

# Inspeção de Qualidade em Descrições de Casos de Uso: Uma Proposta de Modelo e Artefatos

José Eduardo Zindel Deboni<sup>1</sup>, Rosângela Gregolin<sup>1</sup>

<sup>1</sup>Instituto de Pesquisas Tecnológicas (IPT)  
Av. Prof. Almeida Prado 532 – 05508-901 – São Paulo – SP – Brazil  
deboni@voxxel.com.br , rgregolin@yahoo.com.br

**Abstract.** *Use case models are used as a method for capturing and specifying functional requirements. However, its use do not guarantee a good quality of specification , because the use of natural language gives way to errors, such as omissions, ambiguities, misunderstandings, and imprecision. With the goal to increase quality in the use case models, this paper proposes a checklist for orientation in the elaboration and inspection of use case descriptions. Techniques for measuring the use case quality are also suggested. An application of the proposal is made.*

**Resumo.** *Modelos de caso de uso são utilizados como um método de captura e especificação de requisitos funcionais. Todavia, sua utilização não garante uma especificação de boa qualidade, pois sua elaboração, feita em sua maior parte em linguagem natural, dá margem à presença de diversos defeitos, tais como omissões, ambigüidades, pouca compreensibilidade e imprecisão. Com o objetivo de aumentar a qualidade dos casos de uso, este trabalho propõe um checklist para a orientação na elaboração e inspeção de descrições de caso de uso. Técnicas para medição da qualidade do caso de uso também são sugeridas. Uma aplicação da proposta exemplifica a inspeção.*

## 1. Introdução

O caso de uso é um meio para capturar e especificar requisitos de um sistema, definindo seu comportamento de acordo com as necessidades dos atores: usuários e outras entidades que interagem com o sistema. O caso de uso deve produzir um resultado de valor para o ator, representando as grandes funcionalidades do sistema. O modelo de caso de uso é composto de diagramas e descrições de casos de uso, e deve descrever com detalhes as funcionalidades do sistema. Todos os requisitos funcionais descritos são utilizados como entrada para diversas atividades no desenvolvimento de software, tais como o projeto, codificação e testes. Portanto, a qualidade do modelo de caso de uso tem um impacto significativo no processo de desenvolvimento e na qualidade do restante do produto [Jacobson 1992]

A descrição de caso de uso é elaborada, em geral, em linguagem natural, o que a torna simples de utilizar e de entender. No entanto, a aparente simplicidade pode representar um perigo para a precisão e clareza, na medida em que pode levar equipes de desenvolvimento a cometer erros como ambigüidades, redundâncias e omissões na

descrição, comprometendo os eventuais benefícios do seu uso [Fogarty 2004].

Embora existam diversas publicações de como construir bons modelos de caso de uso, sua qualidade pode aumentar se, além de seguidas as diretrizes corretas para elaborá-los, também for aplicada uma inspeção de qualidade, como sugerem Anda e Sjoberg (2002). A inspeção é uma atividade de teste de software baseada no exame visual de produtos do desenvolvimento para detectar defeitos, violações de padrões de desenvolvimento, entre outros problemas [IEEE 1990]. Shull et al (2000) mostram ainda que a inspeção é uma maneira sistemática de ensinar a equipe de desenvolvimento como construir um software melhor, atuando também de forma preventiva à ocorrência de defeitos.

O objetivo deste trabalho é desenvolver um instrumento para apoiar a inspeção de qualidade das descrições dos casos de uso, e que possa, ao mesmo tempo, ser utilizado como subsídio para a orientação de equipes de desenvolvimento na elaboração de descrições de caso de uso com qualidade. Este instrumento, fundamentado principalmente na ferramenta de um *checklist*, facilita o encontro de defeitos nas descrições dos casos de uso, antes que elas passem para a próxima etapa do desenvolvimento do software. Visto que o caso de uso é utilizado nas etapas seguintes do decorrer do processo de desenvolvimento de software, aumentar sua qualidade significa aumentar a qualidade do software como um todo, contribuindo para a detecção de falhas antecipada onde se estima que o custo de correção é menor. É importante considerar, no entanto, que a proposta visa detectar se os requisitos foram especificados da forma certa e não se os requisitos certos foram especificados, o que significa que mesmo que um caso de uso obtenha o aceite da inspeção proposta, isso não significa que ele traduza os requisitos corretos para o usuário. A validação de requisitos está fora do escopo deste trabalho.

