

TestCheck – Uma Abordagem Baseada em Checklist para Inspeccionar Artefatos de Teste de Software

Jardelane Brito, Arilo Claudio Dias Neto

¹Instituto de Computação (IComp) – Universidade Federal do Amazonas (UFAM)
Av. General Rodrigo Octávio, 6.200, Campus Universitário Senador Arthur Virgílio
Filho – Setor Norte - Manaus - CEP 69.077-000 – Manaus – AM – Brasil

{jardelane.brito,arilo}@icomp.ufam.edu.br

Abstract. *The quality of testing applied to a software project is an important factor to reach the quality of the final product. Several strategies can be applied to evaluate its quality. A possible strategy is to ensure the quality of artifacts produced during testing process. Thus, with the purpose of evaluating the quality of testing artifacts, contributing to defects detection before tests execution, this paper presents the development of a checklist-based approach, TestCheck, for inspection of software testing artifacts (test plans, cases, and procedures). In order to evaluate the proposed approach, three experimental studies were conducted. They analyzed and indicated the efficacy, efficiency, and maturity of the proposed approach in relation to detection of defects in testing artifacts.*

Resumo. *A qualidade dos testes aplicados em um projeto de software é um fator determinante para a qualidade do produto final. Diversas estratégias podem ser aplicadas para avaliação da sua qualidade. Uma delas é garantir a qualidade dos artefatos produzidos ao longo do processo de testes. Este artigo apresenta os resultados de uma dissertação de mestrado onde foi desenvolvida uma abordagem baseada em checklist, TestCheck, para inspeção de artefatos de teste de software (planos, casos e procedimentos de teste). Para avaliar a abordagem proposta, foram realizados três estudos experimentais. Eles analisaram e indicaram a eficácia, eficiência e amadurecimento da abordagem proposta em relação à detecção de defeitos em artefatos de teste.*

Palavra Chave: *Inspeção de Software, Teste de Software, Artefatos de Teste.*

1. Introdução

O contexto desta pesquisa está relacionado principalmente ao cenário da aplicação de testes em projetos de software. Teste de software consiste em uma investigação experimental conduzida para prover informações aos usuários e envolvidos no processo sobre a qualidade do software em teste no contexto no qual este será operado. Isso inclui, mas não consiste apenas, um processo de executar um programa com a intenção de revelar falhas, apresentando uma análise dinâmica do produto, com atividades planejadas previamente e de modo sistemático (KANER, 2006).

A busca por qualidade de Software fez com que diversas técnicas, ferramentas e estratégias de automação fossem desenvolvidas para tornar os Testes de Software mais

eficientes. Apesar da grande variedade de soluções propostas, o desafio fundamental do teste de software é revelar falhas em um software a ser desenvolvido, e na prática ainda é largamente dependente do desempenho de testadores a partir da execução de testes manuais (ITOKEN et al., 2007). Ainda assim, os testes apresentam bons resultados quanto à qualidade, porém poderiam ser otimizados com o uso de ferramentas para apoiar os testadores neste processo visando à qualidade dos testes em um projeto, evitando que softwares sejam lançados no mercado inadequadamente.

Tendo em vista a importância dos testes como atividade de garantia da qualidade do software e os problemas enfrentados por conta de erros humanos, surge a necessidade de novos mecanismos que auxiliem os testadores neste processo visando à qualidade dos testes em um projeto de software. Diversas estratégias podem ser aplicadas para apoiar a obtenção de qualidade durante as atividades de teste de software, sendo uma delas as atividades de inspeção de software (FAGAN, 1976). Esta técnica é um método de revisão de software que se destaca por ser rigoroso, formal e executado com o objetivo de examinar artefatos de software, para encontrar defeitos antecipadamente.

Assim, objetivo da pesquisa foi a integração de técnicas de inspeção no processo de testes de software a fim de avaliar a qualidade de artefatos produzidos ao longo deste processo. Com isso, pretende-se demonstrar que a utilização de técnica de inspeção em artefatos produzidos ao longo do processo de testes de software pode influenciar positivamente na qualidade desta atividade a partir da detecção de um maior número de defeitos existentes nestes artefatos quando comparado a outras técnicas de inspeção.

Portanto, este artigo apresenta os resultados de uma dissertação de mestrado onde foi desenvolvida uma abordagem baseada em checklist, *TestCheck*, para inspeção de três artefatos de teste de software (planos, casos e procedimentos de teste). Para avaliar a abordagem proposta, foram realizados três estudos experimentais, sendo dois estudos de viabilidade e um estudo de observação na indústria. Os resultados analisados indicam a eficácia, eficiência e amadurecimento da abordagem proposta em relação à detecção de defeitos em artefatos de teste, como será descrito nas próximas seções.

2. Trabalhos Relacionados

Neste trabalho, foi feita uma investigação na literatura técnica em busca de pesquisas que envolvessem aplicação de inspeções ao longo do processo de teste de software. No entanto, poucos trabalhos foram identificados na literatura técnica, o que motivou o desenvolvimento desta pesquisa.

Em (LANUBILE et al., 2007) é apresentada uma pesquisa sobre inspeção em casos de testes automatizados. De acordo com os autores, sempre que os casos de teste são defeituosos, o tempo dos desenvolvedores pode ser desperdiçado para resolver problemas que na verdade não são originários do código de produção. Devido à sua relevância no processo de desenvolvimento, postulou-se que a qualidade dos casos de teste pode ser assegurada através de inspeções de software como um complemento para a atividade de revisão informal. Nesta pesquisa, foi analisado o efeito da realização de inspeções de software em código de testes automatizados. Os inspetores utilizaram um Checklist para análise de código de teste. Os resultados indicam que inspeção de software pode melhorar a qualidade do código de teste, principalmente em relação à possibilidade de repetição dos casos testes, que é uma das qualidades mais importantes para a automação de teste. Por tanto, este resultado indica a viabilidade em aplicar

técnicas de inspeções de software em artefatos do processo de testes como mecanismo para obtenção de melhorias na qualidade dos testes de software.

