

# Is a Picture worth a Thousand Words? A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases

Elizamary Nascimento,  
Williamson Silva, Tayana Conte  
ICOMP – UFAM  
Manaus, AM – Brazil  
{elizamary.souza,williamson,  
silva,tayana}@icomp.ufam.edu.br

Igor Steinmacher  
DACOM - UFTPR  
Campo Mourão, PR – Brazil  
igorfs@uftpr.edu.br

Jobson Massollar,  
Guilherme Horta Travassos  
COPPE - UFRJ  
Rio de Janeiro, RJ – Brazil  
{jobson, ght}@cos.ufrj.br

## ABSTRACT

Use cases specifications are artifacts employed in all stages of software development, from the requirements elicitation to implementation. During this process, issues related to ambiguity, redundancy, inconsistency, and incompleteness can affect these specifications. These issues can harm software engineers' understanding and, consequently, affect the software quality. Given this context, this paper describes an empirical study to evaluate two different use cases specifications approaches (textual and graphical-based forms). We compared the approaches by assessing the degree of correctness and the time spent to generate the specifications. In addition, we performed an analysis focusing on evaluating the ease of use and usefulness of each approach. The quantitative results showed that textual form and graphical-based specifications presented similar levels of correctness and the time spent to generate them were also similar. The qualitative results indicated that the subjects had difficulties using both approaches; however, subjects stated that graphic-based specifications were easier and more useful to specify use cases.

## CCS Concepts

Software and its engineering/Software notations and tools/System description languages/Unified Modeling Language (UML)

## Keywords

Use cases specifications, UML, empirical study

## 1. INTRODUÇÃO

A representação dos requisitos de software é fundamental para o sucesso dos projetos [31]. Dentre as representações, casos de uso (*Use Cases* - UC) são um dos artefatos utilizados pelos engenheiros de software para descrever e documentar requisitos do software [26]. Os casos de uso foram originalmente propostos por Jacobson *et al.* [13] e são comumente utilizados para capturar as funcionalidades específicas de um software [26]. Além disso, os casos de uso auxiliam em diferentes atividades do processo de desenvolvimento, tais como, planejamento e estimativa, análise,

projeto, desenvolvimento e testes [4, 3, 11], sendo utilizados como meio de comunicação entre os membros da equipe de projeto e outros *stakeholders* [9]. Por estes motivos, casos de uso são comumente utilizados na indústria [1, 4, 5, 9, 10, 21].

Existem diferentes abordagens (formatos e estilos) que podem ser utilizadas para especificar casos de uso [26]. A escolha por uma dessas abordagens dependerá das necessidades do projeto de software e das características de sua equipe, por exemplo, o nível de experiência dos membros da equipe de projeto, o tamanho do projeto, a organização e a coesão [4]. Estas abordagens podem ser caracterizadas de duas formas: textual e gráfica.

Nas abordagens textuais, os casos de uso são normalmente apresentados como um conjunto de cenários, descritos em linguagem natural (narrativa) e obedecendo a uma estrutura sintática [4, 12]. Utilizando esse tipo de abordagem, os *stakeholders* familiarizados com a notação não necessitam de treinamento formal e os desenvolvedores não precisam de ferramentas de modelagem específica [1]. No entanto, o comportamento do caso de uso não é apresentado de forma direta e intuitiva [12]. Além disso, as especificações geradas podem apresentar diferentes defeitos, tais como: descrições ambíguas, que possuem diferentes interpretações [6]; e informações incompletas nas quais faltam descrições de fluxos ou regras necessárias para o entendimento completo do caso de uso [26].

As abordagens gráficas utilizam notações gráficas que auxiliam a representar os cenários do caso de uso [12], por exemplo, por meio de diagramas. Esse tipo de abordagem pode ser útil para mostrar de forma mais direta o comportamento de um caso de uso, ou seja, os aspectos dinâmicos (fluxos alternativos, opções do sistema, dentre outros) [12, 24]. Contudo, os engenheiros podem ter dificuldade em representar corretamente os requisitos do software no diagrama [20]. Um problema potencial é a omissão de fluxos/regras de negócios que devem ser descritos ou referenciados no diagrama [11, 20]. Com isso, as especificações de casos de uso geradas a partir da representação gráfica podem se tornar confusas, com descrições incompletas/ incorretas e de difícil entendimento [9, 24, 26].

Embora ambas as abordagens possuam elementos em comum, a abordagem textual é considerada adequada para detalhar, validar e testar os requisitos do software [24, 26] e a abordagem gráfica (diagramas) é adequada para visualizar o comportamento global do caso de uso e detalhar os aspectos dinâmicos relacionados à funcionalidade do software [12]. No entanto, não há consenso sobre qual abordagem é mais eficiente em diferentes contextos, assim como não há uma definição sobre o grau de formalismo mais

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBES '16, September 19 - 23, 2016, Maringá, Brazil

Copyright is held by the owner/author(s).

Publication rights licensed to ACM.

ACM 978-1-4503-4201-8/16/09\$15.00

DOI: <http://dx.doi.org/10.1145/2973839.2973855>

adequado para essas abordagens [26]. Se as abordagens forem utilizadas corretamente e os casos de uso forem bem especificados, eles especificam os requisitos do software de forma completa, correta e de fácil compreensão pelos engenheiros de software e *stakeholders* [4]. Especificar casos de uso de forma clara, correta e coerente é importante para se obter um produto de software com qualidade [28]. Logo, é importante investigar e comparar a facilidade (de aprendizado e uso) e a utilidade dessas duas formas de especificar casos de uso.

Com o objetivo de compreender a influência destas diferentes representações de casos de uso, este artigo descreve a condução de um estudo experimental realizado com o objetivo de avaliar duas diferentes abordagens para especificação de casos de uso (uma textual e uma gráfica). Para a abordagem textual, escolheu-se a abordagem proposta por Cockburn [9] (com algumas adaptações), pois esse é um dos formatos mais citados na literatura [26]. Com relação a abordagem gráfica, foram selecionadas abordagens que se baseiam em diagramas de atividades da UML [23], por ser um dos diagramas mais utilizados para representar o comportamento dos UCs [12, 30]. Em seguida, essas abordagens foram avaliadas, sendo a baseada no UCMModel [19] selecionada por oferecer uma estrutura sintática e semântica do UC e apoiar procedimentos de testes funcionais. Além disso, as duas abordagens (textual e gráfica) utilizam os elementos propostos por Jacobson *et al.* [14] para descrever casos de uso. Para observar a influência de uma ou outra abordagem de representação, verificou-se qual o grau de correteude das especificações geradas a partir dos casos de uso e o tempo gasto por cada participante ao usar ambas as abordagens. A análise da correteude das especificações foi realizada utilizando um método chamado Avaliação de Atributos de Verificação dos Modelos de UC [4]. Foi realizada uma análise qualitativa das percepções dos participantes em relação à facilidade (de aprendizado e uso), a utilidade e a preferência em relação às abordagens utilizadas para especificar um caso de uso. Com base nos resultados, identificou-se qual formato é mais fácil de aprender e de ser usado para construir uma especificação de caso de uso. Adicionalmente, foram identificadas as dificuldades, os pontos positivos e negativos observados pelos participantes.

O restante deste artigo está organizado da seguinte forma: a Seção 2 aborda conceitos sobre as formas de especificação de casos de uso utilizando a abordagem textual e a abordagem gráfica (baseada no *UCModel*). A Seção 3 descreve o planejamento e execução do estudo experimental. A Seção 4 apresenta os resultados quantitativos e qualitativos obtidos neste estudo. E a Seção 5 discute os resultados e as considerações finais.

## 2. ESPECIFICAÇÃO DE CASOS DE USO

A especificação de um caso de uso tem como propósito representar “a especificação de um conjunto de ações realizadas por um sistema que produz um resultado observável que é, normalmente, de valor para um ou mais atores ou outros *stakeholders* do sistema” [23]. Existem diferentes abordagens para representar os casos de uso e estes podem ser especificados em diferentes formatos [5, 26]. A escolha por uma ou outra abordagem depende de vários aspectos, como a experiência dos membros da equipe, a visibilidade do comportamento do caso de uso e o grau de formalismo [15]. Entre as abordagens utilizadas estão as que utilizam o formato textual e o formato gráfico. Na literatura foi encontrado apenas um trabalho que apresenta os pontos fortes e as limitações das formas de representar UCs utilizando a abordagem textual e gráfica (diagrama de atividades) [8]. No entanto, esse trabalho não apresenta um estudo experimental comparando as duas abordagens. A seguir é

apresentada uma breve descrição de cada abordagem para especificar casos de uso.