## **2. Considerações sobre inspeção e o *checklist***

Para Gilb e Graham (1993), a inspeção consiste em um valioso investimento no esforço de desenvolvimento de um software, já que há um aumento na produtividade nas fases posteriores à inspeção que se utilizarão do produto inspecionado: redução de testes de execução, redução na manutenção, diminuição do prazo de entrega do software e redução do total de defeitos. Consideram que uma inspeção é capaz de descobrir de 60% a 80% de defeitos.

Shull et al. (2000) propõem que o processo de inspeção seja constituído das etapas de: planejamento, detecção de defeitos, registro dos defeitos encontrados e correção dos defeitos, e afirma que inspeções efetuadas por um único inspetor podem ser tão úteis quanto uma reunião de vários inspetores.

O *checklist* é uma ferramenta de inspeção de qualidade que apresenta as diretrizes para a leitura de um determinado artefato com o objetivo de encontrar defeitos. Consiste em uma lista de questões que o inspetor deve identificar se estão presentes ou não no artefato em teste [Travassos e Mafra 2005].

Para Laitenberger et al. (2002), o *checklist* é a técnica de leitura mais empregada para a detecção de defeitos no contexto de inspeção de software. Anda e Sjoberg (2000) enfatizam que essa popularidade se dá pela facilidade de sua aplicação.

Gilb e Graham (1993) sugerem que o *checklist* deva ser derivado de regras de elaboração do produto a ser inspecionado, incluindo, a cada questão, uma referência à regra da qual a questão é derivada. Essas regras devem ser informadas previamente ao autor do produto a ser inspecionado. É conveniente que o conjunto de questões de um *checklist* não deva exceder uma única página, mantendo-se no máximo em 25 questões. Por esta razão, as questões devem se concentrar nos maiores defeitos. As questões devem ser diretas e objetivas, e devem esclarecer as regras de elaboração do produto inspecionado. Também devem ser formatadas de forma que a resposta negativa represente a identificação de um problema. Além disso, cada questão deve conter sugestões de níveis de severidade, dependendo do impacto que o defeito encontrado pode causar.

### 3. Modelo e Artefatos para inspeção da qualidade

Para a elaboração do *checklist* de descrições de casos de uso foi, inicialmente, criado um modelo de qualidade com os atributos de qualidade e as regras de elaboração de uma descrição de caso de uso. A partir desse modelo, deriva-se o *checklist* que verifica se as descrições de caso de uso inspecionadas atendem ao modelo de qualidade.

#### 3.1. Elaboração de um modelo de qualidade

Para a criação do modelo de qualidade, foram analisados e comparados os atributos de qualidade e suas características dos modelos de qualidade do IEEE (1998), de Davis et al. (1993) e de Fabbrini et al. (2001), resumidos na Tabela 1.

A Tabela 1 compara os atributos de qualidade. Nela busca-se sintetizar em uma definição única os diversos atributos de qualidade empregados por cada modelo de qualidade, para dela se extrair os atributos de qualidade a serem empregados para elaborar o modelo de qualidade da descrição de casos de uso e dele o *checklist* para apoiar a inspeção.

**Tabela 1. Comparativo de modelos de qualidade de especificação**

Atributo de qualidade (IEEE, 1998)	Atributo de qualidade (DAVIS et al., 1993)	Atributo de qualidade (FABBRINI et al., 2001)	Definição
Não ambigüidade	Não ambigüidade	Compreensibilidade	Termos que dão margem a múltiplas interpretações devem constar em glossário, com clara definição.
	Completeza		Contemplação de todos os requisitos.
Completeza			Definição de resposta do sistema para as entradas de dados em todas as situações, considerando valores válidos e inválidos.
Completeza		Consistência	As referências devem estar numeradas e os documentos anexos devem estar referenciados.
Completeza		Compreensibilidade	Deve haver explicações para todas as figuras, tabelas, diagramas,