Em (HEDBERG et al., 2006) foi realizada uma pesquisa focando no papel das revisões técnicas em Mobile-DTM (método de desenvolvimento de software para dispositivos móveis). Este processo consiste em típicas técnicas ágeis, tais como programação em pares, que foram reivindicadas para melhorar a qualidade do produto. Portanto, (HEDBERG et al., 2006) propõem a realização de revisões nos testes de aceitação, por apresentar detalhes técnicos vitais para o desenvolvimento, com o objetivo de verificar se os testes realmente correspondem às estórias de usuários e encontrar testes adicionais, que podem ser vitais para a avaliação da estória. No projeto descrito no artigo, a equipe implementou este procedimento de revisão e os resultados foram bem sucedidos. Segundo os autores, o teste de aceitação revisado ficou com mais qualidade e continha menos defeitos.

Em (POON et al., 2010) foi realizada uma avaliação dos testes em um experimento para avaliar se a descrição e escolha de categorias (utilizando o critério de geração de dados de teste chamado partição por categoria) para representação dos dados de entrada em testes funcionais em um projeto são bem realizadas. Após a definição das categorias, o artefato caso de teste foi gerado (somente categorias relevantes são úteis para geração de casos de teste) e submetido a dois estudos. O primeiro estudo realizado teve como principal objetivo investigar como os tipos e a quantidades de erros cometidos em uma abordagem de identificação *Ad Hoc* variam entre testadores inexperientes e experientes. Para o segundo estudo, foi entregue um checklist com objetivo de auxiliar os revisores. O objetivo principal foi avaliar a eficácia do checklist proposto, em termos de sua capacidade para ajudar os testadores a reduzir a ocorrência de categorias ausentes e problemáticas. O resultado do primeiro estudo apresentou evidências de que a experiência no desenvolvimento e teste de software não ajuda a melhorar a qualidade do conjunto de categorias identificadas utilizando uma abordagem de identificação *Ad Hoc*. Já no segundo estudo, observou-se que a utilização do checklist ajudou os testadores de software a reduzir a ocorrência de categorias ausentes e categorias problemáticas de todos os tipos. Apesar disso, este checklist possui propósito diferente do apresentado nesta pesquisa, não sendo assim possível realizar comparações entre as abordagens.

Assim, não foi encontrado nenhum trabalho que aplicasse e avaliasse a utilização da técnica de detecção de defeitos em artefatos de testes de software com o objetivo de melhorar a qualidade desses artefatos. Assim, com o objetivo de melhorar a qualidade dos testes em projetos de software, foi desenvolvida uma abordagem baseada em Checklist, denominada *TestCheck*, que visa inspecionar os artefatos produzidos ao longo do processo de testes. O objetivo com a abordagem é atuar na qualidade dos artefatos produzidos ao longo do processo de teste como mecanismo para se obter a qualidade deste processo. Na próxima seção a abordagem será apresentada com detalhes.

3. *TestCheck*: Inspeção de Artefatos de Teste baseada em Checklist

A abordagem *TestCheck* foi criada a partir de padrões existentes na literatura técnica e a experiência do grupo de pesquisa em Experimentação e Teste de Software (ExperTS¹)

¹ <http://www.icomp.ufam.edu.br/experts>

da Universidade Federal do Amazonas (UFAM) em projetos de software na indústria. A sua concepção foi baseada nas características técnicas identificadas por (LAITENBERGER e DEBAUD, 1997) para compor uma abordagem de inspeção. O intuito foi encontrar uma solução adequada para o objetivo proposto neste trabalho, que é a criação de uma abordagem que permitisse avaliar a qualidade de artefatos produzidos ao longo de um processo de teste de software. Essas características levaram à execução das seguintes tarefas: definição da técnica para detectar os defeitos dos artefatos; escolha dos artefatos de software que devem ser avaliados pela abordagem; definição dos tipos de defeitos que a abordagem busca identificar e especificar os itens que devem compor o(s) checklist(s) proposto(s), conforme descrito em (BRITO et al., 2012 ; BRITO e DIAS-NETO, 2013).

3.1. Escolha dos Artefatos de Testes a serem inspecionados

Para elaboração da abordagem *TestCheck*, usou-se como fonte para seleção dos artefatos de testes de software candidatos a serem inspecionados a lista sugerida pelo padrão IEEE-Std-829 (IEEE, 2008), pois estes se aplicam a projetos de software em diferentes metodologias de desenvolvimento (ex.: baseadas em processos ou metodologias ágeis). Os artefatos candidatos são: plano de teste, especificação de projeto de teste, especificação de caso de teste, especificação de procedimento de teste, log de teste, relatório de incidente de teste e relatório de resumo de teste.

Para escolha dos artefatos, primeiramente foi escolhida a fase do processo de teste onde se deseja aplicar inspeções. Neste momento, optou-se por avaliar artefatos produzidos durante o planejamento dos testes, como forma de minimizar e prevenir a incidência de defeitos durante a execução dos testes, além de antecipar o momento de detecção de defeitos dentro do processo de testes, o que poderia inviabilizar a atividade.

