

TUCCA: Técnica de Leitura para apoiar a Construção de Modelos de Casos de Uso e a Análise de Documentos de Requisitos

Anderson Belgamo e Sandra Fabbri

Departamento de Computação - Universidade Federal de São Carlos (UFSCar)
Caixa Postal 676 - CEP 13565-905 – São Carlos – SSP – Brasil

{belgamo, sfabbri}@dc.ufscar.br

Abstract. *The objective of this paper is to present TUCCA¹ – Technique for Use Case model construction and Construction-based requirements document Analysis. TUCCA is composed of two reading techniques: AGRT – Actor Goal Reading Technique – whose objective is to determine the actors of the system and their goals and UCRT – Use Case Reading Technique – whose objective is to determine the Use Case Model. The steps of these techniques support the construction of the Use Case Model and also embody a review of the Requirements Document. To illustrate the contribution of the technique, the results of two empirical studies carried out in an academic environment, that compare TUCCA and Ad-Hoc for constructing Use Case Models and TUCCA and Checklist for detecting defects in the Requirements Document, are commented.*

Keywords: *reading techniques, Use Case construction, requirements inspection.*

Resumo. *O objetivo deste artigo é apresentar a técnica TUCCA¹ – Technique for Use Case model construction and Construction-based requirements document Analysis. TUCCA é composta de duas técnicas de leitura: AGRT – Actor Goal Reading Technique – cujo propósito é determinar os atores do sistema e seus objetivos e UCRT – Use Case Reading Technique – cujo propósito é determinar o Modelo de Casos de Uso. Os passos dessas técnicas dão suporte à construção de Modelos de Casos de Uso e também incorporam uma revisão do Documento de Requisitos. Para ilustrar a contribuição dessa técnica, são apresentados os resultados de dois estudos experimentais que compararam TUCCA com Ad-Hoc no que se refere à construção de Modelos de Casos de Uso e TUCCA com Checklist no que se refere à detecção de defeitos no Documento de Requisitos.*

Palavras-chaves *técnicas de leitura, construção de Casos de Uso, inspeção de requisitos.*

¹ Previamente referenciada como GUCCRA – Guidelines for Use Case Construction and Requirements Document Analysis

1. Introdução

A UML (Unified Modeling Language), proposta pelo OMG (Object Management Group) [OMG, 2003], é uma linguagem padrão amplamente utilizada no contexto do paradigma de desenvolvimento de software Orientado a Objetos (OO). Ela é composta por uma série de diagramas e modelos que, embora bem definidos sintaticamente, não possuem diretrizes para elaboração, ficando isso muito dependente da subjetividade e experiência do projetista. Tais diagramas e modelos podem ser utilizados em qualquer tipo de processo, pois também não existe um processo específico proposto para a elaboração dos mesmos.

Independentemente desses aspectos, além de ser bastante utilizada no projeto de sistemas OO, algumas técnicas usam a UML como uma ferramenta de suporte e outras têm os diagramas UML como foco de avaliação. Alguns exemplos são: as técnicas de leitura PBR (Perspective Based Reading) [Basili et al., 1996] [Shull et al., 2000], que usam os Modelos de Casos de Uso como suporte para aplicar a perspectiva do usuário na inspeção de Documentos de Requisitos (DR); as técnicas de leitura OORTs (Object-Oriented Reading Techniques) [Travassos et al., 2002] que dão suporte a atividades de validação e verificação em modelos UML de projeto de alto nível e as técnicas de leitura OORTs/ProDeS (Object-Oriented Reading Techniques for ProDeS) [Marucci et al., 2002], que dão suporte a atividades de validação e verificação em modelos UML usados em um processo específico de desenvolvimento OO.

Os Modelos de Casos de Uso (diagrama e especificação) propostos na UML foram baseados na proposta de Jacobson [Jacobson et al., 1992], a qual considera esse item como o centro do processo de desenvolvimento de software. Embora Jacobson tenha sido o precursor dessa idéia, ele não fornece diretrizes para a elaboração das especificações dos casos de uso, mas apenas sugestões que ajudam na identificação dos casos de uso e dos atores. Nos trabalhos de Kulak [Kulak & Guiney, 2000] e Schneider [Schneider & Winters, 2001] encontram-se outras abordagens para elaboração do Modelo de Casos de Uso, mas que também não consideram todos os aspectos relacionados a essa atividade. Kulak e Guiney (2000) sugerem uma forma iterativa de construir esses Modelos, a qual é composta de quatro iterações pré-definidas. No entanto, eles não fornecem diretrizes para identificar e especificar os casos de uso. Schneider e Winters (2001) fornecem algumas questões para ajudar a identificar atores e algumas sugestões para criar casos de uso do tipo CRUD (*create, read, update e delete*). Assim como em outros trabalhos, eles não fornecem diretrizes que ajudem a escrever a especificação dos casos de uso. Já Cockburn (2001), embora forneça algumas sugestões para escrever as especificações e algumas diretrizes para criar certos tipos de casos de uso, não estabelece qualquer ligação entre os casos de uso e o DR. No trabalho de Achour [Achour et al., 1999] embora sejam feitas algumas sugestões sobre o estilo e o conteúdo que deve ser usado nas especificações dos casos de uso, o próprio autor, por meio de um experimento, observou que elas não eram suficientes para melhorar essa tarefa.

Assim, considerando especificamente os Modelos de Casos de Uso, existem na literatura algumas diretrizes que ajudam a sua construção, mas elas não são escritas de uma forma procedimental tal que a experiência e a subjetividade do projetista interfiram o mínimo possível.

Dessa forma, para reduzir o grau de subjetividade e a necessidade da experiência do projetista, definiu-se a técnica de leitura TUCCA (*Technique for Use Case model*

construction and Construction-based requirements document Analysis), que tem o objetivo principal de auxiliar na construção de Modelos de Casos de Uso, fornecendo procedimentos mais sistemáticos que direcionem a modelagem e permitam que o modelo construído não apresente tanta variação se desenvolvido por diferentes pessoas. À medida que o modelo é construído, essa técnica também propicia condições de inspecionar o Documento de Requisitos.

Este artigo apresenta os principais passos da técnica TUCCA e também comenta dois estudos experimentais que foram realizados com ela: o primeiro estudo teve como objetivo avaliar a viabilidade de aplicação da técnica, no que diz respeito ao seu entendimento, facilidade de uso, ajuda na elaboração dos modelos e padronização obtida nos modelos elaborados comparando-a com a abordagem Ad-Hoc; o segundo estudo teve por objetivo comparar a técnica com a abordagem Ad-Hoc para construção dos modelos e com a técnica Checklist para a avaliação do Documento de Requisitos.