			acrônimos e unidades de medida.
Correção	Correção		A especificação deve representar exatamente o que o sistema deve fazer.
	Compreensibilidade		Os requisitos devem ser entendidos pelos envolvidos com o mínimo de explanação.
Verificabilidade	Verificabilidade, Precisão	Testabilidade	Todo requisito deve ser preciso, para verificação futura se foi atendido.
		Testabilidade	A especificação deve evitar presença de opiniões pessoais, palavras que indicam opção, sentenças com verbos que não exprimem ações, menção a funcionalidades ainda não especificadas.
Consistência	Consistência interna e consistência externa		Os requisitos não podem se conflitar com requisitos do mesmo sistema ou de sistemas externos.
	Alcançabilidade, Prototipação		A especificação deve ser acompanhada de protótipo de interface.
	Independência de ferramenta		Deve haver mais de uma ferramenta de interface capaz de viabilizar o requisito
Rastreabilidade para frente	Rastreabilidade		Cada requisito tem que ter um único nome ou n° de referência.
Rastreabilidade para trás			Cada requisito tem que ter sua origem em documentos iniciais do projeto.
Modificabilidade	Modificabilidade, Não redundância		A especificação deve ser organizada de forma a ser facilmente modificável. Deve ser simples, evitando redundâncias.
Modificabilidade	Armazenamento eletrônico		A especificação deve estar eletronicamente armazenada, de forma organizada.
Classificação por importância	Classificação por importância e por estabilidade		Os requisitos devem ser classificados para ajudar na priorização de esforços
	Referência à versão		O requisito deve ter referência a qual versão será implementado.
		Compreensibilidade	As sentenças devem ter quantidade limitada de palavras e evitar mais de um verbo principal.

Após o comparativo, os atributos selecionados para compor o modelo de qualidade das descrições de caso de uso são: completeza, compreensibilidade, precisão, não ambigüidade, consistência, independência de ferramenta e de interface, rastreabilidade, e objetividade. O atributo prototipação sugerido por Davis et al. (1993) recebeu o nome de compreensibilidade, visto ser a compreensibilidade o objetivo da prototipação. Também foi classificado como compreensibilidade o atributo de modificabilidade, que define a característica de que a especificação deve evitar

redundâncias, assim como conter índices para aumentar a compreensibilidade e facilidade de manutenção [IEEE 1998]. Os atributos de classificação por importância e referência à versão de Davis et al. (1993) não foram considerados no contexto restrito da inspeção proposta, por terem uma utilização voltada para o gerenciamento do processo de desenvolvimento e não restrita à qualidade da descrição de caso de uso.

### **3.2 Modelo de Qualidade para Descrição de Caso de Uso**

Com base nos atributos selecionados e derivados da Tabela 1, e adicionados das diretrizes de elaboração de modelo de caso de uso de Spence e Bittner (2003), Jacobson et al (1992), Jacobson (2003), Gottesdiener (2003), OMG (2007) e Ambler (2003), criou-se um modelo de qualidade, descrito a seguir:

#### **Atributo de Qualidade 1: Completeza**

**Regra 1.1.** A descrição de caso de uso deve ser a especificação detalhada de um caso de uso representado no diagrama com o mesmo nome da descrição de caso de uso. Um modelo de caso de uso, portanto, quando chega na etapa de descrição do caso de uso, necessita ter o diagrama.

**Regra 1.2.** A descrição de caso de uso deve conter minimamente o nome do caso de uso, nome dos atores, fluxo básico e alternativo completos.

**Regra 1.3.** A descrição de caso de uso deve especificar o fluxo básico e alternativo da forma mais completa possível, prevendo no fluxo alternativo decisões alternativas de usuários e exceções.

#### **Atributo de Qualidade 2: Compreensibilidade**

**Regra 2.1.** As sentenças devem evidenciar um diálogo exato entre ator e sistema. Isto significa que para que o sistema execute uma ação, é necessária uma ação invocativa do ator, e para que o ator execute uma ação, é necessária uma ação de disponibilidade do sistema, para possibilitar a ação ao ator.