A partir da escolha da fase do processo de testes, a seleção dos artefatos a serem inspecionados foi feita com base nos resultados de um *survey* publicado em (DIAS-NETO, 2006). Neste estudo, percebeu-se a maior ênfase das organizações de software (brasileiras) na produção dos artefatos plano de teste, caso de teste e procedimento de teste ao longo de seus processos de teste. Outro fator que motivou esta escolha surgiu após uma parceria do grupo de pesquisa ExperTS com um instituto de pesquisa e desenvolvimento sediado na região Norte do Brasil, onde alunos do grupo de pesquisa atuam como membros da equipe de testes em projetos reais, e dentre suas atribuições estão a elaboração e revisão de planos, casos e procedimentos de teste. Mais detalhes sobre os artefatos selecionados para inspeção segundo o padrão IEEE-Std 829 (2008):

- Plano de Teste: descreve os objetivos dos testes, níveis de teste a serem aplicados, divisão de tarefas, atividade, cronograma e recursos necessários para os testes, identificação dos riscos relacionados aos testes, abordagem de teste a ser aplicada e critérios para aceitação e rejeição do software.
- Especificação de Caso de Teste: define com um nível adequado de detalhe as informações necessárias no que se refere às entradas e saídas do software que está sendo testado. Isso inclui todos os casos de testes identificados.
- Especificação de Procedimento de Teste: descreve todas as restrições especiais, nível de teste que está relacionado a um caso de teste, pré-condições e os passos necessários para execução de um procedimento de teste.

A Figura 1 apresenta graficamente a evolução da fase de planejamento de teste sugerido pelo padrão IEEE-Std 829, incluindo as atividades de inspeção (cinza diamante) para avaliar os três artefatos de teste selecionados.

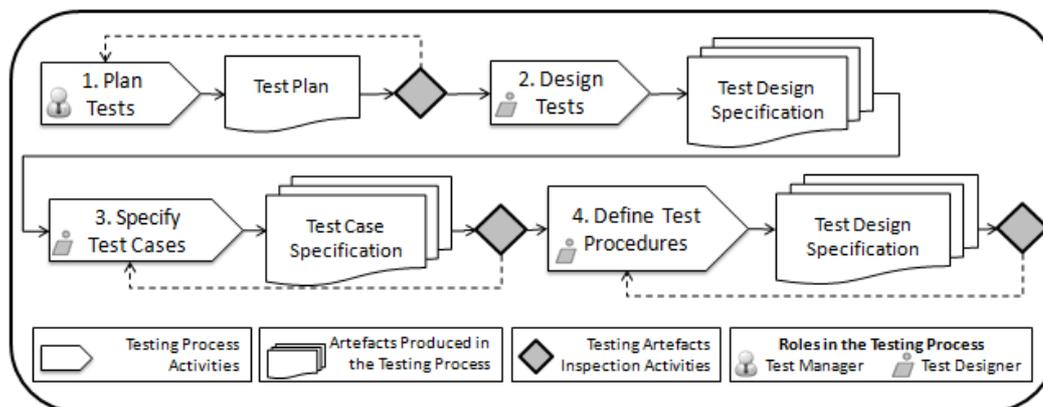


Figura 1. Evolução do Subprocesso de Planejamento dos Testes (BRITO e DIAS-NETO, 2012)

3.2. Descrição de uma Taxonomia de Defeitos

A inspeção busca detectar defeitos em artefatos de software. Existem diferentes tipos de defeitos e é importante que estes sejam classificados de forma a facilitar sua detecção. A taxonomia que foi adotada nesta pesquisa foi definida por SHULL (1998), composta pelos seguintes tipos: omissão, ambiguidade, inconsistência, fato incorreto e informação estranha (Tabela 1).

Tabela 1. Descrição dos tipos de defeitos

Tipo Defeito	Descrição
Omissão	1. Quando um elemento (ex: item de teste, tarefa, dado de entrada/saída, etc.) do planejamento de teste necessário para viabilizar sua execução não foi definido;
Ambiguidade	2. Quando a forma como os elementos do planejamento de teste ou suas responsabilidades foram definidos dificulta seu entendimento, prejudicando a execução dos testes (ex: Homônimos).
Inconsistência	3. Quando elementos descritos no planejamento dos testes possuem mesma descrição, mas nomes/identificadores distintos (Sinônimo); 4. Quando um elemento do planejamento dos testes presente em um artefato (caso ou procedimento de teste) não foi definido no plano de teste; 5. Quando a representação de um elemento não condiz com a semântica estabelecida pela abordagem de documentação dos testes.
Fato Incorreto	6. Quando um elemento não foi descrito ou representado de forma correta (ex: adoção de um tipo incorreto para uma entrada de dados). 7. Quando não é possível mapear um elemento do planejamento de teste de um artefato para outro (ex: do caso de teste para o procedimento de teste).
Informação Estranha	8. Quando não é possível determinar o papel de um elemento do planejamento de teste no atendimento aos requisitos especificados.

3.3. Definição dos Itens de Avaliação do Checklist

Para compor os *checklists* que formam *TestCheck*, vários itens de avaliação foram definidos e agrupados de acordo com o artefato de teste a ser inspecionado. No entanto,

esses itens foram detalhados em questões que visam atender a dois propósitos: (1) avaliar a consistência dos artefatos de teste entre si e (2) avaliar o atendimento dos testes aos requisitos, conforme a Figura 2.