### 2.1 Abordagem Textual

A primeira abordagem para especificar um caso de uso foi proposta por Jacobson *et al.* [14], que define vários atributos do caso de uso: nome do caso de uso, pré-condições, pós-condições, fluxo principal, fluxos alternativos e campos de fluxos alternativos para especificar as necessidades dos requisitos. Essa abordagem é amplamente aceita para especificar os requisitos funcionais do software [24] por: (a) ser um formato menos formal e simples, (b) possuir uma estrutura pré-definida que pode ser utilizada de acordo com o contexto do software, [8, 14], e (c) proporcionar flexibilidade aos engenheiros de software para especificar os requisitos [24].

Para exemplificar a descrição textual de um caso de uso, a Tabela 1 apresenta a especificação do caso de uso “Concluir compra” de um sistema de compras via *Web*. Essa abordagem é baseada no formato proposto por Cockburn [9] que utiliza os mesmos elementos propostos por Jacobson *et al.* [14]. A primeira coluna apresenta os elementos utilizados para descrever o caso de uso e a segunda coluna apresenta as informações que o caso de uso deverá atender. Os elementos da primeira coluna são: (1) **Nome do UC**: nome do caso de uso, (2) **Descrição**: uma breve descrição sobre o objetivo do caso de uso, (3) **Atores**: nome dos atores que participam do caso de uso, (5) **Pré-condição**: descrição textual das condições que deverão ser atendidas para iniciar o caso de uso, (6) **Pós-condição**: descrição textual das condições que deverão ser atendidas após a execução do caso de uso, (7) **Fluxo Principal**: uma sequência de ações numeradas que descreve o caminho principal do caso de uso, sem considerar nenhum erro. (8) **Fluxos Alternativos**: uma sequência de ações numeradas que representa um caminho alternativo do caso de uso, (9) **Fluxos de Exceção**: uma sequência de ações numeradas que descreve um comportamento dos casos de uso quando algo inesperado acontece e (10) **Regras**: são as regras de negócio ou as restrições do caso de uso que devem ser consideradas em sua execução.

Tabela 1: Exemplo de um caso de uso, adaptado de [19]

Nome do UC	Concluir Compra
<b>Descrição:</b>	Permiti ao cliente fechar a compra, realizar o pagamento e obter o número do pedido.
<b>Atores:</b>	Cliente
<b>Pré-condição:</b>	Cliente deve estar autenticado
<b>Pós-condição:</b>	Cliente realiza o pagamento e obtém o número do pedido
<b>Fluxo Principal:</b>	1. O sistema lista os produtos que constam no carrinho de compras e disponibiliza as opções “Efetuar pagamento” e “Excluir item” [A1]; 2. O cliente seleciona a opção “Efetuar pagamento”; 3. O sistema calcula o valor do pedido, apresenta o valor do pedido e solicita os dados de pagamento [R1, R2]; 4. O cliente fornece os dados de pagamento; 5. O sistema valida os dados de pagamento, gera o pedido e apresenta o número do pedido;
<b>Fluxos Alternativos:</b>	A1 – Excluir item 1. O cliente seleciona um item da lista de produtos no carrinho de compras e seleciona a opção “Excluir item”; 2. O sistema exclui o item selecionado e retorna para o passo 1 do fluxo principal;
<b>Fluxos de Exceção:</b>	Não aplicável
<b>Regras:</b>	R1 – É obrigatório que o cliente tenha um desconto de 10% no valor do pedido se ele tiver mais de 1000,00 em compras nos últimos 30 dias. R2 – É obrigatório que o frete seja gratuito se o valor do pedido for maior que 500,00.

## 2.2 Abordagem Gráfica

Diagramas de atividades da UML podem ser utilizados para especificar graficamente os casos de uso [2, 12, 20]. Massollar *et al.* [19] propuseram um meta-modelo (*UCModel*) que utiliza o diagrama de atividades para especificar de forma gráfica os aspectos comportamentais do caso de uso. Este meta-modelo permite a estruturação da especificação dos casos de uso segundo um conjunto de critérios e restrições bem definidos, visando aprimorar o grau de formalismo das especificações dos casos de uso [19]. O *UCModel* estende o diagrama de atividades da UML [23] acrescentando novos elementos para a descrição do comportamento do caso de uso. Desses novos elementos adotados, há três tipos que representam as ações (interação) do ator e do sistema, representados através de estereótipos no diagrama [19]: (a) `<<actor_action>>`: representa uma ação do ator com o sistema na qual este faz uma requisição ao sistema informando os dados necessários; (b) `<<system_action>>`: representa uma ação do sistema em que os resultados gerados não são diretamente observados pelo ator; e (c) `<<system_response>>`: representa uma ação do sistema em que os resultados são direta ou indiretamente observados pelo ator. Essa ação pode representar os resultados gerados anteriormente ou pode solicitar outros dados ao ator.

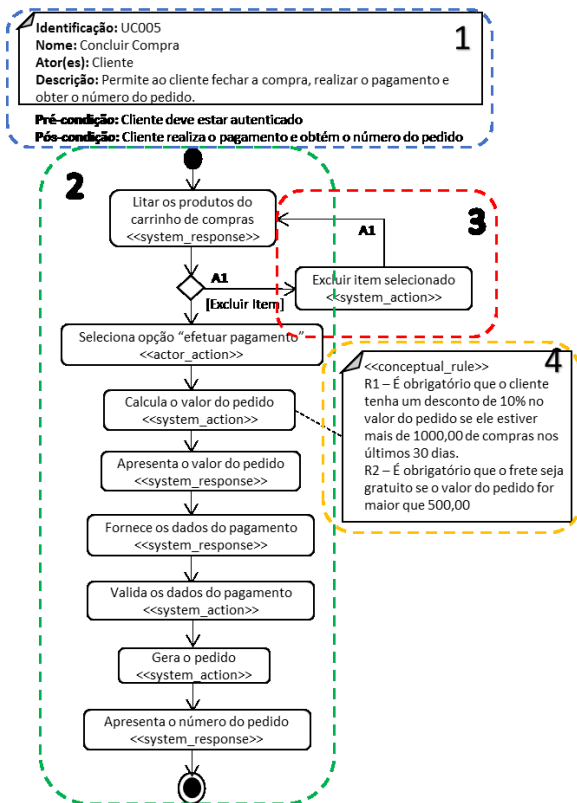


Figura 1: Exemplo de uma especificação de caso de uso utilizando *UCModel*, adaptado de Massollar *et al.* [19]

Outros elementos utilizados são os tipos de regras associadas ao caso de uso. Essas regras devem estar sempre associadas às ações que a afetam. Os estereótipos que representam essas regras são [19]: `<<conceptual_rule>>`: regra relacionada a perspectiva de conceituação – restringe as ações do sistema, estabelecendo condições para sua execução; `<<navigation_rule>>`: regra relacionada a perspectiva de navegação – restrições de quais caminhos podem ser navegados e quais informações serão solicitadas ao ator; `<<presentation_rule>>`: regra relacionada a

perspectiva de apresentação – restrições de como as informações serão apresentadas ao ator. A Figura 1 apresenta a abordagem gráfica representando uma especificação de um caso de uso gerados partir do *UCModel*.

O item 1 da Figura 1 representa o conjunto de elementos que define o caso de uso (identificação, nome, atores, descrição, pré e pós-condições). O item 2 mostra o fluxo principal que contém uma sequência de ações que começa do nó inicial até o nó final. Cada ação é identificada pelos estereótipos que representam a ação do ator e sistema: `actor_action`, `system_action` e `system_response`. O item 3 da Figura 1 mostra o fluxo alternativo identificado pelo nó de decisão. Neste ponto do diagrama, existe um caminho alternativo que pode ser executado dependendo da condição associada ao fluxo alternativo. No exemplo, o fluxo alternativo é identificado por “A1”. O item 4 da Figura 1 refere-se as regras de negócio que estão representadas pelos identificadores R1 e R2 usando os comentários no diagrama de atividades e classificados com os tipos de regras. Além disso, cada regra de negócio está associada à ação que será impactada por ela.

