

Um Modelo sobre as Dificuldades para Especificar Casos de Uso

Elizamary Nascimento¹, Williamsom Silva¹, Breno Bernard Nicolau de França²,
Bruno Gadelha¹, Tayana Conte¹

¹ Universidade Federal do Amazonas, Manaus – Amazonas, Brazil

² Universidade Estadual de Campinas, São Paulo, Brazil

{elizamary.souza, williamson.silva,bruno,tayana}@icomp.ufam.edu.br,
{breno}@ic.unicamp.br

Resumo. Contexto: Casos de Uso (Use Cases – UCs) tornaram-se um importante artefato para a especificação dos requisitos de software. No entanto, há várias dificuldades que impedem estudantes e engenheiros de software de especificarem UCs de forma correta. **Objetivo:** Para explorar e entender as dificuldades em especificar UCs, foi realizado um estudo qualitativo. **Método:** Utilizou-se entrevistas semiestruturadas com estudantes visando identificar estas dificuldades. A análise dos dados foi conduzida utilizando procedimentos do método Grounded Theory (GT). **Resultados:** Foi proposto um modelo que apresenta as dificuldades encontradas para especificação de UC. Estas dificuldades foram classificadas nas categorias: abstração dos requisitos, identificação e descrição dos fluxos, relacionamento entre as regras de negócio e os fluxos na especificação do UC. **Conclusões:** O modelo serve como base para futuras pesquisas na área, bem como no apoio à sugestão de práticas para melhorar o processo de ensino/aprendizagem dos alunos na especificação de UC.

Palavras chave. Casos de Uso, Dificuldades em Especificar Casos de Uso, Análise Qualitativa.

1 Introdução

A indústria de software vem adotando Casos de Uso (Use Cases - UC) para especificar os requisitos de sistemas [2] [7]. Jacobson et al. [9] introduziram o conceito de UC como uma forma dos profissionais de software obterem uma melhor compreensão dos requisitos de um sistema. Casos de Uso consistem normalmente de duas partes: o diagrama de UC e a especificação de UC [21]. O diagrama fornece uma visão geral das funcionalidades do sistema [5]. A especificação representa o comportamento das funcionalidades do sistema detalhando a interação entre os atores e o sistema [6].

A especificação do UC pode ser uma atividade demorada e propensa à falhas, pois as especificações geralmente são escritas em linguagem natural [17]. Além disso, profissionais e estudantes em engenharia de software apresentam dificuldades durante a especificação do UC, tais como: dificuldade em especificar os passos dos fluxos nos

UCs, dificuldade em organizar as informações na especificação do UC, dentre outras [1] [5] [13]. Estas dificuldades podem afetar a qualidade do software, pois vários tipos de defeitos podem ser inseridos nas especificações de UCs e propagados para os outros artefatos, como informações ambíguas, incompletude de informações devido à ausência de passos nos fluxos e ausência de Regras de Negócio (RN) [2] [5] [13].

Para compreender melhor as dificuldades de especificação de UC, foi realizado um estudo qualitativo que tem como objetivo responder a seguinte questão de pesquisa: “*Quais as dificuldades percebidas ao especificar casos de uso durante o desenvolvimento de software?*”. Para responder esta questão de pesquisa, realizou-se entrevistas semiestruturadas com 24 estudantes (de graduação e pós-graduação) que possuíam experiência prévia em especificação de Casos de Uso. A análise dos dados obtidos foi feita qualitativamente empregando procedimentos do método de Grounded Theory (GT) [20]. Com base nos resultados, foram identificadas as dificuldades percebidas pelos estudantes ao especificar UC e foi desenvolvido um modelo que tem por objetivo sumarizar estas dificuldades. Além disso, são sugeridas algumas práticas que foram identificadas na literatura que podem melhorar o processo de ensino de especificação de UC.

Este artigo está organizado da seguinte forma: a Seção 2 aborda conceitos sobre a especificação de UC e os trabalhos relacionados. A Seção 3 descreve a condução e execução do estudo qualitativo. A Seção 4 apresenta os resultados e o modelo sobre dificuldades para especificar UC. A Seção 5 apresenta as discussões dos resultados do estudo. A Seção 6 apresenta as principais ameaças à validade do estudo. Por fim, na Seção 7 apresenta as conclusões e os trabalhos futuros.

2 Background e Trabalhos Relacionados

2.1 Especificação de Casos de Uso

De acordo com Tiwari e Gupta [21], existem 20 propostas de templates que podem ser utilizados para especificar UC. Os templates geralmente apresentam duas partes: a parte de informações estáticas e de informações dinâmicas do UC [12]. As informações estáticas abrangem os seguintes elementos: ator (es), pré-condição (ões) e pós-condição (ões). As informações dinâmicas são formadas pelos fluxos do UC [19]: um fluxo principal, nenhum ou mais fluxos alternativos e os fluxos de exceções. Cada fluxo do UC (principal, alternativo e exceção) é descrito através de uma sequência de passos que descrevem a interação entre o usuário e o sistema [9] [19]. O fluxo principal é executado quando o usuário aciona o UC [12] e é finalizado quando o objetivo do usuário é alcançado (no fluxo principal ou alternativo) ou quando o objetivo é abortado pelo usuário [23]. Os fluxos alternativos descrevem outros cenários e são iniciados a partir do fluxo principal [23]. O fluxo de exceção ocorre quando algo inesperado ocorre na interação entre o usuário e o sistema [6] [12]. Para que os fluxos alternativos e de exceções sejam executados, as condições devem ser indicadas nos passos do fluxo principal ou alternativo do UC [12] [23].