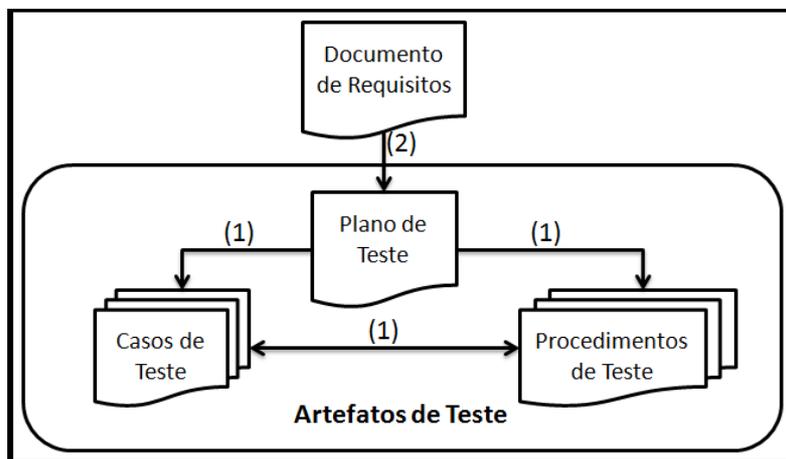


Figura 2. Perspectivas de avaliação dos itens que compõem *TestCheck* (BRITO et al., 2012)

Cada questão dos checklists propostos foi elaborada de forma que sua resposta positiva (SIM) representa a inexistência de defeito no artefato avaliado em relação ao conteúdo que ela se propõe a avaliar. Caso sua resposta seja negativa (NÃO), isso sugere que defeitos existem no artefato e precisam ser reportados ao moderador da inspeção. Além disso, os checklists foram definidos com um conjunto abrangente de itens de avaliação (questões) com o objetivo de permitir a adaptação às características dos artefatos de teste a serem avaliados em uma organização. Assim, foi definida uma opção “Não Aplicada” que pode ser respondida para uma questão caso aquele item não se adeque ao tipo de artefato adotado pela organização.

O processo de criação dos checklists da abordagem ocorreu artefato a artefato, iniciando-se pelo desenvolvimento de um checklist para o artefato Plano de Teste, em seguida passou-se para o artefato Caso de Teste e, por fim, foi definido o checklist para o artefato Procedimento de Teste. Definidos todos os checklists, deu-se início à fase de avaliação experimental de *TestCheck*.

4. Avaliação Experimental de *TestCheck*

Com o propósito de avaliar a abordagem, fez-se necessário a realização de estudos de avaliação visando caracterizar essa nova abordagem em relação à sua capacidade em identificar defeitos e visando permitir a transferência dessa tecnologia para o contexto industrial, reduzindo os riscos de implantação em um contexto industrial. Para isso, a metodologia definida por SHULL et al. (2001) foi utilizada. Ela é composta por quatro etapas, das quais duas foram aplicadas nesta pesquisa.

No primeiro momento foram realizados dois estudos de viabilidade que tiveram como objetivo caracterizar os itens de avaliação (checklist) que formam a abordagem *TestCheck* em relação a sua eficiência e eficácia para identificação de possíveis defeitos em artefatos de teste. Seus planejamentos, projetos e resultados estão descritos com

detalhes em (BRITO e DIAS-NETO, 2012). Em seguida, realizou-se um estudo de observação em ambiente industrial para avaliar o comportamento de *TestCheck* em projetos reais da indústria, conforme publicado em (BRITO e DIAS-NETO, 2013)

4.1. Primeiro Estudo de Viabilidade

Esse estudo objetivou a inspeção de documentos de teste reais através dos checklists que compõe a abordagem *TestCheck*. Com os resultados obtidos a partir da realização desse estudo é possível caracterizar a abordagem proposta em relação aos resultados obtidos com a aplicação de uma abordagem *Ad Hoc*. Entretanto, a comparação entre as duas abordagens não foi o objetivo principal, ou seja, não se pretendia identificar qual a melhor abordagem de avaliação através dessa comparação. Até porque intuitivamente uma abordagem que auxilie o inspetor em como procurar e onde procurar um defeito terá melhores resultados quando comparada a uma avaliação *Ad Hoc*. Tendo em vista as características da abordagem *Ad Hoc*, um possível questionamento é o porquê de utilizá-la para comparar com a abordagem proposta. Esta abordagem *Ad Hoc* foi aplicada por não ter sido identificado na literatura técnica outra abordagem similar à *TestCheck* que pudesse ser aplicada nos estudos.

Portanto, o objetivo dessa avaliação é de caracterizar *TestCheck* em relação a um controle ou *baseline*. Para esse estudo, o *baseline* escolhido foi uma abordagem *Ad Hoc*, pois é uma abordagem que tem sido utilizada em contexto industrial.

- **Participantes**

Manifestarem interesse em participar do estudo 14 alunos de graduação e pós-graduação matriculados na disciplina IEC 485 – Qualidade de Software nos cursos de graduação e pós-graduação do Instituto de Computação (IComp) da UFAM no período 2011/02. Entretanto, para participar do estudo, os estudantes tiveram que participar de um treinamento com uma explicação dos checklists que compõem a abordagem *TestCheck*, além de terem cursado alguma disciplina que abordou o tema teste de software ou terem participado de projetos reais atuando em tarefas relacionadas a teste de software.

- **Critério Para Seleção dos Grupos**

Para seleção dos grupos, cada aluno (inspetor) foi analisado e avaliado individualmente, por meio de um questionário de caracterização que os mesmos preencheram com informações que permitiram identificar o grau de conhecimento e experiência. Para este estudo, os mesmos foram agrupados em grupos (A e B). Para compor as equipes, certas precauções foram tomadas visando reduzir eventuais fatores de confusão, por conta do nível de qualificação dos participantes. Para isso, os grupos foram mesclados de acordo com as informações contidas no formulário de caracterização.

- **Artefatos Inspeccionados**

Foram inspeccionados dois projetos contendo os artefatos de teste (chamados projeto *Industrial* e *Acadêmico*). Os artefatos do projeto *Industrial* foram provenientes de um projeto real desenvolvido na indústria para um sistema de informação cujo objetivo de gerenciar usuários, seus perfis e funcionalidades em um software web para gestão de projetos. Os artefatos de teste do projeto *Acadêmico* foram provenientes de um trabalho de uma disciplina de teste de software ministrada em 2010 onde alunos do IComp/AM (que não são os participantes do estudo) projetaram testes para um software de controle

de uma biblioteca. Assim, os projetos entregues as equipes são reais e originais, não tendo sido inserido qualquer tipo de defeito pelos experimentalistas.