**Regra 2.2.** A escrita deve ser em voz ativa ao invés de passiva. (Exemplo de voz ativa: Atendente seleciona opção de finalizar pedido. Exemplo de voz passiva: Opção de finalização de pedido é selecionada pelo atendente).

**Regra 2.3.** A escrita deve ser em tempo presente ao invés de passado ou futuro.

**Regra 2.4.** É recomendável que a escrita de fluxos longos, que ocupem mais de uma página, utilize subtítulos para comunicar as idéias-chaves.

**Regra 2.5.** A descrição de caso de uso com fluxo de mais de uma página deve conter índice e paginação.

**Regra 2.6.** As tabelas, figuras, acrônimos devem ser seguidas de explicações que facilitem o entendimento.

**Regra 2.7.** A descrição deve tratar o quê o sistema deve fazer e não como fazer, e, portanto, devem evitar presença de especificações de código de sistema.

**Regra 2.8.** A descrição deve definir requisitos de sistema e não de negócio, evitando descrever o que ocorre fora do sistema sem ação direta ao sistema.

**Regra 2.9.** A presença dos artefatos de suporte aumenta a compreensibilidade: glossário, protótipo de interface, especificação de requisitos não funcionais, modelo de domínio, regras de negócio separadas do fluxo básico e alternativo do caso de uso.

**Regra 2.10.** O fluxo alternativo, quando causa desvio da compreensão do fluxo básico e tem mais de uma linha, deve ser descrito separadamente do fluxo básico.

**Regra 2.11.** A descrição do caso de uso deve conter breve descrição, com especificação clara do propósito do caso de uso.

#### **Atributo de Qualidade 3: Precisão**

**Regra 3.1.** Os termos devem ser precisos e quantificáveis, evitando termos impossíveis de quantificação, como “muito”, “pouco”, “adequado”, “claro”, “fácil” “longo”, “curto”, “rápido” etc.

**Regra 3.2.** Os termos que indicam opção: “possivelmente”, “alternativamente”, “no caso de”, “se”, só podem ocorrer se especificarem fluxo alternativo.

#### **Atributo de Qualidade 4: Não ambigüidade**

**Regra 4.1.** Termos que dão margem a mais de uma interpretação devem constar em glossário, de forma clara.

#### **Atributo de Qualidade 5: Consistência**

**Regra 5.1.** Os termos utilizados devem ser iguais quando se referem à mesma coisa, evitando-se, por exemplo, ora utilizar-se o termo “pedido” e ora utilizar-se o termo “requisição”.

**Regra 5.2.** Qualquer referência do caso de uso deve ser corretamente numerada ou nomeada, tanto na origem como no destino para que permita fácil localização.

#### **Atributo de Qualidade 6: Independência de ferramenta e de interface**

**Regra 6.1.** A descrição de caso de uso deve evitar termos que indicam dependência de ferramenta ou interface de usuário, como por exemplo, presença do termo “clicar”, “aba”, “botão” etc.

#### **Atributo de Qualidade 7: Rastreabilidade**

**Regra 7.1.** As ações especificadas na descrição do caso de uso devem ser identificadas preferencialmente através de numeração para permitir a rastreabilidade ao longo do processo.

#### **Atributo de Qualidade 8: Objetividade**

**Regra 8.1.** As sentenças devem ser objetivas e evitar redundâncias, evitando-se sentenças redundantes, longas e excessivas, que poderiam ser reduzidas sem perder o sentido. Também se devem evitar sentenças desnecessárias e fora do escopo do problema.

### **3.2. Checklist para Descrição de Caso de Uso**

Com base no modelo de qualidade para descrição de caso de uso, cria-se um artefato para inspeção da descrição de caso de uso na forma de um *checklist*, elaborando

questões que buscam localizar não-conformidades ao modelo de qualidade. O artefato se apresenta na Tabela 2.

Para cada questão, é sugerida uma classificação do impacto para o defeito encontrado, categorizados em impacto alto (A), médio (M) e baixo (B). A classificação do impacto é uma proposta que pode ser reavaliada pela equipe para cada fase do desenvolvimento, tipo de sistema a ser desenvolvido e objetivos da aplicação da inspeção.