Além disso, existem as regras de negócio que representam declarações que descrevem procedimentos ou restrições relacionadas a organização ou ao contexto

onde a organização atua [8]. As regras de negócio afetam diretamente o comportamento do sistema, ou seja, os Casos de Uso contêm restrições que devem ser seguidas durante sua execução que devem ser identificadas e detalhadas na especificação do UC [10].

2.2 Trabalhos Relacionados

Os estudos apresentados a seguir foram realizados em contextos e com objetivos diferentes. Contudo, apresentam resultados que mostram que os participantes de cada estudo tiveram dificuldades ao utilizar os artefatos da UML.

Anda et al. [1] realizaram um estudo de caso visando identificar as dificuldades percebidas por profissionais de software (desenvolvedores, gerentes de projetos, dentre outros) que estavam atuando em projetos de software que utilizavam UML. Sobre as especificações de UCs, os resultados mostraram que os profissionais tiveram dificuldades tanto para definir o nível de detalhes quanto para organizar as informações na especificação do UC.

Bolloju [5] realizou um estudo experimental que identificou as dificuldades que analistas novatos em desenvolvimento de software possuem ao utilizar artefatos da UML em projetos de software. Com relação as especificações dos UCs, os resultados mostraram que as dificuldades mais frequentes na especificação de UC estavam relacionadas à descrição de passos dos fluxos nos UCs, ou seja, houve dificuldade para especificar os passos nos fluxos, dificuldade em especificar as informações de forma não ambígua e dificuldade em referenciar corretamente os fluxos alternativos nos passos do UC.

Nascimento et al. [13] realizaram um estudo experimental com o propósito de comparar dois formatos para especificar UC (textual e gráfico) usados por estudantes sem experiência prévia na indústria. Os resultados mostraram que houve mais dificuldades em identificar e descrever fluxos e regras dos Casos de Uso utilizando o formato gráfico (baseado em diagramas de atividades UML). No entanto, os estudantes tiveram mais dificuldades para descrever os fluxos e regras de negócio utilizando o formato textual. Além disso, os estudantes tiveram dificuldades em entender o requisito e, assim, seguir os passos corretamente para a especificação do UC utilizando os formatos textual e gráfico.

Dos estudos apresentados acima, dois estudos (Anda et al [1] e Bolloju [5]) abordam sobre as dificuldades de artefatos da UML utilizados no desenvolvimento de software, apresentando apenas algumas das dificuldades em especificar UCs. Nascimento et al. [13], por sua vez, realizaram um estudo inicial no contexto de UC e identificaram algumas dificuldades ao especificar UC. Visando complementar estes resultados, o foco deste artigo é identificar quais são as dificuldades encontradas durante o processo de escrita da especificação do UC.

3 Método de Pesquisa

Um estudo qualitativo foi realizado com o objetivo de identificar quais as dificuldades percebidas pelos estudantes ao especificar UC. Para realizar a coleta dos

dados qualitativos, foram utilizadas entrevistas semiestruturadas. Segundo Manotas et al. [11], as entrevistas são úteis para reunir um conjunto de observações e percepções qualitativas dos participantes. Além disso, entrevistas ajudam a obter uma compreensão ampla do contexto em que os entrevistados atuam [11].

3.1 Planejamento

As perguntas incluídas no roteiro da entrevista foram desenvolvidas colaborativamente por dois pesquisadores e revisadas por outros três pesquisadores. Após a revisão, realizou-se um estudo piloto a fim de avaliar a clareza das perguntas, a estrutura e o tempo total da entrevista. O resultado do estudo piloto mostrou que não eram necessárias alterações importantes no roteiro da entrevista. A Tabela 1 apresenta o roteiro com as principais questões utilizadas durante a condução da entrevista. O pacote experimental contendo mais informações sobre este estudo encontra-se disponível em um relatório técnico [14].

Tabela 1. Roteiro Semiestruturado.

Parte I – Background do Participante
Q1 - Você já participou de projetos de desenvolvimento de software (na indústria ou na academia)? Q2 - Você já participou de projetos de desenvolvimento de software que utilizaram casos de uso (na indústria ou academia)?
Parte II – Especificação de Casos de Uso
Q3 - Qual a sua percepção sobre a utilização da especificação de casos de uso durante o desenvolvimento de software?
Parte III – Dificuldades em Descrever Casos de Uso
Q4 - Você tem/teve alguma dificuldade para especificar um caso de uso?

Conforme visto na Tabela 1, as perguntas foram apresentadas em formato de funil, começando com questões gerais e avançando para questões mais específicas, conforme sugerido por Runeson e Host [18].

3.2 Participantes e Contexto do Estudo

O estudo teve como participantes estudantes de graduação e pós-graduação de duas universidades (Universidade Federal do Rio de Janeiro - UFRJ e Universidade Federal do Amazonas - UFAM) de diferentes regiões do Brasil. Todos os alunos tinham cursado anteriormente a disciplina de Engenharia de Software, ou seja, já tinham uma experiência prévia com a especificação de UC. Ao todo, foram entrevistados vinte e quatro participantes, conforme a Tabela 2.

Tabela 2. Resumo dos Participantes do Estudo.