## 3. ESTUDO EXPERIMENTAL

Com o propósito de comparar especificações de casos de uso (utilizando as abordagens textual e gráfica) foi realizado um estudo experimental para comparar as abordagens em termos de: (a) menor tempo para especificar um caso de uso; e (b) maior grau de correteza da especificação gerada.

### 3.1 Planejamento do Estudo

O objetivo deste estudo experimental é apresentado na Tabela 2 usando o paradigma GQM [7]. A versão completa do pacote experimental encontra-se disponível em um relatório técnico [22].

Tabela 2: Objetivo do estudo usando o paradigma GQM [7]

<b>Analisar</b>	As abordagens textual e gráfica para especificação de casos de uso
<b>Com o propósito de</b>	Caracterizar
<b>Com respeito à</b>	Tempo gasto, correteza na especificação dos casos de uso e percepção sobre facilidade de uso e utilidade
<b>Do ponto de vista</b>	Dos pesquisadores em engenharia de software
<b>No contexto de</b>	Especificação de casos de uso de uma aplicação real por estudantes de graduação dos cursos: Ciência da Computação e Sistemas de Informação da UFAM

#### 3.1.1 Hipóteses

O estudo foi planejado e conduzido a fim de colocar à prova as seguintes hipóteses (nula e alternativa, respectivamente) em relação aos indicadores quantitativos:

- H01: O tempo para utilização das abordagens textual e gráfica para especificar um caso de uso é igual.
- HA1: O tempo necessário para especificar um caso de uso com a abordagem textual é diferente da abordagem gráfica.
- H02: A correteza da especificação do caso de uso com a abordagem textual é igual a com a abordagem gráfica.
- HA2: A correteza da especificação do caso de uso com a abordagem textual é diferente da abordagem gráfica.

#### 3.1.2 Contexto

O estudo foi executado com requisitos, diagramas de caso de uso e *mockups* de um projeto real para um sistema Web de Agendamento de Veículos. O cenário utilizado para especificar o caso de uso está relacionado com a confirmação do condutor após a execução do serviço de agendamento de veículo. O objetivo é permitir a um condutor pesquisar no sistema seus agendamentos e escolher um

deles para concluir. Ao escolher um agendamento o condutor informa os dados de conclusão. O estudo foi conduzido com estudantes de graduação do curso de Ciência da Computação e Sistema de Informação (segundo semestre de 2015) da Universidade Federal do Amazonas (UFAM). Os alunos haviam cursado a disciplina introdutória de “Engenharia de Software” e estavam cursando a disciplina de “Análise e Projeto de Sistemas”.

### 3.1.3 Variáveis de Seleção

As variáveis independentes foram as abordagens para especificação do caso de uso (tratamentos: gráfica e textual). As variáveis dependentes foram os indicadores de tempo e corretude. O tempo é calculado em horas e representa o tempo total gasto por cada participante para especificar o caso de uso utilizando uma abordagem. O indicador corretude verifica o grau de quão correto os casos de uso foram elaborados. Para isso, foram verificados o número de defeitos nos casos de uso criados pelos participantes, utilizando os itens de verificação apresentados na Tabela 4.

### 3.1.4 Participantes

Quarenta e quatro (44) participantes assinaram o formulário de consentimento e preencheram o formulário de caracterização. Este formulário captura dados que permitem avaliar o perfil dos participantes em relação à experiência em especificação de casos de uso e conhecimento em modelagem de sistemas. Os dados informados foram usados para caracterizar o perfil dos participantes em relação à experiência: nenhuma (N), baixa (B), média (M) e alta (A). É considerado um participante com alta experiência ter participado em mais de 5 projetos de software na indústria onde modelou diagramas da UML e atuou como analista na especificação de casos de uso. Um participante com média experiência ter participado de 1 a 4 projetos na indústria onde modelou diagramas da UML e especificou casos de uso. Um participante com baixa experiência ter participado de pelo menos um 1 projeto onde modelou diagramas da UML e especificou casos de uso em sala de aula, com nenhuma experiência o que não tem nenhum conhecimento em modelagem de diagramas da UML ou possui alguma noção sobre a linguagem UML através de leituras/palestras/tutoriais, mas sem experiência prática ou não possui nenhuma experiência prévia em especificação de casos de uso. A Tabela 5 (colunas CI e CII) apresenta a caracterização do perfil dos participantes.

### 3.1.5 Artefatos

Os instrumentos utilizados neste estudo experimental foram: o Termo de Consentimento Livre e Esclarecido (TCLE), o formulário de caracterização, os diagramas de casos de uso, os requisitos da aplicação, os *mockups* (protótipos de baixa fidelidade do caso de uso), instruções para orientar na modelagem com a abordagem gráfica (*UCModel*) e questionários pós-modelagem. Todos os artefatos do estudo foram validados por três autores deste artigo e pelos desenvolvedores do sistema.

### 3.1.6 Treinamento

Todos os participantes receberam treinamento de 2 horas em um mesmo ambiente sobre as duas abordagens de especificação (gráfica e textual). Durante o treinamento também foram realizados exercícios práticos de modelagem para especificar casos de uso com a abordagem gráfica e com a abordagem textual. Este tempo foi suficiente para apresentar as abordagens e exemplificar seu uso.

### 3.1.7 Projeto Experimental

O *design* aplicado no estudo experimental foi *um fator com dois tratamentos* [29]. Com base nos dados coletados através do formulário de caracterização (princípio de *design* de balanceamento), os participantes foram divididos em grupos (princípio de *design* de agrupamento). Os participantes foram

atribuídos aleatoriamente aos grupos). Ambos os grupos especificaram o mesmo cenário. Cada grupo foi formado por 22 participantes. Cada grupo utilizou apenas um tratamento.

### 3.1.8 Execução do Estudo

Os participantes foram alocados em salas diferentes, de acordo com o grupo em que foram atribuídos (ver Tabela 5). O Grupo 1 utilizou a abordagem gráfica e o Grupo 2 utilizou a abordagem textual. Cada participante recebeu individualmente os artefatos descritos na Subseção 3.1.5. Todos os participantes (Grupo 1 e Grupo 2) incluíram o tempo total gasto para a elaboração do caso de uso. Após o experimento, foi aplicado um questionário pós-modelagem visando capturar a percepção dos participantes após o uso da abordagem gráfica. Para isto, apenas o grupo 1 respondeu ao questionário pós-modelagem devido ao foco principal de observação ser o uso da abordagem gráfica. Vale ressaltar que durante o processo de construção dos casos de uso não foi permitida a comunicação entre os participantes e também os participantes não receberam nenhum auxílio dos pesquisadores envolvidos.

## 3.2 Resultados da Análise da Corretude dos Casos de Uso gerados pelas Abordagens

Para avaliar a corretude das especificações geradas pelos participantes do estudo, primeiramente três pesquisadores realizaram a análise dos 16 métodos encontrados na revisão sistemática realizada por Tiwari e Gupta [26]. Esses métodos têm como objetivo avaliar a qualidade de especificações de casos de uso. Após essa análise, os métodos propostos por Phalp *et al.* [24] e por Anda *et al.* [4] foram selecionados. Esses dois métodos foram escolhidos por apresentar itens de verificação que auxiliam a avaliação da qualidade das especificações de casos de uso. Para investigar qual dos métodos auxiliava a identificação do maior número de defeitos nas especificações, foram utilizadas quatro especificações de casos de uso, duas para cada abordagem. A avaliação das especificações foi realizada por um pesquisador e, posteriormente, conferida e discutida por outros dois pesquisadores. O método escolhido foi o de Anda *et al.* [4], denominado “Atributos de Verificação dos Modelos de UC”. Esse método foi escolhido, pois apresenta diretrizes que auxiliam de forma mais detalhada quais informações as abordagens de UCs devem conter para atender aos requisitos de qualidade de um caso de uso. Além disso, a taxonomia proposta pelos autores auxilia na verificação de defeitos do caso de uso de forma guiada. Para cada atributo de verificação são definidas diretrizes que os casos de uso devem seguir. Com isso, a execução da avaliação se torna mais direcionada e orientada para identificar os defeitos da especificação de um caso de uso [4].