Como critérios para a sugestão de impacto alto, foram considerados defeitos que afetam a compreensibilidade de forma mais significativa como a falta de coerência entre a descrição e o diagrama, a falta de definição clara de termos ambíguos em glossário, a ausência do protótipo de interface, e defeitos que afetam a completeza. Como critérios para o impacto médio, foram considerados defeitos que afetam a qualidade do caso de uso de forma considerável, mas num grau menos importante que os defeitos de impacto alto. Como critérios para o impacto baixo, foram considerados defeitos que, se corrigidos, aumentariam a qualidade do caso de uso, mas não impedem a compreensibilidade geral.

**Tabela 2. Checklist de descrição de caso de uso**

Questão	Regra	Im- pacto
1. A descrição de caso de uso é a de um caso de uso representado no diagrama?	1.1	A
2. A descrição de caso de uso contém nome do caso de uso, nome do ator, fluxo básico e alternativo? (Se não houver um fluxo alternativo definido, considerar se ele está especificado dentro do fluxo básico)	1.2	A
3. A descrição de caso de uso com mais de uma página contém índice e paginação?	2.5	B
4. Se houver tabelas ou figuras, elas têm explicação adicional de forma que fiquem compreensíveis para o leitor?	2.6	B
5. Se houver referências, essas são numeradas ou nomeadas da mesma forma na origem (descrição de caso de uso) e no destino (a própria referência)?	5.2	B
6. As frases representam um diálogo entre ator e sistema, evidenciando a ação do ator e a resposta do sistema?	2.1	M
7. As frases se utilizam de subtítulos para comunicar as idéias chaves dos fluxos de forma mais clara?	2.4	B
8. As frases são construídas em voz ativa? (ex.: “Sistema valida a quantia informada” em vez de “A quantia informada deve ser validada pelo sistema”).	2.2	B
9. As frases utilizam o tempo presente?	2.3	B
10. São evitados termos sem quantificação precisa, como “muito”, “pouco”, “adequado”, “claro”, “fácil” “longo”, “curto”, “rápido” “etc”?	3.1	M
11. São evitados termos que indicam opção, como “possivelmente”, “alternativamente”, “no caso”, “se”, etc, sem especificar um fluxo alternativo?	3.2	M
12. Os termos passíveis de mais de uma interpretação constam em glossário, com clara definição?	4.1	A
13. Uma vez utilizado um termo, ele é mantido para referenciar-se ao mesmo elemento?	5.1	M
14. São evitados termos que indicam a prematura especificação de interface, tais como “clicar” “botão” etc?	6.1	B
15. As funcionalidades se restringem ao quê o sistema deve fazer e não em como,	2.7	M

evitando a definição explícita de código na especificação?		
16.A descrição evita requisitos de negócio sem ação direta ao sistema?	2.8	M
17.Há presença de breve descrição ou resumo no início da descrição de caso de uso, que especifique de forma clara o seu propósito?	2.11	B
18.O fluxo básico está aparentemente completo, isto é, há inexistência de evidências claras de incompleta na especificação?	1.3	A
19.O fluxo alternativo está aparentemente completo, isto é, há inexistência de evidências claras de incompleta na especificação?	1.3	A
20.As frases são numeradas para que possibilitem a rastreabilidade?	7.1	M
21.As frases procuram ser objetivas, evitando redundâncias ou presença de informações evidentemente desnecessárias?	8.1	M
22.O caso de uso é acompanhado de protótipo de interface a fim de aumentar a sua compreensibilidade?	2.9	A
23.O caso de uso é acompanhado de especificação de requisitos não funcionais separadas do fluxo de eventos do caso de uso ou em documento de especificação suplementar?	2.9	M
24.O caso de uso é acompanhado de modelo de domínio, mostrando os relacionamentos entre os principais conceitos do sistema, a fim de aumentar a sua compreensibilidade?	2.9	M
25.Se houver regras de negócios associadas, estas estão separadas dos fluxos de evento do caso de uso ou em documento de especificação de regras de negócios?	2.9	B

### 3.3. Artefato de Registro de Inspeção

Para registrar os defeitos encontrados quando da aplicação do *checklist*, e para que sirvam de base para uma medida da qualidade, propõe-se o artefato de registro de inspeção apresentado na Tabela 3.