Un.	UFRJ											UFAM												
Par.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
Per.	A	A	A	I	A	I	I	I	I	I	A	I	A	I	A	A	A	A	A	A	I	I	A	A
Nív.	G	G	G	PG	G	PG	PG	PG	PG	PG	PG	G	G	G	G	G	G	G	G	G	G	G	G	G

Legenda: Par. – Código do Participante; Per. – Perfil do participante: Experiência em Desenvolvimento de Software na Indústria (I) ou na Academia (A); Nív. – Nível de Escolaridade do Participante: G – Graduação; PG – Pós-Graduação;

Na UFRJ, foram entrevistados onze participantes: quatro eram alunos de graduação, cursavam a disciplina de Desenvolvimento de Software Orientado a Objetos e estavam projetando um sistema Web, cuja especificação de requisitos era baseada em UC, em um ciclo de desenvolvimento contínuo; sete eram alunos de pós-graduação (mestrado e doutorado) e seis eram profissionais de software e já haviam desenvolvido mais de dois projetos de software na indústria que utilizavam UCs.

Na UFAM, foram entrevistados treze alunos de graduação que estavam cursando a disciplina de Análise e Projeto de Sistemas. Quatro dos treze participantes já eram profissionais da indústria de software, já tendo utilizado UC na indústria. Todos os participantes da UFAM estavam projetando sistemas Web e Mobile baseados em UC.

3.3 Coleta de Dados

A fim de atender às exigências éticas da pesquisa, foi explicado a cada participante a finalidade da pesquisa e os seus direitos, através de um Termo de Consentimento Livre e Esclarecido, garantindo a confidencialidade dos dados fornecidos e o anonimato do participante. Além disso, as entrevistas foram realizadas individualmente, dentro do tempo previsto (entre quinze e trinta minutos). Assim, os pesquisadores não apressaram a entrevista e o entrevistado pôde se expressar de forma tranquila e sem interrupções. Durante a coleta dos dados, os participantes poderiam expressar sua percepção sobre a especificação de UC de acordo com sua experiência prática (na academia/indústria). A coleta e análise dos dados foi realizada em dois momentos: (i) no primeiro momento, todos os participantes da UFRJ foram entrevistados, as entrevistas foram gravadas e transcritas e o processo de análise dos dados foi iniciado; (ii) no segundo momento, os participantes da UFAM foram entrevistados, depois todo o processo de análise foi realizado novamente, conforme executado no primeiro momento. Para a análise dos dados, todas as entrevistas foram transcritas e importadas para o Atlas.ti¹. Durante este processo não houve nenhuma evolução nos instrumentos e procedimentos para a coleta e/ou análise dos dados.

3.4 Análise dos Dados

Utilizou-se procedimentos de codificação do método Grounded Theory (GT) para realizar a análise dos dados. GT é um método de pesquisa qualitativa que utiliza um conjunto de procedimentos sistemáticos para criar e avaliar teorias substantivas através das fases de Codificação Aberta, Axial e Seletiva [20]. De acordo com Strauss e Corbin [20], o pesquisador pode usar apenas alguns dos seus procedimentos para atender seus objetivos de pesquisa.

Os dados qualitativos foram analisados utilizando um subconjunto das fases do processo sugerido por Strauss e Corbin [20]: codificação aberta (1ª fase) e codificação axial (2ª fase). Na 1ª fase, os códigos foram criados a partir da análise dos dados qualitativos. Em seguida, os códigos foram agrupados de acordo com suas propriedades e dimensões, criando conceitos que representam categorias e

¹ Atlas.ti – The Knowledge Workbench, Scientific Software Development – <http://www.atlasti.com>

subcategorias. Na 2ª fase, os relacionamentos entre os códigos foram feitos. A 3ª fase é a codificação seletiva, que tem por objetivo integrar a teoria, identificando uma teoria central na qual todas as outras categorias devem ser relacionadas. No entanto, a 3ª fase não foi realizada, pois a saturação teórica ainda não foi alcançada [20].

Após realizar a análise, outro pesquisador com bom nível de conhecimento em análise qualitativa verificou os códigos, as categorias e subcategorias criadas e as associações entre os códigos. Todo o processo de codificação foi revisado por este pesquisador, a fim avaliar os resultados obtidos na análise dos dados.

4 Resultados do Estudo

Nesta seção serão apresentados os resultados da percepção dos participantes com relação: às principais dificuldades que os participantes relataram ao especificar um UC, ao modelo proposto de dificuldades em especificar UC resultante, além dos benefícios percebidos da utilização da especificação de UC.

4.1 Dificuldades em Especificar Casos de Uso

A partir da análise dos dados qualitativos, identificou-se cinco categorias de tipos de dificuldades: Abstração do Requisito, Pré e Pós-Condição, Regras de Negócio, Fluxos dos Casos de Uso, e Referências a Fluxos e Regras de Negócio. Estas categorias serão apresentadas a seguir.

Para a categoria “**Abstração do Requisito**” foram associados os códigos que mostram afirmativas sobre as dificuldades em abstrair o requisito durante a especificação do UC, como o código “*Entender o domínio do problema*”. Para este código, estão relacionadas citações que mostram que o entendimento do domínio do problema foi uma dificuldade para alguns participantes. A Tabela 3 apresenta os códigos e as citações dos participantes relacionadas a categoria “**Abstração do Requisito**”.

Tabela 3. Dificuldades relacionadas à Abstração do Requisito.