O artigo está organizado da seguinte maneira: na Seção 2 apresenta-se a técnica TUCCA fornecendo-se exemplos de sua aplicação, passo a passo. Na Seção 3 apresentam-se os estudos experimentais realizados com ela e na Seção 4 apresentam-se as conclusões, trabalhos em andamento e futuros.

2. A Técnica de Leitura TUCCA

Técnicas de Leitura são consideradas pontos chave para se realizar algumas atividades de Engenharia de Software, como por exemplo, atividades de verificação, validação, evolução, manutenção e reuso [Basili et al., 1996]. Essas técnicas têm o foco em um determinado artefato e fornecem ao leitor um processo bem definido que os ajudam na atividade de leitura desse artefato. Como elas possuem um processo de aplicação, é possível conduzir estudos experimentais que tenham, por exemplo, o objetivo de determinar sua efetividade. Em [Basili et al., 1996] é apresentada uma árvore de técnicas de leitura que está dividida, no primeiro nível, em leitura para análise e leitura para construção. A primeira dá suporte à inspeção de algum tipo de artefato e a segunda dá suporte à elaboração de artefatos.

TUCCA foi baseada nas técnicas de leituras PBR-Usuário [Basili et al., 1996] [Shull et al., 2000] e na técnica ER1 da família de técnicas OORTs/ProDeS [Marucci et al., 2002]. PBR-Usuário tem o objetivo de detectar defeitos em Documentos de Requisitos (DR) e usa como modelo de suporte o Modelo de Casos de Uso, embora não forneça diretrizes detalhadas de como elaborá-lo. A técnica ER1 tem o objetivo de detectar defeitos em Modelos de Casos de Uso, comparando-o com o DR que deu origem ao modelo.

Dessa forma, por estar baseada nessas duas técnicas, com os objetivos mencionados, embora o objetivo principal da TUCCA seja fornecer diretrizes para construir Modelos de Casos de Uso de forma mais sistemáticos, ela também fornece, como atividade complementar, suporte para analisar o Documento de Requisitos.

TUCCA é composta de duas técnicas: AGRT (*Actor-Goal Reading Technique*) que tem o propósito de identificar os atores e seus respectivos objetivos em relação ao sistema e a UCRT (*Use Case Reading Technique*) que tem o objetivo de determinar os casos de uso e de elaborar suas especificações.

O Documento de Requisitos é o artefato básico dessas duas técnicas e considera-se que ele siga o padrão IEEE STD 830-1998 [IEEE, 1998] e que ele tenha, ao menos, as seguintes seções: Definições, Funções do Produto, Características do Usuário, Requisitos Funcionais e Requisitos Não-Funcionais.

Nas seções seguintes essas duas técnicas são apresentadas e ilustradas com base em um Documento de Requisitos de sistema de Vídeo Locadora, no qual alguns requisitos são apresentados.

2.1. AGRT (Actor-Goal Reading Technique)

A AGRT possui como entrada o DR e como saída o formulário FAO (Formulário Ator X Objetivo) e o Relatório de Discrepâncias. O objetivo da AGRT é entender o DR de forma que os candidatos a atores e seus respectivos objetivos sejam identificados. Essa informação é registrada no FAO, conforme mostra a Figura 1. Esse formulário foi baseado no trabalho de Cockburn (2001), o qual propõe a elaboração de uma lista de atores e seus respectivos objetivos tal que ele possa ser usado como um ponto inicial de discussão entre o projetista e os *stakeholders* durante a elicitación dos requisitos. Vale ressaltar que a aplicação dessa técnica deve ser feita de forma mecânica, mesmo que, aparentemente, informações inconsistentes estejam sendo registradas, como será exemplificado durante o Passo B da Fase II.

Ator	Objetivo	Referência
Candidato a Ator	Objetivos identificados para o Ator	Número da seção ou do requisito que associa o objetivo ao ator no DR

Figura 1. Formulário Ator X Objetivo (FAO)

2.1.1. Fase I – compreender a funcionalidade descrita no Documento de Requisitos

Para compreender a funcionalidade, os passos dessa fase requerem a marcação, no DR, dos substantivos e dos verbos candidatos a atores e objetivos, respectivamente, bem como das restrições e condições. A Figura 2 mostra uma pequena parte do Documento de Requisitos com esses itens marcados.

<p>Funções do Produto</p> <p>O sistema deve fornecer funcionalidades para que o funcionário possa <u>alugar fita</u> e o cliente <u>possa pesquisar fitas alugadas</u> além da possibilidade de <u>excluir clientes</u> da locadora por parte do gerente.</p> <p>Requisito Funcional 1</p> <p>- Descrição</p> <p>No estado inicial do sistema de vídeo o menu é apresentado. Do menu principal o funcionário pode escolher uma das seguintes opções:</p> <ol style="list-style-type: none"> 1. <u>Alugar uma fita</u> 2. <u>Inserir novo cliente</u> <p>.....</p> <p>Requisito Funcional 18</p> <p>- Descrição</p> <p>Apenas gerentes podem <u>excluir clientes</u></p> <p>- Entrada</p> <p>Gerente entra número da conta do cliente</p> <p>- Processamento</p> <p>Sistema exclui registro de cliente, caso o mesmo já esteja cadastrado</p> <p>- Saída</p> <p>Cliente excluído.</p>		<p>Legenda:</p> <p>Negrito: substantivo</p> <p>Sublinhado: descrição de funcionalidade</p> <p>Itálico: restrição</p>
--	--	---

Figura 2. Exemplo de marcação feita no Documento de Requisitos

2.1.2. Fase II – preencher o FAO com os atores e seus respectivos objetivos

O objetivo desta fase é preencher o FAO, com base nas marcações realizadas na fase anterior. As restrições e condições serão usadas somente na aplicação da UCRT.

- **Passo A:** registrar no FAO os atores e objetivos encontrados na seção de Requisitos Funcionais. Se houver algum objetivo que não foi associado com um

ator, verificar nas outras seções do DR se é possível identificar um ator para ele. Em caso negativo, registrar no Relatório de Discrepância.