Após a definição do método a ser utilizado para avaliar a corretude das especificações de casos de uso geradas no estudo, retirou-se a referência dos participantes, identificando-os com um código. O processo de avaliação das especificações foi realizado em duas etapas. Na primeira etapa um pesquisador avaliou as especificações que os participantes elaboraram a partir dos artefatos entregues (casos de uso, requisitos e os *mockups*). Na segunda etapa, dois pesquisadores revisaram por completo a avaliação realizada.

Durante a avaliação dos casos de uso, houve a necessidade de incluir novos itens para complementar o *Checklist* de verificação proposto por Anda *et al.* [4]. Por exemplo, no atributo de verificação “Incorretude da funcionalidade”, os novos itens serviram para identificar fluxos alternativos, de exceção ou regras de negócio que estavam descritas de forma incompleta. Esses novos itens, estão sinalizados em negrito na Tabela 3 (coluna “Definição e Execução”). Foram utilizados os seguintes atributos de verificação, baseados em Anda *et al.* [4]: Incorretude, Redundância, Falha de Integridade, Inconsistência, Ambiguidade, Falha de Legibilidade, Falha de Nível de Abstração, Falha na

Manutenibilidade e Inverificável. Além disso, foram criadas categorias para os principais tipos de defeitos (ver Tabela 4). Cada defeito possui um grau de severidade. Por exemplo, na Tabela 4, o defeito “15) Regras Incompletas” está associado a categoria “Incorretude”, e classificado ao grau de severidade “Médio”. A severidade do tipo “Grave” foi utilizada para classificar defeitos de omissão de fluxos ou regras de negócio que não foram descritos no

caso de uso. Os defeitos de severidade “Média” foram utilizados para classificar fluxos ou regras de negócio que não foram descritos por completo, ou que foram descritos de forma incorreta, passos de fluxos que ficaram faltando, fatos incorretos ou informações estranhas descritas do caso de uso. Os defeitos de severidade “Baixa” identificaram aqueles defeitos que não prejudicavam a compreensão e entendimento do caso de uso.

**Tabela 3: Exemplo dos Atributos de Verificação de Modelos de Casos de Uso baseados em Anda *et al.* [4]**

Atributos de Verificação	Definição e Execução
1) Incorretude na funcionalidade	Os requisitos identificados devem ser representados corretamente sem informações que mudem as necessidades. Não deve conter: <ol style="list-style-type: none"> <li>1.1. atores com descrições incorretas;</li> <li>1.2. casos de uso que não são independentes ou tarefas distintas;</li> <li>1.3. dependências incorretas entre atores e casos de uso ou entre casos de uso; ou</li> <li>1.4. casos de uso com eventos ou fatos incorretos, fluxos alternativos incorretos ou pré ou pós-condições que não são realmente pré-requisitos para os estados iniciais e finais do caso de uso;               <ol style="list-style-type: none"> <li>1.4.1. fluxos (alternativos/exceção), regras de negócio descritos mas não foram referenciados em nenhum momento no caso de uso;</li> <li>1.4.2. referências incorretas de fluxos (alternativos/exceção) e regras de negócio que não foram descritos no caso de uso;</li> <li>1.4.3. fluxos (principal, alternativos/exceção) e regras de negócio incompletos, quando ficou faltando descrever algumas informações importantes para sua compreensão;</li> </ol> </li> </ol>
3) Falha de Integridade	Todos os requisitos implícitos devem ser atendidos pelo modelo de casos de uso e não deve faltar: <ol style="list-style-type: none"> <li>3.1. atores na forma de pessoas ou entidades externas que têm objetivos relacionados ao sistema;</li> <li>3.2. funcionalidades que não foram descritas como casos de uso ou que os objetivos dos atores que não possuem um caso de uso relacionado;</li> <li>3.3. entradas ou saídas para casos de uso;</li> <li>3.4. eventos que são necessários para a compreensão do caso de uso;</li> <li>3.5. fluxos alternativos; ou</li> <li>3.6. dependências entre o ator e um caso de uso ou entre casos de uso;</li> <li>3.7. <b>Faltou descrever o objetivo do caso de uso;</b></li> <li>3.8. <b>Faltou descrever algum elemento do modelo que está sendo utilizado como: identificador do caso de uso, nome do caso de uso, pré e pós-condições, regras de negócio;</b></li> </ol>
6) Falha de Legibilidade	As descrições de casos de uso devem ser corretas e utilizar uma terminologia comum. Não deve haver informações duplicadas, casos de uso muito grandes ou muito pequenos, ou suposição de como um ator irá executar as suas próprias tarefas internas. Cada caso de uso deve seguir um caminho lógico para um objetivo claro. <p><b>Faltou utilizar um nome ou termo para identificar um fluxo alternativo, de exceção ou regra de negócio, como por exemplo: A1, FA01 para Fluxo Alternativo 1; E1, FE01 para Fluxo de Exceção e RN01, RN ou Regra 1 para Regras de Negócio;</b></p>

## 4. ANÁLISE DOS RESULTADOS

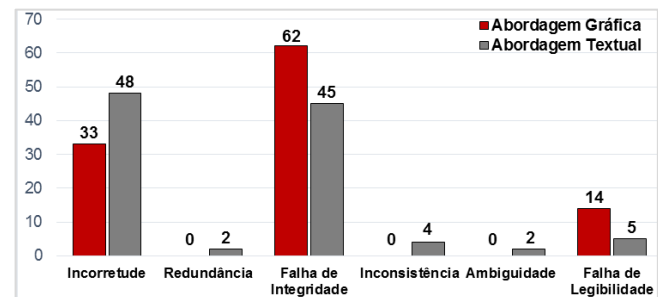
Nesta seção são apresentados os resultados quantitativos e qualitativos obtidos no estudo experimental. Na Subseção 4.1 são descritos os resultados quantitativos relativos ao tempo para construção e a corretude das especificações dos casos de uso gerados pelas duas abordagens. Os resultados qualitativos relativos à percepção dos participantes quanto à facilidade e utilidade de cada abordagem são apresentados na Subseção 4.2.

### 4.1 Resultados Quantitativos

A avaliação dos casos de uso resultou em 215 defeitos, dos quais, 109 foram identificados nos casos de uso criados utilizando a abordagem gráfica e 106 dentre aqueles criados utilizando a abordagem textual. A Figura 2 mostra o número dos diferentes tipos de defeitos identificados na avaliação dos casos de uso, classificados de acordo com os atributos de verificação [4].

O maior número de defeitos encontrados nas especificações geradas pela abordagem gráfica foi de Falha de Integridade (62 defeitos), seguido por Incorretude (33) e Falha de Legibilidade (14). Não foram identificados defeitos de Redundância, Consistência e Ambiguidade nas especificações geradas por essa abordagem. Nas especificações geradas pelos participantes que utilizaram a abordagem textual (Figura 2), os defeitos mais encontrados foram de Incorretude (48 defeitos), seguido pela Falha de Integridade (45), Falha de Legibilidade (05), Inconsistência (04), Redundância (02) e Ambiguidade (02). Nas duas abordagens, os defeitos de Incorretude (textual – 48 defeitos; e gráfica – 33

defeitos) estavam relacionados a descrições incompletas, por exemplo, pré e pós-condições, referências de fluxos alternativos, regras e outros casos de uso que não foram referenciados na descrição do caso de uso. Já os defeitos de Falha de Integridade (textual – 45 defeitos; e gráfica – 62 defeitos), estavam relacionados a omissão de fluxos ou regras que não foram descritos no caso de uso.



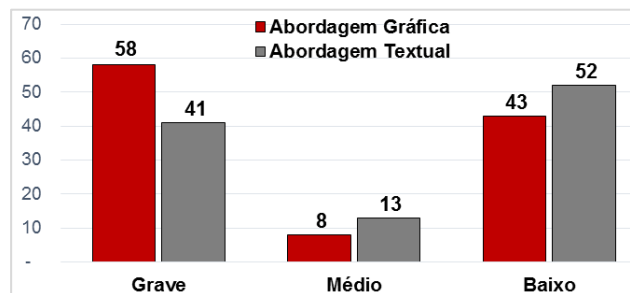
**Figura 2: Número de diferentes tipos de defeitos encontrados nas especificações de casos de uso**

A partir da classificação dos defeitos e o grau de severidade mostrados na Tabela 4, confrontou-se o número de defeitos encontrados por participante, que são apresentados na Seção 4.1.