Códigos (Abstração do Requisito)	Citações
<i>“Entender o domínio do problema”</i>	<i>“Eu acho que a maior dificuldade é o entendimento do problema, mas a dificuldade de escrever (casos de uso) não muito.” – P5</i> <i>“O problema não foi a transcrição para o caso de uso, mas o problema (em si) que não ficou muito claro pra mim.” – P12</i>
<i>“Identificar todas as funcionalidades do sistema e documentá-las”</i>	<i>“Acho que (identificar) algumas funcionalidades pode ter faltado (...) A gente percebeu, esqueceu e eu percebi agora de novo.” – P16</i> <i>“Faltou o caso de uso atender a todos os cenários (...) e percebemos que foi uma questão de omissão mesmo, ou seja, não ter (os cenários) no caso de uso.” – P2</i>

Apesar do participante P12 possuir experiência na indústria e o participante P5 não possuir experiência (ver Tabela 2), ambos tiveram dificuldades em entender o domínio do problema para especificar o UC. Observou-se que independente da experiência do participante, compreender os requisitos era fundamental para

especificar as informações do UC. Para o segundo código (Identificar todas as funcionalidades do sistema e documentá-las), percebeu-se que devido à pouca experiência prática em especificar UCs, os participantes P16 e P2 (alunos de graduação e sem experiência na indústria) tiveram dificuldades para identificar algumas funcionalidades necessárias para o sistema, logo, omitiam alguns fluxos (cenários) na especificação do UC.

Além de dificuldades com a abstração do problema, os participantes também relataram problemas com elementos da sintaxe do UC. Certas dificuldades citadas estão relacionadas à descrição da pré e pós-condição no UC. Foi então criada a categoria “**Pré e Pós-Condição**”, cujos códigos são apresentados na Tabela 4.

Tabela 4. Dificuldades relacionadas à Pré e Pós-Condição.

Códigos (Pré e Pós-Condição)	Citações
“Definir a Pré-condição do UC”	<p>“A primeira dificuldade foi mais na pré-condição (...) eu colocava algumas coisas que não faziam sentido.” – P15</p> <p>“A dificuldade que eu tive logo quando eu fui fazer a primeira especificação era com relação as pré e pós-condições, o fluxo principal.” – P14</p>
“Definir a Pós-condição do UC”	<p>“O que eu ficava com dificuldade era aquele negócio de pós-condição depois que tinha feito o caso de uso, depois que aconteceu aquele comportamento, eu tinha que pensar nas pós-condições daquilo então até hoje eu sinto muita dificuldade.” – P23</p>

Sobre as dificuldades na Pré e Pós-Condição, os resultados qualitativos mostram que apesar dos participantes P14, P15 e P23 possuírem perfis diferentes (P14 – com experiência na indústria; P15 e P23 – sem experiência), estes participantes tiveram dificuldades em descrever/ entender quais informações eram adequadas para definir o que ficaria na pré e pós-condição do UC. Por exemplo, durante a especificação do UC, são descritas informações que podem não estar relacionadas a pré-condição do UC, como Regras de Negócio (RN) ou Requisitos Não-Funcionais (RNF).

Também foram mencionadas dificuldades em identificar e descrever as RNs associadas ao UC. Por esta razão, foi criada a categoria “**Regras de Negócio**”, com os códigos apresentados na Tabela 5.

Tabela 5. Dificuldades relacionadas às Regras de Negócio.

Códigos (Regras de Negócio)	Citações
“Identificar as Regras de Negócio para o UC”	<p>“Eu tento imaginar as regras de negócio do caso de uso (...) Por exemplo validar o login a senha, o cpf (...). Imaginar as regras de negócio eu acho que foi isso, imaginar as restrições do caso de uso na hora.” – P14</p> <p>“As regras de negócio ficam meio intrínsecas, você não consegue separar ou tem dificuldade para separar e identificar uma regra de negócio que está relacionada aquele caso de uso (...)” – P6</p>
“Confundir as RNs com Requisitos Não-Funcionais”	<p>“Particularmente eu confundo requisitos não funcionais com regras de negócio.” – P1</p> <p>“(…) às vezes você fica meio: o que é regra de negócio é requisito não funcional. Então quando você vê pela primeira vez você confunde onde vai localizar aquela regra. É funcional ou não funcional?” – P3</p>
“Confundir as RNs com os Fluxos do UC”	<p>“Muitas vezes eu confundo um pouco fluxos e regras, isso daqui faz parte do fluxo principal, mas também parece uma regra de negócio, então eu confundo às vezes... eu tento imaginar um pouco a situação para tentar abstrair o que é o quê.” – P20</p>

Para o primeiro código, os participantes P14 e P6 mencionaram que tiveram dificuldades em identificar as regras de negócio do UC. Isto pode estar relacionado ao fato de que, durante a especificação do UC, nem todas as informações, como RNs, estão descritas ou claras na descrição dos requisitos do sistema. Para o segundo código, os participantes P1 e P3 confundiram as regras de negócio com os requisitos funcionais ou RNF do sistema. Esta dificuldade pode ter ocorrido devido os participantes ainda possuírem dúvidas sobre estes conceitos (RN e RNF) durante a especificação do UC. Para o terceiro código, o participante P20 afirmou que confundiu a descrição dos fluxos do UC com a descrição das RNs.

Os participantes mencionaram dificuldades em identificar e/ou descrever os fluxos ao UC. A Tabela 6 apresenta os códigos relacionados à categoria “**Fluxos do Caso de Uso**” e algumas das citações associadas.

Tabela 6. Dificuldades relacionadas aos Fluxos do Caso de Uso.