- **Passo B:** registrar no FAO os atores e objetivos encontrados na seção Funções do Produto. O propósito deste passo é assegurar que somente as funcionalidades principais do sistema estejam declaradas nessa seção, de acordo com as recomendações do Std 830-1998 – *IEEE Recommended Practice for Software Requirements Specifications* [IEEE, 1998]. Se essa recomendação estiver contemplada no DR, as informações encontradas nessa seção já devem estar no FAO, pois deveriam ter sido encontradas na seção de Requisitos Funcionais. Se as informações não estiverem registradas no FAO deve-se relatar o fato no Relatório de Discrepância, pois existe funcionalidade ou ator nessa seção que não foi detalhada nos Requisitos Funcionais.
 - **Exemplo:** as referências, nas quais uma associação entre o objetivo “alugar fita” e ator “funcionário” foi encontrada, são adicionadas na coluna “Referências” do FAO. Essas associações ocorreram nos requisitos funcionais RF1, RF7, RF10, RF11 e RF15. A Figura 3 mostra o FAO preenchido após o passo B. Nota-se que o objetivo “alugar fita” está associado ao ator “funcionário” em decorrência da forma como está definido no DR. No entanto, o ator que deveria estar associado a esse objetivo é o “cliente”. Na Fase II, Passo B, a técnica fornece suporte para que esse fato seja corrigido.

Ator	Objetivo	Referência
Funcionário	Alugar fita	RF1, RF7, RF10, RF11, RF15, Funções do Produto
	Inserir novo cliente	RF1
	Excluir cliente	RF1, RF17
	Remover cliente	RF18
Empregado	Registrar novo cliente	RF1
Gerente	Excluir cliente	RF18, Funções do Produto
Cliente	Pesquisar fitas alugadas	Funções do Produto

Figura 3. FAO preenchido após a Fase II da AGRT

2.1.3. Fase III – eliminar as redundâncias do FAO

O objetivo desta fase é eliminar as redundâncias do FAO, isto é, possíveis atores e/ou objetivos que, embora com nomes diferentes, significam a mesma coisa:

- **Passo A:** eliminar do FAO objetivos redundantes relacionados com o mesmo ator.
 - **Exemplo:** embora os objetivos “excluir cliente” e “remover cliente” associados ao ator “funcionário” tenham nomes diferentes, eles possuem a mesma funcionalidade (Figura 3). Assim, somente um dos objetivos deve permanecer no FAO, escolhendo-se o melhor nome entre eles e agrupando as “referências” no objetivo que permaneceu. Essa redundância deve ser relatada no Relatório de Discrepâncias, para que o DR também seja corrigido. A Figura 4 mostra o FAO depois da aplicação do passo A; comparando-a com a Figura 3 pode-se observar as mudanças, ou seja, a permanência do objetivo “excluir cliente” e a junção das referências de “remover cliente” nas referências de “excluir cliente”.

Ator	Objetivo	Referência
Funcionário	Alugar fita	RF1, RF7, RF10, RF11, RF15, Funções do Produto
	Inserir novo cliente	RF1
	Excluir cliente	RF1, RF17, RF18
Empregado	Registrar novo cliente	RF1
Gerente	Excluir cliente	RF18, Funções do Produto
Cliente	Pesquisar fitas alugadas	Funções do Produto

Figura 4. Atualização do FAO depois do passo A, Fase III

- **Passo B:** eliminar do FAO atores identificados com o “nome do sistema” (em decorrência desse nome ter sido classificado como ator por existir no DR algum requisito como “A ‘Vídeo Locadora’ deve fornecer...”) ou com a palavra “sistema”(por motivo similar). Esses atores são o próprio sistema e não atores reais (aqueles que realmente interagem com o sistema). Assim, eles não devem constar como atores no Modelo de Casos de Uso e seus objetivos devem ser realocados para outros atores ou quando isso não for possível, devem ser relacionados na Lista de Objetivos Não Associados. Note que essa lista deve ser usada para corrigir o DR quando for utilizada uma estratégia para aplicar essa técnica.
- **Passo C:** eliminar objetivos redundantes relacionados com vários atores.
 - **Exemplo:** embora o objetivo “inserir novo cliente” do ator “funcionário” e o objetivo “registrar novo cliente” do ator “cliente” tenham nomes diferentes, eles têm o mesmo significado, isto é, a mesma funcionalidade. Deve-se escolher o nome mais apropriado e usá-lo para designar essa funcionalidade. Deve-se também registrar uma discrepância no Relatório de Discrepâncias, pois diferentes nomes estão sendo usados para a mesma funcionalidade. A Figura 5 mostra o conteúdo do FAO depois que o passo C foi aplicado.

Ator	Objetivo	Referência
Funcionário	Alugar fita	RF1, RF7, RF10, RF11, RF15, Funções do Produto
	Inserir novo cliente	RF1
	Excluir cliente	RF1, RF17, RF18
Empregado	Inserir novo cliente	RF1
Gerente	Excluir cliente	RF18, Funções do Produto
Cliente	Pesquisar fitas alugadas	Funções do Produto

Figura 5. FAO após a aplicação do Passo C, Fase III

2.1.4. Fase IV – verificar a seção de Requisitos Não-Funcionais

O propósito dessa fase é assegurar que não exista informação incorreta na Seção de Requisitos Não-Funcionais. Se houver, deve-se relatar o problema no Relatório de Discrepância.

Ao final da aplicação da AGRT, possíveis redundâncias de atores e/ou funcionalidades advindas do DR já não constam mais do FAO.

2.2. UCRT (*Use Case Reading Technique*)

A UCRT possui como entrada o DR e o FAO, gerado pela AGRT, e como saída o Diagrama de Casos de Uso, as Especificações dos Casos de Uso e o Relatório de Discrepâncias. Seu objetivo é a elaboração do Modelo de Casos de Uso. Os objetivos identificados no FAO são analisados e transformados em Casos de Uso, o que fica facilitado em decorrência da eliminação das redundâncias pela AGRT, as quais poderiam causar alguma confusão para defini-los.

Durante a aplicação da UCRT são construídos dois formulários: o FCUP – Formulário de Casos de Uso Preliminares (Figura 6) e o Formulário de Especificação de Casos de Uso (Figura 7), o qual foi baseado em [Kulal & Guiney, 2000], [Schneider & Winters, 2001], [Cockburn, 2001] e [Ryser & Glinz], sendo que as diretrizes para preenchimento dos campos “curso normal” e “cursos alternativos” foram baseadas nas diretrizes propostas em [Cockburn, 2001] e [Achour et al., 1999].