Visto que o termo “defeito” pode causar desconforto e resistência por parte dos autores de produtos inspecionados, sugere-se no registro a substituição do termo por “não-conformidade”, baseado na norma ISO 19011 [ABNT 2002], que fornece orientação sobre a gestão de programas de auditoria e propõe que registros que não atendam aos critérios da auditoria sejam classificados como “não-conformidade”.

**Tabela 3. Registro de inspeção em descrição de caso de uso**

Nome do caso de uso: Data da inspeção: Responsável pela inspeção: Hora inicial: Hora final: Tempo despendido: Quantidade de páginas inspecionadas:						
Nº. da questão	Impacto (A/M/B)	Registros de não-conformidades				Observações (Localização, obs.)
		N/A	Sim	Não	Qtde	
Outras Observações:						

Após a identificação da descrição de caso de uso a ser inspecionada, registra-se para cada questão do *checklist*, na forma de linhas: o impacto daquela questão obtido na



Tabela 1, as não-conformidades encontradas em cada questão, onde N/A (não se aplica) indica que a questão não se aplica ao produto inspecionado, S (sim) indica que o produto inspecionado atende à questão e N (não) indica que o produto não atende à questão e, portanto, caracteriza um defeito. Alguns tipos de defeitos podem ocorrer mais de uma vez na descrição, e por isso a quantidade encontrada deve ser registrada. Uma coluna permite que o inspetor faça observações em cada item inspecionado.

#### 4. Medição da Inspeção

A busca de uma técnica para a medição da inspeção é importante para se obter um resultado de aceite para o produto inspecionado e prover uma noção mais clara e confiável da qualidade de um caso de uso, que de outra forma seria baseada apenas na intuição. Mora e Denger (2003) efetuaram amplo levantamento de publicações de trabalhos no campo de medições de qualidade de especificação de requisitos e constataram de que não existe o consenso de uma medição padronizada nesse campo.

Propõe-se uma técnica de medição da inspeção da descrição de caso de uso com base nos resultados da aplicação do *checklist*. Esta medição pode ajudar na criação de critérios objetivos de aprovação ou reprovação dos casos de uso inspecionados. A Tabela 4 descreve um artefato de registro dos defeitos encontrados e permite obter um resultado quantitativo de cada descrição de caso de uso do modelo inspecionado.

**Tabela 4. Registro para a medição da inspeção**

Critério de aceite (em %):							
Impacto	Peso	Qtde de questões aplicadas	Qtde de não-conformidades encontradas	Peso total	Peso das não-conformidades	Resultado quantitativo	Descrição do resultado quantitativo: (Aprovada ou Reprovada)
Alto						MQ =	
Médio							
Baixo							
Total				TotalPT =	TotalNC =		

O critério de aceite da descrição refere-se a um percentual máximo de defeitos que o produto inspecionado poderá ter e que será obtido pela somatória da coluna de resultado quantitativo. Se o resultado quantitativo ultrapassar o critério de aceite, não obterá um aceite da inspeção, sendo assim reprovado. Pode-se, por exemplo, usar um critério de maior tolerância nas fases preliminares de um projeto e aumentar o rigor, diminuindo a tolerância em fases mais avançadas do projeto.

Para cada nível de impacto devem ser registrados: o peso atribuído para aquele tipo de defeito, o número de questões que foram classificadas com aquele nível de impacto e o número de não conformidades encontradas naquele nível de impacto. A coluna Peso total é o resultado da multiplicação do peso com o número de questões aplicadas. A coluna de Peso das não-conformidades é o resultado da multiplicação da coluna de Peso pela coluna de Qtde de não-conformidades encontradas.

O resultado quantitativo da medida da qualidade (MQ) refere-se a um percentual do peso das não-conformidades encontradas em relação ao peso das questões aplicadas,

seguindo a seguinte fórmula:

$$MQ = \text{TotalNC} * 100 / \text{TotalPT}$$

Onde:

TotalNC = Somatória do Total do peso das não-conformidades e

TotalPT = Somatória do Total do peso das questões aplicadas.