Códigos (Fluxos do UC)	Citações
“Identificar e Descrever os Fluxos no UC”	<p>“Inicialmente há a dificuldade de escrever e identificar quais fluxos alternativos serão descritos.” – P16</p> <p>“Eu não pensava muito no erro do sistema, então era mais na parte certa eu não pensava em erros, só na parte certa (...) no caminho certo, então eu errava muito nisso (...) Eu não sabia muito bem o que era fluxo alternativo, aí eu precisei pesquisar e ver o que era fluxo alternativo (...)” – P19</p> <p>“Isso de primeira foi complicado, organizar. Esse daqui é meu fluxo principal e esse daqui é meu alternativo (...). Eu não conseguia ver muito bem o que ia ser o alternativo, ou quando ia dar erro (no caso de uso).” – P20</p>
“Descrever a interação entre ator e sistema para os Fluxos do UC”	<p>“A maior dificuldade foi definir como seria a interação entre o sistema e o ator, eu acho que não tem algo específico (...) para todos os fluxos.” – P15</p>
“Descrever a interação de outros sistemas que interagem com o UC”	<p>“(…) quando tem sistemas externos interagindo com o sistema aí eu acho meio difícil de colocar isso. Porque quando eu estou especificando eu não sei que tipo de resposta aquilo vai me dar (...) Saber como ele ia devolver e saber como eu ia tratar isso no meu caso de uso.” – P22</p>

Para o primeiro código, os participantes que mencionaram a dificuldade na identificação e descrição dos fluxos (principal e alternativos) foram alunos de graduação e que não possuem experiência na indústria. Além disso, o participante P16 afirmou anteriormente que teve dificuldade em abstrair as funcionalidades do sistema. Esta dificuldade pode ter influenciado na dificuldade em identificar quais seriam os fluxos alternativos da especificação do UC. O segundo e o terceiro códigos apresentam dificuldades mais específicas. No segundo código, o participante P15 mencionou que teve dificuldade em descrever a interação do ator e sistema nos fluxos do UC. No terceiro código, apesar do participante P22 possuir experiência em UCs na indústria, o participante mencionou que teve dificuldade em descrever a interação do sistema com outros sistemas que interagem com o UC.

Para a categoria “**Referências de Fluxos e Regras de Negócio**”, foram associados os códigos que mencionam dificuldades em referenciar os fluxos e RN nos passos dos fluxos do UC. A Tabela 7 apresenta os códigos e algumas citações relacionadas.

Sobre as dificuldades relacionadas ao primeiro código, os participantes P13 e P23 tiveram dificuldades em referenciar os fluxos (principal e ou/alternativos) nos passos do UC e por conta disso ocorreram dúvidas em saber onde referenciar o fluxo

alternativo nos passos do UC. No segundo código, o participante P10 teve dificuldade em referenciar as RNs nos passos dos fluxos principal e/ou alternativos do UC.

Tabela 7. Dificuldades relacionadas as Referências dos Fluxos e RNs nos passos do UC.

Códigos (Referências dos Fluxos e Regras de Negócio)	Citações
“Referenciar os diferentes Fluxos nos passos do UC”	“(…) os membros da equipe esqueceram de incluir o fluxo alternativo e esqueciam de linkar.” – P13 “Eu fico com dúvida. Ah! Mas o fluxo alternativo vai vir na hora que o ator informa ou na hora que o sistema válida? Então eu acho que a minha maior dificuldade é onde chamar o fluxo alternativo.” – P23
“Referenciar as Regras de Negócio nos Fluxos do UC”	“(…) questões de regras de negócio, as vezes eram mal explicadas ou não deveriam estar naquela subseção (de regras de negócio do UC).” – P10

Com base nos resultados encontrados na análise qualitativa, foi elaborada uma representação gráfica (Figura 1), apresentando o modelo com as dificuldades percebidas durante o processo de especificação de UC. Para a construção do modelo, primeiro extraiu-se as categorias da análise qualitativa. Em seguida as dificuldades que estavam relacionadas a essas categorias foram classificadas, ou seja, os códigos de cada categoria foram agrupados. A partir do modelo observou-se que podem existir possíveis relações entre as dificuldades apresentadas, discutidas na Seção 5. O modelo tem por objetivo: (a) auxiliar alunos e profissionais de software que utilizam UC na academia e/ou na prática na indústria; (b) indicar pontos de melhoria em diretrizes, técnicas e abordagens que auxiliam profissionais de software na especificação de UC; e (c) auxiliar os pesquisadores na construção de técnicas voltadas para o ensino da especificação de UC.

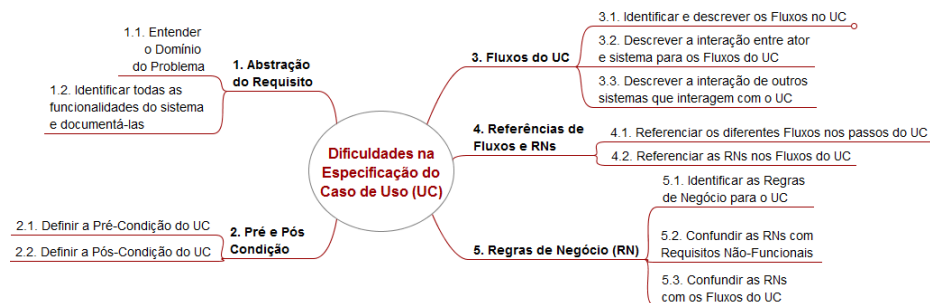


Figura 1. Modelo de Dificuldades sobre Especificação de Casos de Uso.