Nº	Ator	Caso de Uso	Referência	Relacionamento	Candidato a Include	Especificado?
Número seqüencial	Nome do ator (vem do FAO)	Nome do Caso de Uso (pode ser uma junção dos objetivos do FAO)	Agrega todas as referências associadas com os objetivos que deram origem ao Caso de Uso	Associa os casos de uso relacionados de acordo com o conjunto de referências	Associa casos de Uso que podem ser do tipo <<include>> para outros casos de uso	Marcar S/N para informar se o caso de uso já foi ou não especificado

Figura 6. FCUP – Formulário de Casos de Uso Preliminares

Número do Caso de Uso:	
Nome do Caso de Uso	O nome do caso de uso fornece uma identificação única
Resumo	Uma breve descrição do caso de uso
Ator Participante (Primário)	Aquele que fornece as informações
Ator Operador (Secundário)	Aquele que opera o computador
Ator Genérico	Quando for usado o relacionamento de generalização para atores
Pré-condição	Pré-condições são situações que devem ser satisfeitas para o caso de uso
Curso Normal	Descrição de ações passo-a-passo, eventos e respostas do sistema geradas durante o comportamento normal, esperado do sistema
Curso Alternativo	Lista de fluxo de ações que são geradas por comportamento excepcional ou alternativo
Evento Disparador	Nomes dos eventos que disparam a execução do caso de uso
Include	Números dos casos de uso associados pelo estereótipo <<include>>
Extend	Números de casos de uso associados pelo estereótipo <<extend>>
Requisitos Funcionais	Números dos requisitos funcionais do DR associados ao caso de uso
Requisitos Não-Funcionais	Números dos requisitos não-funcionais do DR associados ao caso de uso
Autor	Identificação de quem escreveu e mantém o caso de uso atualizado
Date	A data em que o caso de uso foi criado
Versão	Código associado à versão do caso de uso

Figura 7. Formulário de Especificação de Casos de Uso

2.2.1. Fase I – compreender os objetivos que foram registrados no FAO

O propósito desta fase é compreender os objetivos que foram registrados no FAO para criar os casos de uso preliminares.

- **Passo A:** transformar em caso de uso os objetivos do FAO que possuem na coluna “Referência” uma associação com a seção “Funções do Produto”. O propósito de fazer isso é que nessa seção são descritas as grandes funcionalidades do sistema, as quais sempre acabam se tornando casos de uso. Faça uma marcação nos respectivos objetivos no FAO.
 - **Exemplo:** satisfazem essa condição os objetivos “alugar fita”, “excluir cliente” e “pesquisar fitas alugadas” da Figura 5.
- **Passo A.1:** completar as informações nas colunas do FCUP associadas com o caso de uso identificado e fazer uma marcação nos objetivos usados para gerar esses casos de uso. Além disso, os atores com o mesmo objetivo devem ser agrupados, bem como suas referências. A Figura 8 ilustra a aplicação do passo A. Observe a junção das informações do ator gerente (o qual possuía como referência a seção Funções do Produto) com as do ator “cliente”, referentes ao caso de uso “excluir cliente”.

Nº	Ator	Caso de Uso	Referência	Relacionamento	Candidato a Include	Especificado?
1	Funcionário	Alugar Fita	RF1, RF7, RF10, RF11, RF15, Funções do Produto			
2	Cliente	Pesquisar fitas alugadas	Funções do Produto			
3	Gerente Cliente	Excluir cliente	RF18, Funções do Produto, RF1, RF17, RF18			

Figura 8. FCUP após a aplicação do Passo A, Fase I

- **Passo B:** identificar no FAO os objetivos que possuem o mesmo conjunto de referências, independentemente dos atores associados, pois eles podem ser passos de um mesmo caso de uso, isto é, podem estar associados à mesma funcionalidade. Transforme esses objetivos em um ou mais casos de uso, de acordo com a funcionalidade descrita no DR e faça uma marcação nos objetivos do FAO que foram utilizados nesse passo.
 - **Exemplo:** na Figura 5 os atores “funcionário” e “empregado” possuem o objetivo “inserir novo cliente” associado ao mesmo conjunto de referências (RF1). Neste caso, o objetivo é o mesmo e deve ser transformado em um caso de uso no FCUP.
- **Passo B.1:** complete as colunas do FCUP relacionadas com o caso de uso que foi definido no passo B e faça uma marcação nos objetivos do FAO que foram utilizados. A Figura 9 mostra o FCUP após a aplicação do passo B.

Nº	Ator	Caso de Uso	Referência	Relacionamento	Candidato a Include	Especificado?
1	Funcionário	Alugar fita	RF1, RF7, RF10, RF11, RF15, Funções do Produto			
2	Cliente	Pesquisar fitas alugadas	Funções do Produto			
3	Gerente Funcionário	Excluir cliente	RF18, Funções do Produto, RF1, RF17, RF18			
4	Cliente Empregado	Inserir novo cliente	RF1			

Figura 9. FCUP após a aplicação do Passo B, Fase I

- **Passo C:** transforme em casos de uso os objetivos que sobraram no FAO.

2.2.2. Fase II – fazer a especificação dos casos de uso identificados

O propósito desta fase é especificar os casos de uso relacionados no FCUP e identificar possíveis associações entre eles usando os estereótipos <<include>> e/ou <<extend>> da UML.

- **Passo A:** verifique no FCUP se existe algum caso de uso cuja coluna “Referência” possui referência apenas à Seção Funções do Produto.
 1. Se houver informação suficiente para especificar o caso de uso, isso significa que a Seção Funções do Produto possui muitos detalhes, os quais estão em lugar errado do DR. Especifique o caso de uso de acordo com as instruções do passo B e marque um “S” na coluna “Especificado?”.
 2. Se não houver informação suficiente para especificar o caso de uso, significa que essa funcionalidade foi mencionada na Seção Funções do Produto e não foi detalhada por nenhum Requisito Funcional do DR. Marque um “N” na coluna “Especificado?”.

Em ambos os casos relate o problema no Relatório de Discrepâncias pois ou a Seção de Funções do Produto possui informação que não deveria possuir ou a funcionalidade não foi descrita nos Requisitos Funcionais.

– **Exemplo:** na Figura 9 o caso de uso “pesquisar fitas alugadas” do ator “cliente” possui na coluna “Referência” somente uma referência, à Seção Funções do Produto.