**Tabela 4: Defeitos e Grau de Severidade**

Defeitos	CAT	GS
1. Pré e Pós-condições, Eventos Incompletos	Incorretude	B
2. Referências de Fluxos, Regras Incorretos	Incorretude	B
3. Faltou referenciar UCs/Fluxos/Regras no UC	Incorretude	B
4. Identificação de Atores e UCs Incorretos	Incorretude	B
5. Descrição do Caso de Uso Incorreta	Incorretude	B
6. Fluxos, Eventos que realizam a mesma tarefa	Redundância	B
7. Faltou identificar/descrever o ator, pré e pós-condição	Falha de Integridade	B
8. Fluxos com numeração incorreta	Inconsistência	B
9. Eventos Inconsistentes com o comportamento do UC	Inconsistência	B
10. Não utilizou uma terminologia comum	Falha de Legibilidade	B
11. Fluxos e Regras não claros (confusos)	Falha de Legibilidade	B
12. Descrições ambíguas nos eventos, Fluxos, Regras	Ambiguidade	M
13. Regras Inconsistentes com o comportamento do UC	Inconsistência	M
14. Fluxos incorretos com mais de um cenário	Incorretude	M
15. Regras Incompletas	Incorretude	M
16. Omissão de Passos (faltam passos em algum fluxo)	Incorretude	M
17. Fato Incorreto/Informações estranhas descritos nos Fluxos/Regras	Incorretude	M
18. Omissão de Fluxos/Regras	Falha de Integridade	G

**Legenda: Defeitos** – Descrição dos defeitos; **CAT** – Categoria do tipo de defeito; **GS** – Grau de Severidade; **B** – Defeito Baixo; **M** – Defeito Médio; **G** – Defeito Grave – **UC** – Caso de Uso



**Figura 3: Número de defeitos por grau de severidade**

A Figura 3 mostra o número de defeitos encontrados nas especificações geradas por cada abordagem distribuídas por grau de severidade. Os casos de uso especificados utilizando a abordagem gráfica tiveram o maior número de defeitos do tipo Grave (58), 17 defeitos a mais que a aqueles especificados com a abordagem textual (41). Isto mostra que nessas especificações não foram descritos os fluxos ou regras de negócio importantes para a completude do caso de uso. Em comparação com a abordagem textual, as especificações geradas com a abordagem gráfica obtiveram menor número de defeitos de grau de severidade Médio e Baixo.

A Tabela 5 sumariza os resultados relativos à quantidade de defeitos e o tempo gasto por participante para elaborar a especificação. As colunas P1 e P2 apresentam o código dos participantes que utilizaram, respectivamente, a abordagem gráfica e a textual. As colunas ND (1) e ND (2) apresentam respectivamente a quantidade de defeitos encontrados na avaliação

dos casos de uso gerados pelas abordagens gráfica e textual. As colunas T1 e T2 apresentam o tempo, em minutos, que cada participante utilizou para especificar o caso de uso. As colunas CI e CII apresentam os níveis de conhecimento, respectivamente, em modelagem e em especificação de casos de uso dos participantes.

**Tabela 5: Resumo do número de defeitos, tempo gasto e caracterização por participante**

Grupo 1 – Abordagem Gráfica					Grupo 2 – Abordagem Textual				
P1	CI	CII	ND (1)	T1 (min)	P2	CI	CII	ND (2)	T2 (min)
S1	B	B	4	60	S30	B	B	3	50
S2	B	B	7	64	S31	M	M	5	40
S3	B	B	9	45	S32	M	M	9	27
S4	B	B	6	45	S33	B	B	6	35
S5	B	B	4	59	S34	B	B	4	30
S6	B	B	4	58	S35	B	B	3	42
S7	B	B	4	28	S36	M	M	5	40
S8	B	B	2	47	S37	B	M	2	38
S9	B	B	5	41	S38	B	B	5	40
S10	M	B	5	53	S39	B	B	5	46
S11	B	B	6	62	S40	B	B	3	47
S12	B	M	6	65	S41	B	M	7	60
S13	B	B	4	81	S42	B	B	5	74
S14	B	B	6	63	S43	B	B	4	65
S15	B	B	7	45	S44	B	M	4	47
S16	B	M	4	57	S45	B	B	5	51
S17	B	B	5	50	S46	B	B	4	80
S18	M	M	5	55	S47	B	B	4	73
S19	B	B	3	59	S48	B	B	5	78
S20	B	B	2	56	S49	B	B	3	78
S21	B	B	6	60	S50	B	B	8	82
S22	B	B	5	65	S51	B	B	7	44

**Legenda:** P – Participantes; CI – Conhecimento em Modelagem UML; CII – Experiência em especificação com casos de uso; N – Nenhuma experiência; B – Baixa; M – Média; e A – Alta; ND (1) - Número de defeitos das especificações geradas pela forma gráfica; ND(2) - Número de defeitos das especificações geradas pela forma textual; T1 (min) - Tempo gasto em min. para especificar a forma gráfica; T2 (min) - Tempo gasto em min. para especificar com a forma textual.

Para analisar o tempo e a corretude obtida pelos grupos, foram realizadas análises estatísticas utilizando a ferramenta SPSS V.20<sup>1</sup>. Conduziu-se o teste de normalidade de Shapiro-Wilk com  $\alpha=0.05$  para tempo e corretude. O teste mostrou que a distribuição dos valores de tempo não é normal para ambos os grupos (com  $p=0,328$  para as especificações da abordagem gráfica e  $p=0,036$  para as especificações da abordagem textual) e que a distribuição dos valores de corretude não é normal (com  $p=0,318$  para as especificações da abordagem gráfica e  $p=0,093$  para as especificações da abordagem textual). Devido a esses resultados, utilizou-se o teste não-paramétrico Mann-Whitney.

A Figura 4 mostra o gráfico de boxplot com a distribuição de tempo dispendido pelos participantes por abordagem. Observou-se que a mediana do grupo que utilizou a abordagem gráfica é maior que a mediana do grupo que utilizou a abordagem textual. No entanto, quando as duas amostras foram comparadas utilizando o teste de Mann-Whitney, não foi possível encontrar diferença estatística significativa entre os dois grupos ( $p=0,341$ ). Esses resultados apoiam a H01 que afirma que *não há diferença entre utilizar a abordagem textual e abordagem gráfica para especificar um caso de uso com relação ao indicador de tempo*.

<sup>1</sup> IBM SPSS Software - [www.ibm.com/software/br/analytics/spss/](http://www.ibm.com/software/br/analytics/spss/)

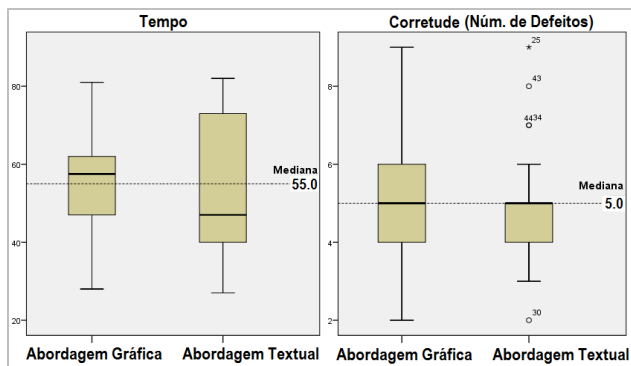


Figura 4: Boxplots do tempo e corretude por abordagem

A Figura 4 também mostra o gráfico de *boxplot* comparando a distribuição do número de defeitos (corretude) por abordagem. Notou-se que a corretude para os dois grupos é similar. O resultado do teste não-paramétrico mostrou que não há diferença estatisticamente significativa entre os grupos ( $p = 0,606$ ). Esses resultados apoiam a H02, que afirma que *não há diferença entre utilizar a abordagem textual e a abordagem gráfica para especificar um caso de uso com relação ao indicador de corretude*.