4.2 Percepção sobre os benefícios de utilização de UC

Além de relatar suas dificuldades, os participantes também comentaram quais suas percepções sobre os benefícios de Casos de Uso. Foi então criada uma categoria a parte, chamada “Benefícios em Utilizar UC no Desenvolvimento de Software”. A Tabela 8 apresenta seus códigos e exemplos de citações relacionadas.

No primeiro código, os participantes P10 e P12 por possuírem experiência na indústria mencionaram que a especificação de UC é necessária, pois, detalha os fluxos

(principal, alternativos e exceções). Além disso, afirmaram que a especificação do UC é útil para detalhar a interação do ator com o sistema, pois descreve uma sequência lógica dos passos que serão desenvolvidos e assim alcançar o resultado esperado. No segundo código, os participantes P7 e P23 mencionaram que a especificação do UC ajuda a equipe de projeto a entender o que será desenvolvido para uma funcionalidade específica do sistema, pois apresenta de forma estruturada toda a ideia do UC.

Tabela 8. Benefícios em Utilizar UC no Desenvolvimento de Software.

Códigos (Benefícios)	Citações
<i>“Ajuda a descrever o comportamento e ver o que vai ser desenvolvido para o caso de uso”</i>	<i>“Porque você tem (...) uma visão geral do caso de uso, você tem os fluxos principais, os fluxos alternativos, as exceções, então é um detalhamento, que ao meu ver é necessário para o desenvolvimento.” – P10</i> <i>“A especificação do caso de uso (em si) chega mais perto do software (...) pois mostra o passo a passo, então nesse ponto eu acho que ela ajuda a (...) pensar na sequência de passos para alcançar o resultado do software” – P12</i>
<i>“É importante para todos da equipe visualizarem o que foi pensado para o sistema”</i>	<i>“(...) ajuda, por exemplo, a estruturar a ideia dos requisitos, de forma que o time de desenvolvimento consiga realmente entender a ideia...” – P7</i> <i>“...a descrição do caso de uso serve para o desenvolvedor ou para a equipe de desenvolvimento mostrar, como um todo, como vão desenvolver.” – P23</i>

5 Discussão dos Resultados

Nesta seção, são discutidos os resultados obtidos baseado nas categorias resultantes da análise apresentada na seção anterior. Além disso, são apresentadas sugestões para melhorar o aprendizado e a prática de UC, com base nas dificuldades definidas para o modelo.

Para responder à questão de pesquisa, as dificuldades percebidas pelos participantes ao especificar UC são discutidas nas categorias definidas para o modelo. Além disso, os resultados dos estudos relacionados são comparados com as evidências encontradas neste estudo.

Primeiramente, os benefícios percebidos da utilização da especificação de UC no desenvolvimento de software estão relacionados ao detalhamento do comportamento do UC (através dos fluxos principal, alternativos e exceções). A especificação de UC ajuda a estruturar a ideia do UC, facilitando o entendimento da funcionalidade que será desenvolvida pela equipe de projeto. Esses resultados são similares aos de Nascimento et al. [13], pois evidenciam que a especificação do UC foi útil para detalhar e organizar os cenários do UC.

Na categoria “**Abstração do Requisito**”, as dificuldades encontradas neste estudo são semelhantes aos de Nascimento et al. [13], pois evidenciam que os participantes tiveram dificuldades para entender o requisito e seguir os passos corretamente para a especificação do UC [13]. Dessa forma, as dificuldades identificadas nesta categoria podem afetar outras categorias do modelo, como “**Fluxos do UC**” e “**Regras de Negócio**”, uma vez que é necessário que os requisitos do sistema sejam compreendidos, a fim de transcrevê-los corretamente na especificação do UC. Além disso, no contexto real, a definição correta dos fluxos e das RNs são necessários para a completude da especificação do UC.

Na categoria “**Fluxos do UC**”, os resultados deste estudo são semelhantes as dificuldades apresentadas em Bolloju [5] e Nascimento et al. [13]. Bolloju [5] afirma que houve dificuldades em identificar e descrever os fluxos e o estudo de Nascimento et al. [13] reforça os resultados de Bolloju. Nesta categoria, também ocorreram dificuldades mais específicas, como descrever os passos da interação do ator e sistema nestes fluxos. Estas dificuldades podem impactar no detalhamento do comportamento do UC, pois a organização e descrição dos fluxos são importantes para obter a compreensão de toda a funcionalidade.

Na categoria “**Regras de Negócio**”, as dificuldades ocorreram quando os participantes foram descrever as RNs no UC. Por exemplo, os alunos confundiram a descrição das RNs com a descrição dos RNFs ou confundiram com a descrição dos fluxos do UC. Esta confusão entre RNs e RNF pode estar relacionada ao fato do aluno ainda possuir dúvidas sobre restrições do sistema, uma vez que uma RN foca nas restrições do domínio do problema e RNFs estão relacionados ao nível em que atributos de qualidade impactam no atendimento às funcionalidades providas pelo sistema.