- **Execução do Estudo**

O estudo consistiu na simulação de uma sessão de inspeção de software. Os participantes em um primeiro momento aplicaram a técnica *Ad Hoc* em artefatos de teste de software. Cada participante recebeu individualmente um projeto (*Industrial* ou *Acadêmico*) contendo um plano de teste, cinco casos de teste, cinco procedimentos de teste. Para auxiliar e tirar qualquer dúvida, foi também entregue um documento de requisitos, não sendo necessário inspecioná-lo. Neste momento, os membros do grupo A receberam os artefatos do projeto Industrial e os membros do grupo B receberam os artefatos do projeto Acadêmico. Após a inspeção, os participantes entregaram as listas de discrepância, ficando assim aptos a receber o treinamento para segunda etapa do projeto. Neste primeiro momento houve duas desistências, e assim deu-se prosseguimento ao estudo, com apenas 12 participantes (7 do grupo A e 5 do grupo B).

- **Resultado do Primeiro Estudo de Viabilidade**

Após as listas de discrepâncias terem sido integradas a uma única lista, e ser feita a classificação entre defeitos e falso-positivos, foi realizada a análise dos dados. O cálculo realizado para encontrar a eficácia é dado por *Defeitos/Discrepância* (BARCELOS, 2006) e eficiência pela fórmula *Defeitos/Tempo* (KOLLANUS et al., 2009; BARCELOS et al., 2006). Discrepâncias equivalentes apontadas por mais de um inspetor eram diretamente classificadas como defeitos e apenas um registro era mantido. As discrepâncias restantes foram analisadas e avaliadas, sendo classificadas como defeitos ou falso-positivos. Para esta análise, houve a participação do pesquisador envolvido e um especialista em teste de software para definição da lista final de defeitos.

A primeira análise realizada neste estudo foi a análise das médias obtidas. A Tabela 2 apresenta os resultados deste estudo, com os valores totais obtidos. Analisando a tabela, é possível constatar que *TestCheck* obteve em média melhores resultados quando comparado a técnica *Ad Hoc*.

Tabela 2. Análise da Média do Estudo de Viabilidade 1.

Variável	Técnicas de Inspeção: Valor (Média)		Projetos de Software: Valor (Média)	
	<i>Ad Hoc</i>	<i>TestCheck</i>	Industrial	Acadêmico
Eficácia	0,54 def/disc	0,77 def/disc	0,75 def/disc	0,65 def/disc
Eficiência	0,08 def/min	0,24 def/min	0,18 def/min	0,16 def/min

Após os resultados preliminares foi necessário aplicar teste estatístico, para avaliar a existência de diferenças estatísticas entre as técnicas de inspeção (Tabela 3).

Analisando a tabela é possível observar que existe diferença estatística entre as técnicas para as variáveis *Defeitos*, *Eficiência* e *Eficácia*, pois o valor do *p-value* ficou abaixo de 0,05. É possível observar também que não existe diferença estatística para as variáveis *Tempo* e *Falso Positivo*. No entanto, este resultado foi considerado positivo e já era esperado no que diz respeito à avaliação da viabilidade da técnica *TestCheck*, já

que esta técnica introduz novos elementos (checklists) que poderiam se transformar em fatores que fazem a inspeção dos artefatos de teste de software mais difíceis. Analisando os projetos, observou-se que estes não influenciaram nos resultados deste estudo, ficando comprovado através do *p-value* que ficou acima de 0,05. Dessa forma, observou-se que a técnica *TestCheck* foi, neste estudo, o fator que influenciou estatisticamente na detecção de defeitos nos artefatos de teste de software.

Tabela 3. Análise de Variância - ANOVA (BRITO e DIAS-NETO, 2012)

Variável	Análise das Técnicas		Análise dos Projetos	
	<i>p-value</i>	Diferença Estatística?	<i>p-value</i>	Diferença Estatística?
Defeitos	0,0003	SIM	0,9555	Não
Falso Positivo	0,3078	NAO	0,0539	Não
Tempo	0,2697	NAO	0,6699	Não
Eficiência	0,0120	SIM	0,3951	Não
Eficácia	0,0101	SIM	0,0616	Não

- **Evolução da Abordagem para o Segundo Estudo de Viabilidade**

Apesar do bom desempenho obtido por *TestCheck*, entendeu-se que a técnica está em processo de amadurecimento. Então, foram analisados e avaliados pontos de possíveis melhorias. A quantidade de falso-positivos gerados a partir da aplicação de *TestCheck* chamou atenção, sendo um possível candidato à melhoria. Foi obtida uma taxa de 11,7 falso-positivos por participante, considerada alta, pois a taxa de defeito foi de 39,16 por participante. Com isso, aproximadamente, a cada 3 defeitos, 1 falso-positivo foi gerado por *TestCheck*. Portanto, o objetivo principal da evolução de *TestCheck* em um primeiro momento foi reduzir a taxa de falso-positivos encontrados no primeiro estudo realizado e manter/aumentar a taxa de defeitos detectados, quando comparada à técnica *Ad Hoc*.

Assim, foi realizada uma análise individual para cada questão que compõe os checklists que formam *TestCheck*. Todas as questões dos 3 checklists que tiveram uma taxa de defeitos por falso-positivo maior ou igual a 3,0 (valor este obtido no cenário geral, conforme citado anteriormente), seriam potenciais candidatas a serem evoluídas para a próxima versão da técnica. Os itens que não geraram nenhum falso positivo não foram incluídos na análise.