A descrição do resultado quantitativo compara MQ com o critério de aceite pré-estabelecido. Isto é, se o resultado quantitativo for menor ou igual ao critério de aceite, a descrição do caso de uso é aprovada, caso contrário é reprovada.

## 8. Avaliação experimental do *checklist* de inspeção

Para avaliar a inspeção proposta, realizou-se uma aplicação com três estudantes de pós-graduação em engenharia de software, que foram apresentados ao *checklist* a fim de atuarem como inspetores. Duas descrições de casos de uso extraídos de dois modelos de caso de uso de projetos reais de sucesso foram avaliadas pelos estudantes. O objetivo da aplicação foi o de avaliar a facilidade de entendimento e uso do *checklist*, assim como a sua precisão pela coerência das respostas dos inspetores.

A opinião geral do grupo foi de que a aplicação do *checklist* foi fácil de executar e que o entendimento das questões ocorre sem dificuldade. A forma de elaboração do *checklist*, em uma única página, a objetividade das questões e as regras detalhadas sobre cada item a ser inspecionado, facilitam a aplicação do *checklist*. O tempo médio da aplicação foi de 20 minutos para cada descrição, composta de cerca de sete páginas cada.

Na descrição do primeiro caso de uso, 21 questões foram aplicadas e foram encontradas, na média, 8 não-conformidades, levando a uma MQ que varia de 26% a 30% como mostra a Tabela 5. Como o critério adotado foi de 40% a descrição do primeiro caso de uso foi aprovada por todos.

**Tabela 5. Resumo das medições de qualidade da aplicação em caso de uso nº1**

Inspetor	Número de não-conformidades encontradas	Peso TotalNC	MQ	Resultado do diagrama de caso de uso
1	7	11	26,19%	Aprovada
2	9	13	30,95%	Aprovada
3	7	12	28,57%	Aprovada

A segunda descrição do caso de uso permitiu aplicar 22 questões. A Tabela 6 descreve um resumo das medições de qualidade obtidas da aplicação do *checklist* pelos 3 inspetores.

Neste exemplo houve um entendimento pelo 3º inspetor de que a questão 14 não foi atendida em diversos momentos da descrição. Observa-se que apesar do rigor diferenciado do inspetor, o resultado final não se alterou, dando indícios que o modelo pode gerar uma independência do resultado em relação ao inspetor.

Conforme exposto na Tabela 6, a descrição de caso de uso foi aprovada nas três inspeções, pois as inspeções obtiveram percentuais de defeitos inferiores ao critério de aceite, que é de 40%.

Importante ressaltar que todos os impactos sugeridos, assim como os pesos e critério de aceite de 40%, são propostas passíveis de serem adequadas a cada aplicação.

**Tabela 6. Resumo das medições de qualidade da aplicação em caso de uso nº2**

Inspetor	Número de não-conformidades encontradas	Peso TotalINC	MQ	Resultado do diagrama de caso de uso
1	2	5	11,63%	Aprovada
2	2	5	11,63%	Aprovada
3	8	11	25,58%	Aprovada

## 9. Conclusões e perspectivas de trabalhos futuros

Embora o modelo de caso de uso seja amplamente utilizado como um método de especificação de requisitos funcionais, sua utilização não garante uma especificação de boa qualidade, já que as descrições em linguagem natural dão margem à presença de diversos defeitos. Objetivando proporcionar a melhoria da qualidade dos modelos de caso de uso, este trabalho propõe um *checklist* como subsídio para a orientação de equipes de desenvolvimento na elaboração de descrições de caso de uso e para que clientes utilizem-no para a inspeção de qualidade destas descrições.

O *checklist* proposto sugere uma classificação dos defeitos em graus de severidade, o que combinado com as respostas do *checklist* possibilita associar uma medida da qualidade da descrição de caso de uso inspecionada. Em conjunto com um critério pré-definido de aceite, pode-se obter um resultado objetivo da aprovação ou reprovação da descrição de caso de uso. Esta medida não é absoluta e deve ser motivo de ajustes preliminares para cada equipe de desenvolvimento antes de poder ser aplicada amplamente.