Como forma de reduzir ou evitar as dificuldades descritas, sugere-se a adoção de algumas práticas recomendadas pela literatura, como forma de melhorar a aprendizagem dos alunos. A primeira prática sugerida é a inclusão de inspeção durante as aulas, por ser um meio eficiente que auxilia na identificação de defeitos e também ajuda a melhorar a qualidade dos documentos de software [3]. A inclusão da inspeção auxilia os alunos a relembrar os conceitos ensinados sobre UCs, a compreender os principais tipos de defeitos nas especificações e a evitar que estes sejam repetidos em outras modelagens. Outra prática que pode ser adotada é ensinar utilizando anti-padrões [4], ou seja, apresentar aos alunos más soluções de modelagem. Assim como na inspeção, adotar anti-padrões durante as aulas sobre UCs melhora a capacidade dos alunos de identificar os problemas de modelagem, entender a forma errada para solucionar determinado cenário e desenvolver soluções corretas. Outra prática que pode ser utilizada é especificar uma funcionalidade a partir de um software já desenvolvido [15]. A adoção desta prática pode auxiliar os alunos a visualizar melhor os vários cenários (fluxos alternativos e exceções) que compõem um UC e a entender a importância de alguns conceitos, por exemplo, as pré e pós-condições e as regras de negócio.

Espera-se que a identificação das categorias de dificuldades apoie o processo de ensino-aprendizagem, além de motivar a melhoria de técnicas e abordagens no ensino/aprendizagem da especificação de UC, para que os alunos possam desenvolver diferentes habilidades, como abstração de diferentes contextos de sistemas e na escrita da especificação do UC. Assim, as dúvidas e dificuldades relacionadas as categorias do modelo poderão ser reduzidas e corrigidas, preparando os alunos para enfrentarem os desafios no desenvolvimento de software com UC na indústria.

6 Ameaças à Validade

Neste estudo tiveram algumas ameaças que podem afetar a validade dos resultados. Nesta seção, são discutidas as principais ameaças que podem ser classificadas em: validade interna, validade externa, validade de constructo e confiabilidade [16] [22].

Validade Interna: Neste estudo foram consideradas duas principais ameaças que representam um risco de interpretação imprópria dos resultados: (I) influência e experiência dos moderadores; e (II) participantes não relatarem dificuldades. Em relação a ameaça (I), poderá ocorrer um efeito diferente nos resultados das entrevistas, pois estas foram individuais e conduzidas por dois pesquisadores separadamente. Contudo, o segundo pesquisador acompanhou a condução da entrevista piloto juntamente com o primeiro autor. Esse acompanhamento permitiu que as conduções das entrevistas fossem realizadas de modo semelhantes. Em relação a ameaça (II), para evitar que o participante não relatasse nenhuma dificuldade, os pesquisadores solicitavam que o participante exemplificasse alguma dúvida que teve durante a especificação de UC. Isto foi possível, pois os participantes estavam usando UC em trabalhos práticos desenvolvidos durante a disciplina ou em projetos de software que estavam participando e/ou já haviam atuado.

Validade Externa: Neste estudo, considerou-se como ameaça a comparação com outros contextos, devido existirem fatores relacionados ao background dos participantes e a grade curricular dos cursos nas universidades. No entanto, os diferentes perfis dos participantes ajudaram a capturar diferentes visões sobre as dificuldades em especificação de UCs. Além disso, os cursos apresentam disciplinas semelhantes na qual os participantes desenvolveram projetos de software com UCs.

Validade de Constructo: Neste tipo de ameaça, considerou-se o roteiro de entrevistas induzir o participante a ter resultados favoráveis ao pesquisador. Contudo, o roteiro de entrevista era semiestruturado e, portanto, continha perguntas abertas para capturar as dificuldades na especificação de UC. Para minimizar esta ameaça, o roteiro de entrevista foi avaliado por três pesquisadores. Após a avaliação, realizou-se uma entrevista piloto com um estudante de graduação com experiência prévia em especificação de UC, a fim de verificar se o roteiro atingiria seu objetivo. Os resultados da entrevista foram avaliados em conjunto com outro pesquisador e não foi preciso realizar melhorias no roteiro.

Confiabilidade: Neste estudo, considerou-se como ameaça o viés do pesquisador na interpretação dos dados qualitativos. Para tratar esta ameaça, outro pesquisador com alto nível de conhecimento em análise qualitativa e especificação de UC acompanhou todas as etapas da análise dos dados qualitativos.

7 Conclusões e Trabalhos Futuros

Este artigo apresentou um estudo qualitativo que teve como objetivo investigar as dificuldades percebidas ao especificar casos de uso durante o desenvolvimento de software. Com base nos resultados da análise qualitativa, foi proposta uma versão inicial de um modelo que sumarizou as dificuldades em especificar UCs.

As dificuldades apresentadas no modelo foram classificadas nas seguintes categorias: Abstração dos Requisitos, Pré e Pós-Condições, Fluxos do UC, Regras de Negócio e Referências de Fluxos e RNs. Percebeu-se que existem dificuldades no modelo que são comuns tanto para estudantes que estão aprendendo, quanto para os estudantes que possuem experiência em especificar UCs. Estas dificuldades podem estar relacionadas a (i) falta de aprendizado prático da descrição textual de UCs e a (ii) falta de informações do sistema, necessárias para a descrição do UC. Além disso,

percebeu-se que algumas dificuldades ocorreram apenas com os estudantes que não possuíam experiência na indústria, tais como dificuldades na identificação e descrição dos fluxos (principal e alternativos) e confundir RNs com os RNFs do sistema. Já os estudantes que possuíam experiência apresentaram dificuldades na descrição da interação do sistema com outros sistemas que interagem no UC e em referenciar as RNs nos passos dos fluxos do UC.

Espera-se que as dificuldades sintetizadas no modelo, sirvam de base para futuras pesquisas na área, bem como no apoio à sugestão de práticas para melhorar o processo de ensino/ aprendizagem dos alunos na especificação de UC.