#### 4.1.1 Percepção sobre a abordagem gráfica

A fim de avaliar a percepção dos participantes quanto à facilidade (aprendizado e de uso) e à utilidade da abordagem gráfica, questionários pós-modelagem foram aplicados e analisados. Os participantes do Grupo 1 forneceram suas respostas em uma escala de seis pontos, baseados nos questionários aplicados por Lanubile *et al.* [17] (variando de concordo totalmente a discordo totalmente). Como sugerido por Laitenberger e Dreyer [16], não foi utilizado um valor neutro como intermediário, pois estes não oferecem informações sobre a tendência dos participantes (concordar ou discordar). Os itens avaliados estão na Tabela 6.

Tabela 6: Facilidade e Utilidade da abordagem gráfica

Facilidade	(Q1) Foi fácil aprender a especificar um caso de uso com esta abordagem?
	(Q2) Foi fácil construir a especificação do caso de uso utilizando esta abordagem?
Utilidade	(Q3) Utilizar esta abordagem aumentou meu entendimento sobre a interação do ator com o sistema?
	(Q4) A abordagem foi útil para especificar um UC?

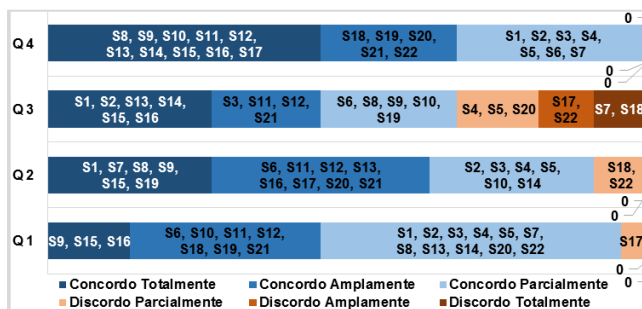


Figura 5: Percepção sobre a abordagem gráfica

A Figura 5 apresenta a percepção dos participantes com relação à facilidade de especificar casos de uso utilizando a abordagem baseada na forma gráfica. Na figura o eixo horizontal refere-se ao

grau de aceitação dos participantes. Nas barras foram inseridos códigos que representam os participantes (S1, S2, e assim sucessivamente) da Tabela 5.

Sobre a facilidade em aprender e utilizar a abordagem gráfica (Q1), apenas o participante S17 discordou parcialmente que seja fácil aprendê-la. Para a facilidade em construir casos de uso utilizando a abordagem gráfica (Q2), os participantes S18 e S22 discordaram parcialmente. O participante S22 teve dificuldades no início, pois se confundiu em relação aos diagramas da UML. Os participantes S7 e S18 discordaram totalmente (Q3) que essa abordagem seja útil para o entendimento da interação do caso de uso, destacando a dificuldade de entender a interação utilizando essa abordagem. E sobre a utilidade da forma gráfica para especificar casos de uso (Q4) não houve nenhuma discordância.

#### 4.1.2 Percepção sobre a preferência das abordagens para especificar casos de uso

Para uma análise comparativa sobre as abordagens, os participantes também responderam a três questões, apresentadas na Tabela 7.

Tabela 7: Preferências das abordagens para especificar UC's

Facilidade	(Q1) Qual a abordagem mais fácil para especificar um caso de uso?
Utilidade	(Q2) Qual a abordagem mais útil para especificar um caso de uso?
Empregabilidade	(Q3) Para especificar casos de uso de uma aplicação, qual a sua opção?

A Figura 6 apresenta a análise sobre a percepção dos participantes onde o eixo horizontal refere-se ao grau de preferência dos participantes sobre essas questões. Percebe-se que 64% (14 participantes) preferem utilizar a abordagem gráfica por ser mais fácil, 67% (16) por ser mais útil e 50% (11) utilizariam em uma próxima aplicação. No entanto, 36% (8) preferem utilizar a abordagem textual por ser mais fácil, 29% (6) por ser mais útil e 21% (5) para especificar casos de uso em uma próxima aplicação. Além disso, 4% (1) e 29% (7) preferem utilizar as duas abordagens, em conjunto.

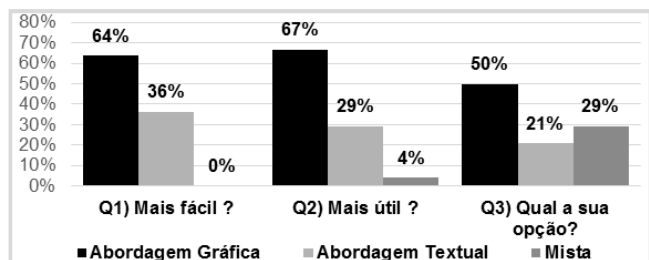


Figura 6: Percepção dos Participantes sobre a Preferência das abordagens para especificar casos de uso

## 4.2 Resultados Qualitativos

Para analisar os dados qualitativos sobre a percepção dos participantes (comentários adicionais) contidos nos questionários, utilizou-se alguns procedimentos de codificação [25], com o auxílio da ferramenta Atlas.ti<sup>2</sup>. O objetivo da análise qualitativa foi identificar as dificuldades encontradas pelos participantes no uso da abordagem gráfica. Além disso, analisaram-se os comentários sobre a preferência de uso das abordagens (textual e gráfica) para especificação. Os dados qualitativos extraídos do questionário pós-

<sup>2</sup> Atlas.ti ± The Knowledge Workbench, Scientific Software Development ± <http://www.atlasti.com>

modelagem foram analisados. Inicialmente, criaram-se conceitos (ou códigos) relevantes sobre a percepção dos participantes quanto as abordagens. Após isso, os códigos foram analisados e foram criados relacionamentos entre eles, além de criar as categorias. O processo de codificação foi revisado por um pesquisador mais experiente. As citações e códigos foram discutidos e, ao final, foram gerados 170 códigos associados a 11 categorias. Na subseção 4.2.1 são apresentadas as principais categorias sobre a percepção da abordagem gráfica. Na subseção 4.2.2 são apresentadas as categorias mais importantes quanto à preferência das abordagens.

#### 4.2.1 Facilidade e utilidade da abordagem gráfica

Entre os códigos criados na categoria “**Facilidade em aprender a abordagem gráfica**”, três deles apontam evidências de facilidade:

“O UCMModel utiliza uma abordagem visual que ajuda a enxergar melhor o caso de uso” – S46

“O modelo abrange todos os aspectos de uma boa especificação como declaração de atores, descrição dos eventos, regras etc” – S21

“Achei fácil utilizar UCMModel, devido ao conhecimento prévio de especificação de casos de uso” – S15

Os três participantes que indicaram essas facilidades possuem baixo conhecimento em Diagramas UML e em Especificação de casos de uso, demonstrando que tiveram facilidade em aprender a abordagem gráfica.

Na categoria “**Facilidade em construir com a abordagem gráfica**”, algumas facilidades foram listadas pelos participantes:

“É fácil, pois a intimidade com os elementos do diagrama ajuda a visualizar melhor as ações” – S13

“É bastante intuitivo em questão do fluxo das atividades” – S7

Os códigos relacionados às categorias sobre facilidade em aprender e construir com a abordagem gráfica mostram indícios que os participantes não tiveram muitas dificuldades. Porém, foram identificadas algumas dificuldades em aprender e construir com a abordagem gráfica:

“UCMModel possui elementos que causam confusão” – S5

“... o UCMModel deixa o caso de uso poluído com muita informação e texto no diagrama.” – S10

Na categoria “**Utilidade para compreender a interação do caso de uso**”, dois participantes afirmaram que o uso da abordagem gráfica não melhorou a compreensão:

“Não mudou o entendimento, apenas melhorou a visualização.” – S7

“Acredito que essa propriedade é bem melhor atendida na especificação textual (já que é mais metódico). Com UCMModel eu não percebi um aumento de nada na verdade” – S18

Na segunda categoria “**Utilidade para especificar casos de uso**”, a maioria dos participantes concordou que a abordagem é útil. Algumas evidências sobre essa utilidade são:

“Simplificou o modo de visualizar o problema” – S9

“Muito útil, fica bem melhor para manipular e ler” – S8

Também foram identificados pontos negativos do uso da abordagem gráfica na especificação de casos de uso:

“Quando faço diagrama tendo a “resumir” os textos para caberem nas etiquetas, especificando de forma textual acredito que faço uma especificação mais rica em detalhes.” – S1

“É útil, porém menos intuitivo que a especificação comum.” – S7

#### 4.2.2 Preferência das abordagens (textual e gráfica) pela facilidade e utilidade

Na análise sobre a preferência das abordagens para especificar casos de uso pela **facilidade**, encontrou-se evidências relacionadas às duas abordagens. Argumentos relacionados às preferências pela abordagem gráfica foram:

“Devido mostrar as descrições no diagrama e ter uma melhor compreensão do caso de uso.” – S2

“UCMModel por ser mais sucinto e também por poder ser melhor detalhado com o uso de tag e notas.” – S21

As preferências em usar a forma textual por ser “mais fácil” foram:

“Basicamente é feita por tópicos, facilitando tanto a escrita como a visualização” – S4

“Possui menos regras ligadas à sua sintaxe” – S7

Os participantes S2 e S21 preferem utilizar a abordagem gráfica por compreender melhor o caso de uso no diagrama e por possuir uma descrição mais sucinta. Já os participantes S4 e S7 preferem a forma textual por possuir menos regras na sua construção e ser mais fácil a escrita e a visualização do caso de uso.