Na aplicação da proposta, o *checklist* mostrou-se eficaz no encontro de não-conformidades nos casos de uso, uma vez que os resultados dos inspetores foram bastante semelhantes. Acredita-se que o uso do *checklist* possa tornar a inspeção uma atividade mais independente das habilidades e experiência do inspetor.

Através da técnica de medição proposta, é possível prover uma noção mais clara da qualidade da descrição de caso de uso inspecionada, provendo uma objetividade maior para a decisão se o produto poderá passar para a próxima etapa de desenvolvimento ou se terá que retornar para a correção dos defeitos encontrados. A adoção da proposta em processos de desenvolvimento de software poderá representar, além de uma forma de apontamento de não-conformidades, também uma forma de prevenção de defeitos, pois os especificadores de modelos de caso de uso poderão se basear nos modelos de qualidade propostos para a elaboração da descrição, garantindo sua qualidade.

Uma importante limitação da proposta, que deve ser ressaltada, é a que diz respeito ao objetivo da inspeção. O *checklist* proposto visa verificar que se está construindo uma descrição corretamente e não validar que a descrição reflete as necessidades do usuário. O processo de inspeção é um momento adequado de se realizar a validação da especificação do sistema junto ao usuário, mas a inspeção proposta não colabora para este objetivo.

Como trabalho futuro de pesquisa, sugere-se a automatização de inspeções em modelos de caso de uso, e a possibilidade da criação de uma ferramenta de automatização de inspeção. Sugere-se a ampliação da aplicação do *checklist* em um número maior de projetos para observar sua real eficácia.

## 10. Referências

- ABNT (2002) ISO 19011 – Diretrizes para Auditorias de Sistemas de Gestão da Qualidade e/ou Ambiental. Associação Brasileira de Normas Técnicas.
- Ambler, S. W. (2003) The Elements of UML Style. Cambridge University Press. 1<sup>st</sup> Edition.
- Anda, B. e Sjoberg, D.I.K.(2002) Towards an Inspection Technique for Use Case Models. 14th International Conference on Software Engineering and Knowledge Engineering.
- Bittner, K. (2000) Why Use Cases are Not Functions. The Rational Edge.
- Davis, A et al. (1993) Identifying and Measuring Quality in a Software Requirements Specification. Proceedings of the First International Software Metrics Symposium. Baltimore.
- Fabbrini, F et al. (2001) An Automatic Quality Evaluation for Natural Language Requirements. 7th International Workshop on Requirements Engineering (REFSQ'01).
- Fogarty, M. (2004) A Practitioner's Guide to Writing Use Cases. IEEE Software. Vol 21. n.2, March/April
- Gilb, T. e Graham D. (1993). Software Inspection. Addison-Wesley.
- Gottesdiener, E. (2003) Use Cases: Best Practices. Rational Software. <http://www.eg.bucknell.edu/~cs475/F04-S05/usecases.pdf>
- IEEE (1990) Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990.
- IEEE (1998) Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998.
- Jacobson, I. (2003) Use Cases: Yesterday, Today, and Tomorrow. The Rational Edge.
- Jacobson, I. et al. (1992) Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley. 1<sup>st</sup> Edition Revised.

- Laitenberger, O. et al. (2002) An Industrial Case Study to examine a non-traditional Inspection Implementation for Requirements Specifications. Eighth IEEE International Symposium on Software Metrics.
- Mora, M. e Denger, C. (2003) An Initial Literature Survey on Measurement Approaches for Requirement Specifications. Fraunhofer IESE – Institute Experimentelles Software Engineering. Relatório Técnico nº 096.03/E.  
<http://publica.fraunhofer.de/starweb/servlet.starweb?path=pub.web&search=N-18789>
- OMG (2007) Object Management Group. Unified Modeling Language, version 2.1.1.  
<http://www.omg.org/technology/documents/formal/uml.htm>
- Shull, F, et al. (2000) “How Perspective-Based Reading can Improve Requirements Inspections”. Computer. IEEE. vol.33 , n.7 , pages 73-79.
- Spence, I e Bittner, K. (2003) Use Case Modeling. Addison-Wesley.
- Travassos, G. H. e Mafra, N.M. (2005) Técnicas de Leitura de Software: Uma Revisão Sistemática. XIX Simpósio Brasileiro de Engenharia de Software.