Ao total, 31 questões (de um total de 75 questões que compõem os 3 checklists) foram selecionadas como candidatas à evolução e foram revisadas e algumas reescritas para a criação de uma nova versão de *TestCheck*. Ao criar os questionamentos, buscou-se que estes fossem coerentes, com termos adequados ao contexto, e simples, para impossibilitar várias interpretações. Para uma melhor avaliação, pessoas alheias à pesquisa manifestaram opinião em relação ao entendimento, clareza, simplicidades das questões após sua leitura.

4.2. Segundo Estudo de Viabilidade

Com o objetivo de avaliar a evolução de *TestCheck* a partir da nova versão produzida após o primeiro estudo de viabilidade, fez-se necessário avaliar os resultados obtidos em um segundo estudo experimental (segundo estudo de viabilidade) para analisar se as mudanças implementadas refletem em melhorias para a técnica proposta.

- **Planejamento do Segundo Estudo de Viabilidade**

O segundo estudo de viabilidade foi conduzido seguindo todos passos realizados no primeiro estudo. Ao total, 19 alunos manifestaram interesse em participar. Estes foram divididos em dois grupos com (A) 9 e (B) 10 alunos. Foram utilizados os mesmos projetos do Estudo 1. Após a execução da atividade de inspeção, os participantes entregaram as listas de discrepância, sendo que houve 9 desistências, ficando apenas 6 pessoas no grupo A e 4 no grupo B aptos a receberem o treinamento para utilização da segunda abordagem, *TestCheck*, em artefatos de teste.

- **Resultado do Segundo Estudo de Viabilidade**

Para avaliação da técnica *TestCheck*, foram coletados os dados para as mesmas variáveis analisadas no estudo 1. Como o foco principal deste segundo estudo foi avaliar a evolução de *TestCheck* em relação à quantidade de falso-positivos encontrados por inspetor e a manutenção da taxa de detecção de defeitos, primeiramente este aspecto será avaliado.

A Tabela 4 apresenta um comparativo entre as técnicas de inspeção focando na quantidade de defeitos detectados e falso-positivos gerados por participante. A taxa de falso-positivos no primeiro estudo foi de 3:1 (três defeitos por falso positivo). No segundo estudo, essa taxa foi de 7:1, ou seja, a cada sete defeitos encontrados um falso positivo era gerado, o que permite constatar que para este estudo a evolução da abordagem conseguiu alcançar o objetivo em relação à redução da taxa de falso positivo, sem, no entanto, comprometer o segundo objetivo do estudo que foi manter a taxa de detecção de defeitos de *TestCheck* alta quando comparada à técnica *Ad Hoc*.

Tabela 4. Resultados comparativos das técnicas.

Medidas	Ad Hoc		TestCheck	
	Falso-Positivos	Defeitos	Falso-Positivos	Defeitos
Total	29	119	49	350
Média	2,9	11,9	4,9	35,0
Desvio Padrão	2,846	10,969	3,314	22,852

Sabe-se que não se pode extinguir a presença de falso-positivos, pois este é um elemento que faz parte do processo de inspeção e depende de diversos fatores que não necessariamente podem ser controlados. No entanto, estes podem ser otimizados reduzindo alguns destes fatores que levam à ocorrência de falso-positivos, principalmente em relação à redação das questões que formam os checklists, eliminando ambiguidades que podem levar a múltiplas interpretações das questões apresentadas.

Em seguida, foi realizada a análise das médias obtidas, e os resultados estão representados na Tabela 5. Analisando esta tabela, é possível constatar que *TestCheck* obteve em média melhores resultados quando comparado a técnica *Ad Hoc*.

Tabela 5. Análise das Médias.

Variável	Técnicas de Inspeção: Valor (Média)		Projetos de Software: Valor (Média)	
	Ad Hoc	TestCheck	Industrial	Acadêmico
Eficácia	0,80 def/disc	0,88 def/disc	0,86 def/disc	0,85 def/disc
Eficiência	0,09 def/min	0,14 def/min	0,15 def/min	0,11 def/min

Após esta análise, novamente foi necessário aplicar teste estatístico para avaliar a existência de diferenças estatísticas, conforme descrito na Tabela 6.

Tabela 6. Análise de Variância – ANOVA (BRITO e DIAS-NETO, 2012)

Variável	Análise das Técnicas		Análise dos Projetos	
	p-value	Diferença Estatística?	p-value	Diferença Estatística?
Defeitos	0,0099	SIM	0,6313	Não
Falso Positivo	0,1649	NAO	0,6845	Não
Tempo	0,0306	SIM	0,8122	Não
Eficiência	0,0857	SIM	0,9869	Não
Eficácia	0,0474	NAO	0,6419	Não

Analisando o resultado, é possível observar que existe diferença estatística entre as técnicas *Ad Hoc* e *TestCheck* para as variáveis *Defeitos*, *Tempo* e *Eficiência* (resultado similar ao primeiro estudo para *Defeito* e *Eficiência*). Analisando a variável *Tempo*, percebeu-se que a técnica *Ad Hoc* neste estudo demandou menos tempo para execução do processo de inspeção, o que já era esperado. Analisando o projeto de software, não foi encontrada diferença estatística para nenhuma das variáveis, sugerindo que a técnica seria o fator que influencia os resultados.