Na análise sobre a preferência das abordagens pela **utilidade**, encontraram-se evidências relacionadas a escolha da abordagem gráfica:

“É mais visual que a especificação textual, o que acredito que facilita e assim torna mais útil.” – S15

“Porque compreende-se todos os fluxos de forma mais clara sem a necessidade de ficar indo e voltando entre as páginas.” – S20

Entretanto, a abordagem textual também foi escolhida pela utilidade em especificar casos de uso, pelas seguintes razões:

“Apresenta as informações importantes e é mais rápida de fazer.” – S4

“É possível especificar o caso de uso de uso em mais detalhes, sem prejudicar o entendimento” – S17

Além disso, foi identificado que as duas abordagens podem ser úteis para diferentes situações, como:

“A textual é mais completa e específica, porém o UCMModel permite uma visualização melhor das interações.” – S6

“As duas são bem úteis. Mas dependendo de cada projeto onde é aplicada a UCMModel é mais prática.” – S22

Para a preferência de utilizar uma das abordagens em uma aplicação, os participantes utilizariam a abordagem gráfica por ser mais simples e visual. Conforme as seguintes declarações:

“Achei mais simples e tão útil quanto a especificação textual.” – S15

“Por ser mais prática e pela “visualização” do caso de uso ser mais fácil devido a utilização dos diagramas.” – S1

No entanto, a abordagem textual também foi escolhida para usar em outra aplicação, por ser mais detalhada e organizada, como mostram as evidências abaixo:

“Para facilitar a pessoa que vai fazer a modelagem o melhor uso para aplicação. Tendo em vista o problema bem detalhado.” – S14

“Mais organização, melhor entendimento, mais detalhes.” – S18

E também ocorreram evidências de utilizar as duas abordagens (textual e gráfica) em conjunto. Elas foram consideradas úteis para



especificar casos de uso e uma pode complementar a outra, como mostram as evidências abaixo:

*“Eu especificaria os casos de uso utilizando uma abordagem mista. Em que ocasiões? Quando estivesse difícil “visualizar” o textual. Creio que os dois podem se completar, porém tomaria muito tempo.” – S6*

*“Eu especificaria os casos de uso utilizando uma abordagem mista. Em que ocasiões? Escrever: Textual e Visualizar: UCMoel.” – S4*

Os resultados qualitativos mostram que a abordagem gráfica baseada no UCMoel [19] pode auxiliar a melhorar a forma como as informações dos casos de uso são apresentadas. Além disso, pode contribuir para a leitura e compreensão geral do caso de uso facilitando sua realização, uma vez que essa abordagem utiliza elementos conhecidos pelos participantes. A abordagem gráfica contribuiu na identificação e visualização das principais funcionalidades e regras que serão implementadas a partir do caso de uso. No entanto, a abordagem gráfica apresentou algumas dificuldades relacionadas a legibilidade do diagrama, ou seja, a especificação gerada pode ficar visualmente grande e de difícil entendimento para os profissionais de software.

Em relação à abordagem textual utilizada (baseada em [9]), os resultados mostram que essa abordagem é útil para detalhar e organizar os cenários, tornando-a mais simples para especificar o caso de uso. Os resultados deste estudo coincidem com os resultados encontrados por Bolloju *et al.* [8], que também afirmam que o formato textual é mais adequado para mostrar o detalhamento dos passos e para demonstrar a interação dos atores no caso de uso. Contudo, foi observado no estudo que a leitura do caso de uso não é facilmente entendida pelos participantes, pois na maioria das vezes, a descrição não é sucinta e simples.

## 5. CONCLUSÕES

Este artigo apresentou um estudo experimental, que comparou duas abordagens utilizadas para especificar casos de uso (abordagem textual e gráfica). Essas abordagens representam formas diferentes de especificar casos de uso. Para isso, verificou-se a correteude de cada especificação e o tempo utilizado por cada participante e as percepções dos participantes em relação a facilidade (de aprendizado e uso) e a utilidade das abordagens.

A análise dos resultados quantitativos mostrou que não há diferença significativa entre as abordagens em relação ao tempo necessário para especificar os casos de uso e a correteude obtida nas especificações. No entanto, a avaliação das especificações geradas mostraram que foram encontrados diferentes tipos de defeitos para as duas abordagens. Os principais defeitos encontrados nas especificações geradas pela abordagem gráfica foram de falha de integridade, incorreteude e falha de legibilidade. Isso pode ter sido ocasionado devido ao fato dos participantes: (a) apresentarem dificuldades em modelar no diagrama os fluxos e regras de negócio do caso de uso; e (b) utilizarem uma abordagem que possui alguns elementos diferentes do que estão familiarizados. No que diz respeito a abordagem textual, os defeitos mais encontrados foram de incorreteude e falha de integridade. Isso pode ser justificado pelo fato dos participantes não utilizarem corretamente os elementos dessa abordagem, o que resultou em descrições incompletas, com omissão de passos/ fluxos/ regras, referências (regras/fluxos) incorretas ou a falta de referências nos passos.

Com relação a percepção dos participantes, notou-se que ter conhecimento em diagramas de atividades da UML foi importante para facilitar o aprendizado e construção da especificação do caso

de uso utilizando a abordagem gráfica. A principal característica quanto a facilidade em aprender a abordagem gráfica foi motivada por ser “visual”, possibilitando aos participantes uma melhor compreensão e entendimento dos casos de uso. A facilidade em construir especificações, com essa abordagem, também está relacionada com esta característica (a visualização). Além disso, a abordagem parece ser intuitiva para a construção, uma vez que os participantes já estão familiarizados com a modelagem em diagramas.

Houve pouca concordância sobre a utilidade da abordagem gráfica em relação a melhoria do entendimento da interação do caso de uso. Segundo os participantes, a abordagem textual e o diagrama de atividades deixam claro esse entendimento. No entanto, a maioria dos participantes (67%), concordou que a abordagem gráfica é útil para especificar casos de uso. As principais características quanto a esta utilidade foram: mostrar a interação e regras do caso de uso de forma visual, simplificar o problema e dar uma visão geral do caso de uso, melhorar a leitura e manipulação do caso de uso, facilitar a compreensão e entendimento do caso de uso.

Os motivos dos participantes que preferiram utilizar a abordagem gráfica em uma aplicação foram: por ser prática e mais fácil para visualizar o caso de uso (em forma de diagrama). E a preferência de utilizar a abordagem textual foi citada por quem prefere ter uma especificação mais detalhada e organizada, sendo melhor para entendimento do caso de uso.

Com isso, percebeu-se que cada abordagem pode ser útil para atender a diferentes situações ou necessidades de casos de uso e projetos de desenvolvimento de software. Por exemplo, a abordagem textual foi considerada mais detalhada para especificar o caso de uso e a abordagem gráfica, por ser visual, foi considerada melhor para a compreensão do caso de uso.