- **Passo B:** selecione o próximo caso de uso ainda não especificado (veja a coluna “Especificado?”) que possui o menor número de referências associado a ele. Chame esse caso de uso de Cbase. O “No.” do Cbase deve ser associado, na coluna “Relacionamento” com todos os casos de uso já especificados que possuírem na coluna “Referência” um subconjunto das referências do CBase e vice-versa.

– **Exemplo:** considerando que os casos de uso 2, 3 e 4 já foram especificados, o próximo caso de uso selecionado é o “Alugar Fita” o qual contém todo o conjunto de referências associado ao caso de uso “Inserir novo cliente”. A Figura 10 mostra o resultado da aplicação desse passo.

Nº	Ator	Caso de Uso	Referência	Relacionamento	Candidato a Include	Especificado?
1	Funcionário	Alugar fita	RF1, RF7, RF10, RF11, RF15, Funções do Produto	4		
2	Cliente	Pesquisar fitas alugadas	Funções do Produto			S
3	Gerente Funcionário	Excluir cliente	RF18, Product Functions, RF1, RF17			S
4	Cliente Empregado	Inserir novo cliente	RF1	1		S

Figura 10. FCUP completado após a aplicação do Passo B, Fase II

Após feito esse relacionamento, este passo é subdividido em sub-passos que dão suporte à especificação detalhada do caso de uso, para que os campos relacionados no Formulário de Especificação de Casos de Uso (Figura 7) sejam completados. Esses sub-passos fornecem procedimento detalhado que ajuda a identificar vários tipos de discrepâncias no DR, além de na própria especificação, ajudando a identificar generalização de atores, identificação do curso normal e dos cursos alternativos, casos de uso que são do tipo <<include>>, etc. Por motivo de espaço, não é possível exemplificar a aplicação completa desse passo.

Porém, vale ressaltar alguns pontos tratados:

1. Observe que o caso de uso “alugar fita” relacionado ao ator “funcionário” é uma funcionalidade exigida pelo ator “cliente”. Nesse caso, os sub-passos não mostrados guiarão o leitor na identificação do ator “funcionário” como sendo o operador (aquele que opera o sistema) enquanto o ator participante (aquele que fornece as informações) será identificado como o “cliente”. Essa característica veio do fato que o Documento de Requisitos possuía um defeito de que apenas o ator operador “funcionário” estava relacionado com o objetivo.
2. O caso de uso número 3 (“excluir cliente”) é uma funcionalidade, que segundo a restrição marcada no DR (vide requisito funcional RF18), deve ser realizada “apenas” pelos gerentes. Assim, durante a aplicação dos passos de especificação dos casos de uso, essa discrepância será identificada e preenchida no Relatório de Discrepâncias.

3. A coluna “Include” do FCUP será preenchida durante a especificação dos casos de uso de modo que os casos de uso relacionados com o estereótipo <<include>> sejam estabelecidos.
 - **Passo C:** decidir se o relacionamento entre os casos de uso com o estereótipo <<include>> pode ser descartado, possibilitando a junção dos casos de uso. Quando na coluna “Candidato a Include” houver apenas um número, o caso de uso daquela linha do FCUP pode, eventualmente, se tornar passos do caso de uso cujo número está na coluna.
 - **Passo D:** associar os requisitos Não-Funcionais com os casos de uso especificados. Esse passo permite que, além da associação possam ser identificadas discrepâncias referentes a requisitos funcionais encontrados na Seção de Requisitos Não-Funcionais.

3. Resumo dos Estudos Experimentais Conduzidos com a Técnica TUCCA

Dois estudos experimentais já foram realizados utilizando a técnica TUCCA. O primeiro teve como objetivo principal avaliar o uso da técnica, isto é, avaliar a viabilidade de aplicação, seu entendimento e o quanto ela auxilia na construção de Modelos de Casos de Uso, de forma que, independentemente da experiência e da subjetividade da pessoa, os Casos de Uso elaborados por pessoas diferentes podem ser parecidos comparando-se esses modelos com modelos gerados de forma Ad-Hoc. No segundo estudo, já possuindo alguns indicativos advindos do primeiro estudo, o objetivo principal foi comparar TUCCA com Ad-Hoc, no que diz respeito à elaboração dos Modelos de Casos de Uso, isto é, se a técnica realmente ajuda a construir modelos mais similares (como havia indícios do primeiro estudo) e comparar TUCCA com Checklist, para avaliar a ajuda na identificação de defeitos no Documento de Requisitos.

Ressalta-se que ambos os estudos foram realizados no contexto acadêmico e seus resultados não podem ser generalizados para a indústria devido às características dos participantes e à complexidade dos Documentos de Requisitos utilizados.

3.1. Primeiro Estudo: Viabilidade de Aplicação da TUCCA

- **Objetivo:** avaliar a viabilidade de aplicação da TUCCA, sua facilidade de compreensão e sua contribuição para construir Modelos de Casos de Uso mais parecidos com um modelo Oráculo do que utilizando uma abordagem Ad-Hoc.
- **Participantes:** 18 alunos de graduação dos cursos de Bacharelado em Ciência da Computação e Engenharia de Computação da UFSCar/SP, cursando a disciplina Engenharia de Software, divididos em 6 grupos [Belgamo & Fabbri, 2004]. Todos participantes utilizaram um DR que não era de seu conhecimento para que este não interferisse na utilização da técnica.
- **Artefatos utilizados:** seis DRs elaborados pelos próprios alunos, revisados com um Checklist e posteriormente corrigidos, e um modelo Oráculo construído por uma pessoa conhecedora da técnica. Os DRs tratavam de sistemas de informação que, embora diferentes em tema, possuíam complexidade bastante similar (vídeo locadora, consultório, etc.).
- **Métricas usadas para avaliar os resultados:**
 - **Associação ‘Ator-Caso-de-Uso’:** número de associações ‘ator-caso-de-uso’, relacionadas ao modelo oráculo, encontrado pelo participante.
 - **Ocorrências de Associação ‘Ator-Caso-de-Uso’:** número de vezes que a associação ‘ator-caso-de-uso’ foi encontrada (cada participante tem a chance de

encontrar cada associação ‘ator-caso-de-uso’). O número máximo de ocorrências de cada associação ‘ator-caso-de-uso’ é igual ao número de participantes. O número total de ocorrências de todas as associações ‘ator-caso-de-uso’ (TotalOc) é calculado pela Equação 1:

$$\text{TotalOc} = \sum_{i=1}^n (x_i) \quad \text{onde } x_i \text{ é o número de associações ‘ator-caso-de-uso’ encontradas pelo participante } i \quad (1)$$