4.3. Estudo de Observação na Indústria

Seguindo a metodologia adotada, a segunda fase sugere a aplicação da nova tecnologia objetivando analisar seu amadurecimento. Tendo em vista que os resultados adquiridos no primeiro e segundo experimentos apresentaram indícios da viabilidade de *TestCheck*, esta fase foi realizada em um ambiente industrial localizado em um instituto de pesquisa da região norte. Esta fase tem como finalidade avaliar se a abordagem é adequada/viável para ser utilizada em um ambiente real. Foi analisada sua capacidade de detectar defeitos (avaliação quantitativa) e a satisfação por parte dos inspetores da indústria que utilizaram abordagem (avaliação qualitativa).

O propósito deste estudo é responder à questão: “A abordagem *TestCheck* é adequada no contexto industrial para avaliação de artefatos de teste?”. A partir deste estudo, foi observado como *TestCheck* se comporta dentro de um processo industrial sendo utilizada por profissionais em projetos reais.

Detalhes a respeito do planejamento, projeto e resultados deste estudo estão descritos em (BRITO e DIAS-NETO, 2013).

- **Planejamento do Estudo de Observação**

Os participantes são profissionais com diferente nível de experiência que atuam como membros da equipe de teste em projetos de software (2 Analista de Teste, 1 Estagiário e 1 Gerente de Teste). Os critérios para participação era: manifestar interesse em participar do estudo, assinando o Formulário de Consentimento e Formulário de Caracterização, para que os pesquisadores tivessem conhecimento do grau de experiência de cada participante, participar do treinamento que foi conduzido como uma explicação dos checklists que compõem a abordagem *TestCheck*; preencher o questionário “Pós-Experimento” e, por fim, serem profissionais que atuem na área.

Os artefatos de teste inspecionados (plano, casos e procedimentos de teste) consistem em documentos de um projeto real da própria organização e que já haviam sido avaliados através de uma abordagem própria (*Ad Hoc*), porém sem registro do seu desempenho (ex.: tempo e número de defeitos). Após todo procedimento da primeira etapa, os inspetores receberam um treinamento em relação à abordagem *TestCheck* e a maneira que deveriam proceder durante a atividade. Assim, deu-se início à execução da inspeção com o preenchimento da lista de discrepâncias.

Após a conclusão da inspeção, foi pedido a cada participante que preenchesse o Questionário de Avaliação Pós-Experimento, de forma a obter a sua avaliação qualitativa a respeito da aplicação da abordagem *TestCheck*.

De posse dos Relatórios de Discrepância gerados por cada participante, foi realizada uma única reunião de discriminação para classificação das discrepâncias em defeitos ou falso-positivo. Para qualificar uma discrepância como defeito ou falso positivo, contou-se com o apoio dos próprios participantes e de especialistas em teste de software (como ocorre em qualquer processo de inspeção tradicional).

• Resultado do Estudo de Observação

Analisando os resultados do estudo (Tabela 7), percebeu-se que o inspetor 04 não detectou nenhum defeito nos artefatos inspecionados, por[em a razão [e que este seria o autor do documento. Este foi incluído entre os inspetores com o objetivo de avaliar seu comportamento, analisando se os checklists ajudariam a identificar elementos importantes que poderiam conter nos artefatos. No entanto, este comportamento não foi constatado e o autor do documento acreditou que os artefatos estavam corretos de acordo com seu ponto de vista. Outro fator que pode explicar o baixo desempenho deste inspetor foi o tempo destinado à inspeção (10 minutos), conforme será ainda explicado nesta seção.

Apesar de o autor do documento não ter reportado qualquer discrepância durante a fase de detecção de defeitos, na fase de discriminação, onde as discrepâncias foram classificadas em defeitos ou falso positivo, este concordou com a classificação dos demais inspetores e especialistas em teste a respeito dos defeitos indicados.

Tabela 7. Resultado do estudo de observação realizado na Indústria.

Inspetor	Discrepâncias	Falso-Positivos	Defeitos	Tempo (minutos)	Eficácia (def / dis)	Eficiência
01	8	1	7	85	87%	0,08
02	8	0	8	90	100%	0,09
03	12	0	12	80	100%	0,15
04	Inspetor Removido					
Total	28	1	27	265	96%	0,10

Por se tratar de artefatos de um projeto real já inspecionado previamente, o número de defeitos detectados foi considerado satisfatório pelos próprios membros da organização. Todos os inspetores (com exceção do autor do documento) detectaram um grande volume de defeitos, considerando que o projeto já havia sido revisado pela própria empresa e já havia sido considerado apto a ser executado. Outro ponto positivo

foi a baixa quantidade de falso-positivos gerados pela utilização da abordagem proposta. Este resultado pode ser atribuído a vários fatores, inclusive a evolução dos checklists após os dois primeiros estudos. Outro fator a ser considerado é a experiência prévia de parte dos inspetores em inspeção de software, que pode ter contribuído para o sucesso da inspeção.

Analisando a utilização da abordagem *TestCheck* neste estudo, considerou-se que os resultados foram positivos. Além da alta taxa de detecção de defeitos, percebeu-se que os defeitos encontrados eram de alta severidade, relacionados a funcionalidades importantes do software, o que afetaria a qualidade dos testes. Em relação à eficiência e eficácia, constatou-se que a abordagem *TestCheck* resultou em 96% de eficácia (defeitos por discrepâncias) na detecção de defeitos e eficiência de 0,10 defeitos por minuto.

5. Conclusões

A crescente complexidade nos softwares desenvolvidos requerem testes de qualidade, e estes devem ser planejados de tal modo que satisfaçam seu objetivo principal que é revelar falhas. Durante o processo de testes, artefatos são desenvolvidos para em seguida serem executados. Portanto, é necessário que estes artefatos sejam de qualidade.