Neste estudo, existiram algumas ameaças que podem afetar a validade dos resultados e que foram mitigadas quando possível. As principais ameaças à validade foram: (1) efeitos de treinamento: os participantes receberam treinamento equivalente nas duas abordagens, incluindo atividades teóricas e exercícios práticos; (2) uso de cenários: foi minimizada utilizando cenários escritos em linguagem natural, onde os requisitos e *mockups* (protótipos) deste cenário estavam explícitos, de forma similar aos exercícios realizados durante o treinamento; (3) os participantes do estudo foram estudantes de graduação; no entanto, esta população era de interesse do estudo, uma vez que se desejava caracterizar a facilidade de aprendizado; (4) o estudo foi realizado em ambiente acadêmico: os artefatos utilizados (diagrama de casos de uso, requisitos do sistema e *mockups*) são de uma aplicação real; (5) tamanho e a homogeneidade da amostra, o que realmente limita a conclusão dos resultados, sendo estes considerados indícios e não conclusivos; (6) a escolha da abordagem gráfica (UCMoel); inicialmente, as abordagens gráficas que adotam diagramas de atividades [12, 30] foram avaliadas, sendo a baseada no UCMoel selecionada por oferecer uma estrutura sintática e semântica do UC mais adequada. Somente após sua seleção, os autores da UCMoel foram contatados para colaboração; e (7) a inclusão de novos itens de verificação; durante a avaliação dos casos de uso, foi verificado que os itens do Checklist de Avaliação [4] não auxiliavam na identificação de outros defeitos encontrados nas especificações geradas pelos participantes, tais como identificar fluxos alternativos, de exceção ou as regras de negócio que estavam descritos de forma incompleta. Então, decidiu-se complementar o Checklist com novos itens (esses novos itens, estão sinalizados em negrito na Tabela 3). No entanto, antes de incluir os novos itens no checklist, estes foram discutidos e validados com mais dois pesquisadores especialistas em UC.

Como trabalhos futuros, pretende-se executar novos estudos experimentais, (a) comparando a abordagem gráfica à abordagem textual de “user stories”, uma vez que costumam ser mais sucintas; (b) identificar, de forma mais detalhada, novas dificuldades encontradas por engenheiros de software em ambiente industrial durante a especificação de casos de uso; e (c) avaliar as diferenças dos resultados para praticantes seniores adeptos ou não da UML.

## 6. AGRADECIMENTOS

Os autores agradecem o apoio financeiro da CAPES, através do processo: 175956/2013, e da FAPEAM através dos processos: 062.00600/2014; 062.00578/2014. Agradecemos também aos participantes do estudo e aos pesquisadores do USES-UFAM pelas contribuições na execução deste estudo.

## 7. REFERÊNCIAS

- [1] Achour C.B., Rolland C., Souveyet C., Maiden N.A. 1999. Guiding use case authoring: results of an empirical study. In 4<sup>th</sup> IEEE Intl Symp. on Requirements Engineering, pp. 36–43.
- [2] Almendros-Jiménez, J.M., Iribarne, L. 2004. Describing Use Cases with Activity Charts. In *Metainformatics*, Springer Berlin Heidelberg, pp. 141–159.
- [3] Anda, B., Dreiem, H., Sjøberg, D., Jørgensen, M. 2001. Estimating software development effort based on use cases – experiences from industry. In *UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools*. Springer, Berlin Heidelberg, v. 2185, pp. 487–502.
- [4] Anda, B., Hansen, K., Sand, G. 2009. 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.
- [5] Anda, B.; Jørgensen, M. 2000. Understanding use case models. In *Beg, Borrow or Steal: Using Multidisciplinary Approaches to Software Engineering Research*, (ICSE) Workshop, pp. 94-102.
- [6] Anda, B.; Sjøberg, D. 2002. Towards an inspection technique for use case models. In 14<sup>th</sup> Intl. Conf. on Software Engineering and Knowledge engineering (SEKE). pp. 127–134.
- [7] Basili, V., Rombach, H. 1988. The TAME Project: Towards Improvement Oriented Software Environments. In *IEEE Transactions on Software Engineering*, v. 14, pp. 758 – 773.
- [8] Bolloju, N., Sun, S. X. Y. 2012. Benefits of supplementing use case narratives with activity diagrams—An exploratory study. In: *JSS* v. 85(9), pp. 2182-2191.
- [9] Cockburn, A. 2005. *Escrevendo casos de uso eficazes*. Trad. Roberto Vedoato. Porto Alegre: Bookman, 2005.
- [10] Cox, K.; Jeffery, R.; Aurum, A. 2004. A Use Case Description Inspection Experiment. In University of New South Wales, School of Computer Science and Engineering, Technical report.
- [11] El-Attar, M., Miller, J. 2009. A subject-based empirical evaluation of SSUCDs performance in reducing inconsistencies in use case models. *Empirical Software Engineering*, v. 14, n. 5, pp. 477-512.
- [12] Gutiérrez, J. J., Nebut, C., Escalona, M. J., Mejías, M., Ramos, I. M. 2008. Visualization of Use Cases through Automatically Generated Activity Diagrams. In *Model Driven Engineering Languages and Systems*. Springer Berlin Heidelberg, v. 5301, pp. 83-96.
- [13] Jacobson, I. 1987. Object-oriented development in an industrial environment. In *Conference on Object-Oriented Programming, Systems, Languages & Applications*, pp. 183–191.
- [14] Jacobson, I.; Christerson, M.; Jonsson, P.; Overgaard, G. 1992. *Object-Oriented Software Engineering: A Use-Case Driven Approach*, Addison-Wesley, Reading, MA, .
- [15] Kulak, D.; Guiney, E. 2012. *Use Cases: Requirements in Context*, Addison-Wesley.
- [16] Laitenberger, O., Dreyer, D.M. 1998. Evaluating the usefulness and the ease of use of a web-based inspection data collection tool. In *Intl Software Metrics Symposium*, 122-132.
- [17] Lanubile, F., Mallardo, T., Calefato, F., 2003. Tool support for Geographically Dispersed Inspection Teams. In *Software Process Improvement and Practice*, v. 8, pp. 217–231.
- [18] Larman, C. 2005. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3/e, Pearson Education India, 1-616.
- [19] Massollar, J. L., De Mello, R. M., Travassos, G. H. 2012. Structuring and Verifying Requirements Specifications through Activity Diagrams to Support the Semi-automated Generation of Functional Test Procedures. In *Intl Conf. on the Quality of Information and Communications Technology*, 2012. p. 239-244.
- [20] Mello, R. M, Massollar, J., Travassos, G. H. 2011. Técnica de inspeção baseada em *checklist* para identificação de defeitos em diagramas de atividades. In *Simpósio Brasileiro de Qualidade de Software (SBQS)*, pp. 168-177.
- [21] Mohagheghi, P., Anda, B., Conradi, R. 2005. Effort estimation of use cases for incremental large-scale software development. In 27<sup>th</sup> Intl. Conf. on Software Engineering, pp. 303–311.
- [22] Nascimento, E. S., Silva, W. A. F., Conte, T., Steinmacher, I., Massollar, J., Travassos, G. H., “Relatório Técnico: Estudo Experimental sobre Diferentes Abordagens para Especificação de Casos de Uso”, Relatório de número 004, 2016. Disponível em: <https://uses.induspam.com/relatorios-tecnicos/>.
- [23] OMG Unified Modelling Language Superstructure – version 2.3. <http://www.omg.org/spec/UML/2.3/>, 2010.
- [24] Phalp, K.T., Vincent, J.; Cox, K. 2007. Assessing the quality of use case descriptions. In *Software Quality Journal*, v. 15, n. 1, pp. 69–97.
- [25] Strauss, A., Corbin, J. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 2 ed. SAGE Publications, London
- [26] Tiwari S., Gupta A. 2015. A systematic literature review of use case specifications research. In *Information and Software Technology*, vol. 67, pp. 128–158.
- [27] Tiwari, S., Gupta, A. 2013. A controlled experiment to assess the effectiveness of eight use case templates. In *Asia-Pacific Software Engineering Conference (APSEC)*, pp. 207–214.
- [28] Wiegers, K.E. 1999. Writing quality requirements. In *Software Development*, v. 7, n. 5, p. 44-48.
- [29] Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wessl, A. 2000. *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers.
- [30] Yue, T., Briand, L., Labiche, Y. 2010. An automated approach to transform use cases into activity diagrams, in: T. Khne, B. Selic, M.-P. Gervais, F. Terrier (Eds.), *Modelling Foundations and Applications*, LNCS, vol. 6138, Springer, Berlin Heidelberg, 2010, pp. 337–353.
- [31] Zhang, X., Auriol, G., Baron, C. 2010. Requirements establishment for complex product development using value-focused thinking. In: *Intl. Conf. on Industrial Engineering and Engineering Management (IEEM)*, pp. 2531-2535.