- o **Efetividade**: média percentual das associações ‘ator-caso-de-uso’ encontradas por um grupo de participantes. A efetividade é calculada pela Equação 2:

$$\left(\sum_{i=1}^n (x_i / y) \right) * 100 / n \quad \text{Onde } x_i \text{ é o número de associações ‘ator-caso-de-uso’ encontradas pelo participante } i; y \text{ é o número total de associações ‘ator-caso-de-uso’ no Modelo de Casos de Uso e } n \text{ é o número de participantes no grupo} \quad (2)$$

- o **Eficiência**: média das associações ‘ator-caso-de-uso’ encontradas por cada participante, por hora. A eficiência é calculada pela Equação 3.

$$\left(\sum_{i=1}^n (x_i / k_i) \right) / n \quad \text{onde } x_i \text{ é o número de associações ‘ator-caso-de-uso’ encontradas pelo participante } i; k_i \text{ é o esforço (em horas) usado pelo participante } i \text{ e } n \text{ é o número de participantes no grupo.} \quad (3)$$

- **Resultados:**

- o Os modelos construídos com TUCCA foram mais padronizados do que os construídos com Ad-Hoc, isto é, os modelos relacionados ao mesmo DR foram muito similares entre si, bem como similares ao modelo Oráculo.
- o Utilizou-se a ANOVA [MiniTab, 2005] com um grau de confiança de 95% para analisar os resultados e chegaram-se nos seguintes valores: (i) em relação à **Efetividade**, TUCCA foi melhor que Ad-Hoc quando consideradas as influências da “Técnica” (p-value <0.001) e “Técnica + Documento de Requisitos” (p-value = 0.028); quando considerada a influência apenas do “Documento de Requisitos” não foi possível chegar a nenhuma conclusão (p-value = 0.983); (ii) em relação à **Eficiência**, TUCCA foi melhor que Ad-Hoc quando consideradas as influências do “Documento de Requisitos” (p-value <0.001) e “Técnica + Documento de Requisitos” (p-value = 0.001); quando considerada a influência apenas da “Técnica” não foi possível chegar a nenhuma conclusão (p-value = 0.231).
- o TUCCA facilitou a identificação de Casos de Uso mais padronizados, uma vez que a pessoa tem um conjunto de passos bem específicos para seguir.
- o TUCCA aumentou a efetividade em identificar Casos de Uso e ajudar na decisão de agrupar ou separar funcionalidades contribuindo com os aspectos de coesão do sistema.
- o TUCCA ajudou na definição de Casos de Uso que representam com mais precisão a funcionalidade do sistema, evitando a criação de Casos de Uso com funcionalidades similares, ou funcionalidades extras que poderiam ser supostas pelo projetista.

Salienta-se que durante o estudo, os defeitos identificados durante a aplicação da TUCCA não foram corrigidos no Documento de Requisitos. Entretanto, na prática, é necessário que se defina uma estratégia de aplicação dessas técnicas para assegurar que

ao final, o Modelo de Casos de Uso elaborado seja um modelo preciso, pois à medida que os defeitos são encontrados, o DR deve ser corrigido.

A Tabela 1 mostra alguns resultados, de acordo com as métricas, para os seis documentos utilizados:

Tabela 1. Métricas coletadas para os Documentos de Requisitos

Doc	Técnica	(Associação 'Ator-Caso-de-Uso') / (Total 'Associação Ator-Caso-de-Uso')	(Ocorrências de Associação 'Ator-Caso-de-Uso') / (TotalOcorrência)	Efetividade	Eficiência
A	TUCCA	7/7 (100%)	21/21	100	2.15
	Ad-Hoc	4/7 (57%)	8/21	38.10	1.20
B	TUCCA	16/16 (100%)	45/48	93.75	6.10
	Ad-Hoc	13/16 (81%)	23/48	47.92	4.78
C	TUCCA	14/15 (93%)	41/45	91.11	2.61
	Ad-Hoc	14/15 (93%)	28/45	62.22	3.47
D	TUCCA	14/14 (100%)	38/42	90.48	2.24
	Ad-Hoc	11/14 (78%)	16/42	38.10	3.71
E	TUCCA	22/24 (91%)	60/72	83.3	1.28
	Ad-Hoc	22/24 (91%)	48/72	66.67	6.39
F	TUCCA	2/2 (100%)	6/6	100	0.41
	Ad-Hoc	1/2 (50%)	3/6	50	0.57

3.2. Segundo Estudo: TUCCA x Ad-Hoc e TUCCA x Checklist

- **Objetivo:** comparar a técnica TUCCA com Ad-Hoc no que diz respeito à elaboração de Modelos de Casos de Uso mais padronizados e mais parecidos entre si e comparar TUCCA com Checklist no que diz respeito à detecção de defeitos em Documentos de Requisitos.
- **Participantes:** 40 alunos de graduação dos cursos de Bacharelado em Ciência da Computação e Engenharia de Computação da UFSCar/SP, cursando a disciplina Engenharia de Software. 20 alunos aplicaram a técnica TUCCA (10 alunos no documento de requisitos original e 10 alunos no documento de requisitos corrigido pela técnica Checklist) e 20 aplicaram Ad-Hoc (10 alunos no documento de requisitos original e 10 alunos no documento de requisitos corrigido pela técnica Checklist).
- **Artefatos utilizados:** um Documento de Requisitos – PG (*Parking Garage Control System*), versão em português, utilizado em outros estudos experimentais realizados no contexto do projeto Readers [Maldonado et al., 2001]. Não foi usado Modelo Oráculo.
- **Métricas usadas para avaliar os resultados:**
 - **Efetividade de Casos de Uso:** essa métrica foi estabelecida analisando os modelos gerados por todos os participantes e definindo-se a partir desses modelos um grupo de casos de uso que representava as funcionalidades do DR. Estabelecido esse grupo, a efetividade de cada participante corresponde a quantos casos de uso ele identificou em relação ao grupo.
 - **Efetividade de Defeitos:** número de defeitos encontrados no DR.
- **Resultados:** foram analisados em relação ao aspecto de construção do Modelo de Casos de Uso (Seção 3.2.1) e em relação ao aspecto de análise do Documento de Requisitos (Seção 3.2.2). Neste caso, avaliou-se a contribuição da TUCCA quando: (a) esta foi aplicada no DR inspecionado, usando Checklist, e corrigido após a inspeção e (b) quando a TUCCA foi aplicada sem que o Documento de Requisitos

tivesse sido inspecionado antes. Essas análises são apresentadas nas seções subseqüentes.