Como trabalhos futuros, pretende-se executar novos estudos experimentais para identificar dificuldades na modelagem do diagrama de UC, devido o foco principal deste estudo ser na especificação textual e não no diagrama de UC. Assim, será possível investigar as dificuldades que os alunos encontram durante o uso de UC (diagrama e especificação) para aprimorar o modelo e comparar com as dificuldades já identificadas, além de validá-lo em próximos estudos. Além disso, é preciso investigar a correlação entre os tipos de dificuldades percebidas e seu desempenho real em projetos de desenvolvimento de software, assim como investigar outros fatores que influenciam as dificuldades apresentadas no modelo, como o contexto do sistema e conhecimento/ habilidade técnica do participante.

Agradecimentos

Os autores agradecem o apoio financeiro da CAPES, através do processo 175956/2013, e da FAPEAM através do processo 062.00578/2014. Agradecemos também aos participantes do estudo das Universidades UFRJ e UFAM e aos pesquisadores do ESE/COPPE-UFRJ e do USES/UFAM pelas contribuições na execução deste estudo. Um agradecimento especial ao professor Guilherme Horta Travassos pelo apoio e colaboração durante a condução deste estudo.

Referências

1. Anda, B., Hansen, K., Gulleisen, I., Thorsen, H. Experiences from introducing UML-based development in a large safety-critical project. In *Empirical Software Engineering* v.11, n.4, pp. 555–581 (2006).
2. Anda, B., Hansen, K., Sand, G. An investigation of use case quality in a large safety-critical software development project. In *Information and Software Technology*, v. 51, n.12, pp. 1699–1711 (2009).
3. Anda, B.; Sjøberg, D. Towards an inspection technique for use case models. In *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE '02)*, ACM, NY, USA (2002), pp. 127–134.
4. Balaban, M., Maraee, A., Sturm, A., Jelnov, P. A pattern-based approach for improving model quality. In *Software & Systems Modeling*, v. 14, n.4, pp. 1527-1555 (2015).
5. Bolloju, N. Exploring Quality Dependencies among UML Artifacts Developed by Novice Systems Analysts. In *12th Americas Conference on Information Systems (AMCIS 2006)* pp. 472 (2006).
6. Cockburn, A. *Writing Effective Use Cases*, vol. 1, Addison-Wesley, Boston (2001).

7. Dobing, B., Parsons, J. How UML is used. In *Communications of the ACM*, v. 49 n.5, pp. 109-113 (2006).
8. Halpin, T. Business rule modality. In *International Conference on Advanced Information Systems Engineering (CAiSE'06 Workshops)*, pp. 383-94 (2006).
9. Jacobson, I., Booch G., Rumbaugh J. *The Unified Software Development Process*, Addison-Wesley, New York (1998).
10. Linehan, M. H. SBVR use cases. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, Springer Berlin Heidelberg, pp. 182-196 (2008).
11. Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., Clause, J. An empirical study of practitioners' perspectives on green software engineering. In *38th International Conference on Software Engineering*, pp. 237-248 (2016).
12. Misbhauddin, M., Alshayeb M., Extending the UML use case metamodel with behavioral information to facilitate model analysis and interchange. In *Software & Systems Modeling*, v. 4, n.2, pp. 813-838 (2015).
13. Nascimento, E. S.; Silva, W., Conte, T. U., Steinmacher, I., Massollar, J., Travassos, G. H. Is a Picture worth a Thousand Words? A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases. In *30th Brazilian Symposium on Software Engineering (SBES '16)*, pp. 93-102 (2016).
14. Nascimento, E. S., Silva, W. A. F., França, B. B. N. D.; Gadelha, B; Conte, T. U. Relatório Técnico: Um Modelo sobre as Dificuldades para Especificar Casos de Uso, Relatório de número 0007 (2017). Disponível em: <https://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.
15. Pavlov, V. L., Boyko, N., Babich, A., Kuchaiev, O., Busygin, S. Applying pantomime and reverse engineering techniques in software engineering education. In *Frontiers In Education*, pp. T1E-1-T1E-5 (2007).
16. Petersen, K.; Gencel, C. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In *Joint Conference of the 23rd International Workshop on Software Measurement and the Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, pp. 81-89 (2013).
17. Rago, A., Marcos, C., Diaz-Pace, J. Uncovering quality-attribute concerns in use case specifications via early aspect mining. In *Requirements Engineering*, v.18, n.1, pp. 67-84 (2013).
18. Runeson, P., Host, M. Guidelines for conducting and reporting case study research in software engineering. In *Empirical Software Engineering*, 14, n. 2, p. 131-164 (2008).
19. Savic, D., Antovic, I., Vlajic, S., Stanojevic, V., Milic, M. Language for use case specification. In *34th IEEE Software Engineering Workshop (SEW)*, pp. 19-26 (2011).
20. Strauss, A., Corbin, J. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 2 ed. SAGE Publications, London (1998).
21. Tiwari S., Gupta A., A systematic literature review of use case specifications research. In *Information and Software Technology*, v. 67, pp. 128-158 (2015).
22. Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. *Experimentation in software engineering*. Springer Science & Business Media (2012).
23. Yue, T., Briand, L. C., Labiche, Y. A use case modeling approach to facilitate the transition towards analysis models: Concepts and empirical evaluation. In *International Conference on Model Driven Engineering Languages and Systems*. Springer Berlin Heidelberg, pp. 484-498 (2009).