on Pattern Languages of Programs (Jul 2016)
14. Konrad, S., Cheng, B.H.C.: Requirements patterns for embedded systems. In: Proceedings IEEE Joint International Conference on Requirements Engineering. pp. 127–136 (2002)
15. Nicolas, J., Toval, A.: On the generation of requirements specifications from software engineering models: A systematic literature review. *Inf. Softw. Technol.* **51**(9), 1291–1307 (Sep 2009)
16. Okubo, T., Kaiya, H., Yoshioka, N.: Effective security impact analysis with patterns for software enhancement. In: 2011 Sixth International Conference on Availability, Reliability and Security. pp. 527–534 (Aug 2011)
17. Palomares, C., Quer, C., Franch, X.: Requirements reuse and requirement patterns: a state of the practice survey. *Empirical Software Engineering* **22**(6), 2719–2762 (2017)
18. Palomares, C., Quer, C., Franch, X., Renault, S., Guerlain, C.: A catalogue of functional software requirement patterns for the domain of content management systems. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18–22, 2013. pp. 1260–1265 (2013)
19. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. pp. 68–77. EASE'08, BCS Learning & Development Ltd., Swindon, UK (2008)
20. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* **64**, 1–18 (2015)
21. Tockey, S.: Insanity, hiring, and the software industry. *Computer* **48**(11), 96–101 (Nov 2015)
22. Trakhtenbrot, M.: Mutation patterns for temporal requirements of reactive systems. In: Proceedings - 10th IEEE International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2017. pp. 116–121 (2017)
23. Wen, Y., Zhao, H., Liu, L.: Analysing security requirements patterns based on problems decomposition and composition. In: 2011 1st International Workshop on Requirements Patterns, RePa'11. pp. 11 – 20. Trento, Italy (2011)
24. Withall, S.: Software Requirement Patterns. Best practices, Microsoft Press, Redmond, Washington (2007)
25. Yang, H., Liu, K., Li, W.: Adaptive requirement-driven architecture for integrated healthcare systems. *Journal of Computers* **5**(2) (2010)