3.2.1. TUCCA vs Ad-Hoc

Para analisar os Modelos de Casos de Uso construídos com TUCCA e Ad-Hoc, considerou-se quão homogêneos eles eram e se eram uma boa representação do DR. A Tabela 2 mostra que os números de Casos de uso gerados pela abordagem Ad-Hoc foram bastante variáveis (de 3 a 25) enquanto que com TUCCA esse número foi bem mais próximo (de 6 a 8). Além disso, os Casos de Uso definidos com TUCCA tinham uma semântica bem mais próxima. A Tabela 3 mostra os valores da estatística descritiva gerados a partir dos dados da Tabela 2.

Tabela 2. Número de Casos de Uso definidos por cada participante

Participantes TUCCA	# Caso Uso	Participantes Ad-Hoc	# Caso Uso
G1	8	A1	25
G2	6	A2	8
G3	7	A3	13
G4	6	A4	14
G5	7	A5	11
G6	6	A6	8
G7	7	A7	9
G8	6	A8	9
G9	8	A9	3
G10	6	A10	4

Tabela 3. Estatística Descritiva

	TUCCA	Ad-Hoc
Média	6,7	10,4
Mediana	6,5	9
Desvio Padrão	0,82	6,18
Variância	0,67	38,26

O Desvio Padrão de TUCCA é substancialmente menor que o do Ad-Hoc. Como na análise considerou-se a semântica dos Casos de Uso gerados e não apenas a quantidade, esse resultado provê evidências do auxílio fornecido por TUCCA para que se construam Casos de Uso mais padronizados, isto é, diferentes participantes propuseram casos de uso bastante parecidos.

Nesse aspecto de construção de Modelos de Casos de Uso, a questão a ser respondida era: “Há diferença em se construir Modelos de Casos de Uso com TUCCA e com Ad-Hoc?”. Para responder essa questão, as seguintes hipóteses foram formuladas:

H0: Não há diferença em se construir Modelos de Casos de Uso com as técnicas TUCCA e Ad-Hoc.

Ha: A quantidade e a semântica dos Casos de Uso gerados com TUCCA são mais padronizadas do que as geradas com Ad-Hoc.

Considerando um nível de confiança de 95%, ao aplicar-se o teste Mann-Whitney obteve-se um $p\text{-value}=0.0351$, o que significa que a hipótese H0 pode ser rejeitada. Isso significa que a técnica influencia os Modelos de Casos de Uso gerados. Esse resultado dá suporte aos valores da estatística descritiva apresentada na Tabela 3, bem como à análise dos modelos realizada pelo experimentador. Em outras palavras, a técnica TUCCA auxilia a construção de Modelos de Casos de Uso mais padronizados. No entanto, para que se possa generalizar esse resultado, esse experimento deve ser feito em outros ambientes, como por exemplo, em um ambiente industrial, com pessoas experientes na modelagem com Casos de Uso, para verificar se TUCCA também auxilia nesse contexto.

3.2.2. TUCCA vs Checklist

Para comparar TUCCA e Checklist no que diz respeito à análise do Documento de Requisitos, consideraram-se as seguintes categorias: *defeitos* – itens relatados pelos participantes que realmente eram defeitos do DR e *não defeitos* – itens relatados pelos participantes que não correspondiam a defeitos no DR. A comparação foi realizada de duas maneiras: aplicando TUCCA depois de ter aplicado o Checklist e corrigido o DR e aplicando TUCCA no DR original, sem ter sido inspecionado e corrigido previamente.

a) TUCCA depois de Checklist

O objetivo desta comparação foi avaliar o aspecto complementar entre TUCCA e Checklist, ou seja, verificar se TUCCA era capaz de detectar defeitos adicionais, depois que o Checklist já tinha sido aplicado. Com o Checklist foram detectados 17 defeitos que foram corrigidos. Após isso, aplicando-se TUCCA, mais 10 defeitos foram encontrados. Em relação aos não-defeitos, o Checklist encontrou 26 e TUCCA encontrou 15. Isso mostra que com a aplicação de TUCCA, ao menos nesse estudo, os participantes relataram menos coisas que não eram defeitos do que com a aplicação do Checklist.

Além da análise quantitativa, verificaram-se os tipos de defeitos encontrados e as principais observações a serem feitas são:

- A maioria dos defeitos detectados com TUCCA, 6 defeitos, foi informação relativa a requisitos Não-Funcionais escritos como Requisitos Funcionais.
- 3 dos defeitos encontrados foram relacionados a muita informação agrupada em um único requisito funcional, requisitos redundantes e informação escrita em lugar errado do DR.
- 1 defeito estava relacionado com a estrutura do DR e apontou a ausência de glossário, embora na técnica TUCCA não haja, explicitamente, nenhum passo que explore esse tipo de defeito.

b) TUCCA sem Checklist

O objetivo desta comparação foi verificar se os defeitos encontrados pela TUCCA no DR original eram os mesmos de quando se aplicou TUCCA depois do Checklist. Aplicando-se TUCCA no DR original encontraram-se 23 defeitos e 10 não-defeitos.

A Figura 11 resume esses dados, mostrando a quantidade de defeitos encontrados por Checklist no DR original, por TUCCA aplicada depois do Checklist e por TUCCA aplicada no DR original.

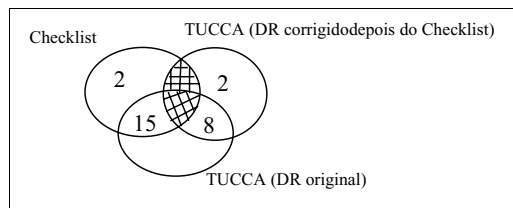


Figura 11. Total de defeitos encontrados pelas diferentes alternativas exploradas

A Figura 11 mostra os seguintes resultados:

- A área hachurada sempre será vazia, pois corresponde à aplicação de TUCCA depois que os erros encontrados pelo Checklist foram corrigidos.
- 8 defeitos foram identificados somente pela TUCCA, tanto quando ela foi usada no DR original, como quando foi usada no DR corrigido.
- Dos 17 defeitos encontrados pelo Checklist, 15 foram encontrados pela TUCCA.
- 4 defeitos não foram encontrados pela TUCCA quando ela foi aplicada no DR original. Dois deles foram encontrados pela TUCCA quando ela foi aplicada no DR corrigido, depois da aplicação do Checklist e dois foram encontrados somente pelo Checklist. Esse fato fornece evidências do aspecto complementar entre Checklist e TUCCA:
 - detectados por TUCCA: ausência de glossário e omissão de funcionalidade.
 - detectados pelo Checklist: omissão de funcionalidade e expressão matemática errada.

4. Conclusões e Trabalhos Futuros

Este artigo apresentou a técnica de leitura TUCCA – *Guidelines for Use Case Construction and Requirements document Analysis* – que tem como principal objetivo auxiliar na construção de Modelos de Casos de Uso e, associado a essa tarefa de construção, propicia condições de se analisar o Documento de Requisitos com o objetivo de detectar defeitos no mesmo. Ela é composta de duas leituras: a AGRT – *Actor X Goal Reading Technique* – que tem por objetivo identificar os atores e os objetivos que eles têm do sistema e a UCRT – *Use Case Reading Technique* – que tem o objetivo de construir os Modelos de Casos de Uso, isto é, o diagrama e as especificações. Os principais passos da técnica foram descritos e exemplificados a fim de mostrar a evolução do processo e ilustrar alguns pontos em que defeitos no Documento de Requisitos podem ser encontrados em decorrência da atividade de construção do modelo.

Dois estudos experimentais já foram realizados com o uso da TUCCA e seus resultados foram comentados. O primeiro estudo teve como objetivo avaliar a viabilidade de aplicação da técnica e sua contribuição em relação a uma abordagem Ad-Hoc de construção de Modelos de Casos de Uso. Os resultados mostraram que a técnica auxilia nessa atividade, tornando-a mais sistemática e menos dependente da experiência e subjetividade do projetista. Notou-se que modelos construídos por diferentes pessoas foram bastante similares quando a técnica foi utilizada. O segundo estudo teve por objetivo comparar TUCCA com Ad-Hoc para a construção dos Modelos de Casos de Uso e TUCCA com Checklist para a detecção de defeitos em Documentos de Requisitos. Em relação à atividade de construção, os resultados confirmaram os anteriores e, em relação à atividade de análise do Documento de Requisitos, os resultados mostraram que existe um aspecto complementar entre TUCCA e Checklist, pois houve vários erros que foram encontrados apenas com a TUCCA, mas houve também alguns erros que essa técnica não conseguiu identificar.

Para um uso mais efetivo dessas técnicas está-se definindo uma estratégia de aplicação das mesmas, considerando-se o processo de desenvolvimento de software. Em decorrência dos resultados dos estudos experimentais e das características algorítmicas de diversos passos da técnica, estão sendo conduzidos trabalhos no sentido de automatizar algumas de suas fases. Para facilitar a automatização outros aspectos têm sido explorados, como por exemplo, uma padronização para o Documento de Requisitos. Além disso, também já foi concluído um estudo comparando TUCCA com

PBR-Usuário, para avaliar a capacidade de detecção de defeitos em Documentos de Requisitos e seus resultados estão sendo analisados. Salienta-se que para uma maior generalização dos resultados, faz-se necessário um estudo com pessoas bastante experientes na modelagem de requisitos utilizando Casos de Uso.

5. Referências

- [OMG, 2003] Object Management Group. Unified Modeling Language Specification. Version 1.5. 2003. URL: <http://www.omg.org/uml/>.
- [Basili et al., 1996] Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sorumgard, S., and Zelkowitz, M. "The empirical investigation of perspective-based reading". *Empirical Software Engineering: An International Journal*. 1(2): 133-164, 1996.
- [Shull et al., 2000] Shull, F., Rus, I., and Basili, V.R. "How Perspective-Based Reading Can Improve Requirements Inspections". *IEEE Computer*, 33(7): 73-79, 2000
- [Travassos et al., 2002] Travassos, G.H., Shull, F., Carver, J., and Basili, V.R. Reading Techniques for OO design inspections. Technical Report CS-TR-4353, UMIACS-TR-2002-33, University of Maryland, Maryland, 2002.
- [Marucci et al., 2002] Marucci, R. A., Fabbri, S. C. P. F., Maldonado, J. C., Travassos, G. H. "OORTs/ProDeS: Definição de Técnicas de Leitura para um Processo de Software Orientado a Objetos", Simpósio Brasileiro de Qualidade de Software, 1., Gramado, RS, Brasil, Outubro, 2002.
- [Jacobson et al., 1992] Jacobson, I, *et al.* Object-Oriented Software Engineering - A Use Case Driven Approach, Addison-Wesley Publish Company, 1992.
- [Kulak & Guiney, 2000] Kulak, D.; Guiney, E. *Use Cases: Requirements in Context*. Addison-Wesley, 2000.
- [Schneider & Winters, 2001] Schneider, G., Winters, J. P. *Applying Use Cases, A Practical Guide*. Second Edition, Addison-Wesley, 2001.
- [Cockburn, 2001] Cockburn, A. *Writing Effective Use Cases*. Boston MA: Addison-Wesley, 2001.
- [Achour et al., 1999] Achour, Ben. C., Rolland, C., Maiden, N. A. M. and Souveyet, C. "Guiding Use Case Authoring: Results of an Empirical Study", *Proc. of the 4th International Symposium on Requirements Engineering*, 1999.
- [IEEE, 1998] IEEE Recommended Practice for Software Requirements Specifications, Std 830-1998, 1998.
- [Ryser & Glinz, 2000] Ryser, J., Glinz, M. "Using Dependency Charts to Improve Scenario-Based Testing", *17th International Conference on Testing Computer Software – TCS'2000*, Washington, D.C., 2000.
- [Belgamo & Fabbri, 2004] Belgamo, A. e Fabbri, S. "Constructing Use Case Model by Using a Systematic Approach: Description of a Study", *Proc. of the VII Workshop on Requirements Engineering*, Tandil, Argentina, 2004.
- [Maldonado et al., 2001] Maldonado, J. C., Martimiano, L. A. F., Dória, E. S., Fabbri, S. C. P. F., Mendonça Neto, M. G. Readers Project: Replication of Experiments. A Case Study using Requirements Documents In: ProTeM - CC 2001 International Cooperation Projects Evaluation Workshop, 2001, Rio de Janeiro. Proceedings of the 2001 ProTeM International Cooperation Projects Evaluation Workshop. Brasília: Conselho Nacional de Desenvolvimento Científico e Tecnológico, 2001. v.1. p.85 – 117.
- [MiniTab, 2005] MiniTab. url: <http://www.minitab.com>. último acesso: 18/06/2005.