Nesta pesquisa, não foram encontradas abordagens que avaliassem os artefatos de testes. Assim, este trabalho propôs a abordagem *TestCheck* para apoiar a detecção de defeitos em artefatos de teste de software, visando melhorar a qualidade destes artefatos e como consequência aumentar a qualidade dos testes e do produto desenvolvido. Para analisar a viabilidade e amadurecimento da abordagem proposta, três experimentos foram realizados seguindo a metodologia proposta por SHULL et al. (2001). Os resultados indicam a viabilidade de *TestCheck* para detecção de defeitos em artefatos de teste, sendo esta eficaz e eficiente quando comparada a uma abordagem *Ad Hoc*. As principais contribuições oferecidas por esta pesquisa à comunidade de teste de software são:

- Desenvolvimento e disponibilização de uma abordagem baseada em checklist (*TestCheck*) para apoiar a detecção de defeitos em artefatos de teste de software, melhorando a qualidade dos testes.
- Mapeamento de características necessárias em artefatos de teste, que podem funcionar como diretrizes de apoio à elaboração (não apenas avaliação) destes artefatos em um projeto de software.
- Elaboração de um pacote de estudos visando a avaliação de técnicas de inspeção de artefatos de teste, desde sua concepção até o passo inicial visando sua transferência para indústria.

Assim como qualquer pesquisa científica, algumas limitações foram observadas no decorrer deste trabalho:

- Os estudos de avaliação da abordagem não são conclusivos, pois foram aplicados em contextos específicos que não permitem a expansão de seus resultados para outros cenários. Novos estudos em outros cenários com diferentes artefatos são necessários.

- Foram desenvolvidos checklists apenas para um subconjunto dos artefatos produzidos ao longo do processo de teste de software. Tal atividade poderia ser expandida para outros artefatos (ex: Laudo de Teste [Relatório de Resumo dos Testes]).
- Na forma atual, a customização dos checklists para aplicação em diferentes cenários é possível, porém custosa para ser realizada manualmente. Um mecanismo de apoio à customização da abordagem poderia ser empregado visando reduzir este esforço.

Apesar de não serem conclusivos, os dados obtidos nos estudos indicam que a abordagem é viável para o que se propôs realizar. Assim, como próximos passos estão previstos:

- A continuidade da metodologia proposta em (SHULL et al., 2001) que foi seguida para avaliação de *TestCheck*, visando a longo prazo sua transferência para ambiente industrial. Estes novos estudos contribuiriam para o amadurecimento da abordagem, introduzindo melhorias dos itens de avaliação e aprimoramento dos questionamentos.
- Implementar a abordagem *TestCheck* em um apoio ferramenta de forma a possibilitar a sua customização para projetos ou organizações específicas. Neste cenário, os profissionais do projeto poderiam escolher os itens que deveriam compor os checklists para ser utilizado. Este passo está em andamento, onde os checklists que compõem *TestCheck* estão sendo integrados à infraestrutura Maraká (DIAS-NETO, 2006), de apoio ao planejamento e controle de teste de software.

Referências Bibliográficas

- Barcelos, R., F., (2006), “Uma abordagem para inspeção de documentos arquiteturais baseada em Checklist”, Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro.
- Barcelos, R.F.; Travassos, G.H., (2006) “ArqCheck: Uma abordagem para inspeção de documentos arquiteturais baseada em checklist” In: Simpósio Brasileiro de Qualidade de Software (SBQS), PP. 174-188, Vila Velha-ES.
- Brito, J., Dias-Neto, A., C., (2012) “Conduzindo Estudos Experimentais para Avaliação de uma Técnica de Inspeção de Artefatos de Teste de Software” - Experimental Software Engineering Latin American Workshop (ESELAW)
- Brito, J., Dias-Neto, A., C., (2012) “TestCheck – Uma Abordagem Baseada em Checklist para Inspeccionar Artefatos de Teste de Software” – Simposio Brasileiro de Qualidade de Software (SBQS)
- Brito, J.; Dias-Neto, A. C. (2013), “Conducting Empirical Studies to Evaluate a Technique to Inspect Software Testing Artifacts”. CLEI Electronic Journal, v. 16, p. 9.
- Dias-Neto, A.C.; Travasso, G.H., “Maraká: Uma Infra-Estrutura Computacional para Apoiar o Planejamento e Controle de Teste de Software”, V Simpósio Brasileiro de Qualidade de Software, Vila Velha, 2006.

- Fagan, M. E. Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal. Riverton, NJ. V.15. n.3.p.182-211, 1976.
- Hedberg, H., sakka J., (2006), "Technical Reviews in Agile Development: Case Mobile-DTM" - International Conference on Quality Software (QSIC'06) IEEE - Department of Information Processing Science, University of Oulu, Finland.
- IEEE Standard 829-2008: Standard for Software and System Test Documentation (Revision of IEEE Std 829-1998)
- Itohen, J., V. Mantyla, M.V., and Lassenius, C., "Defect Detection Efficiency: Test Case Based vs. Exploratory Testing " Helsinki University of Technology, Software Business and Engineering Institute P.O. BOX 9210, FIN-02015 TKK, Finland, 2010.
- Kaner, C., Exploratory Testing, Florida Institute of Technology, Quality Assurance Institute Worldwide Annual Software Testing Conference Orlando, FL, November, 2006.
- Kollanus, S., Koskinen, J., (2009), "Survey of Software Inspection Research" - Department of Computer Science and Information Systems, University of Jyväskylä, Finland - The Open Software Engineering Journal, pp. 15-34.
- Laitenberger, O., DeBaud, J.M., (1997). Perspective-based Reading of Code documents at Robert Bosch GmbH. Information and Software Technology, 39:781-791.
- Lanubile, F., Mallardo, T., "Inspecting Automated Test Code: A Preliminary Study" - Department di Informatica, University of Bari, IEEE Standard 829-1998: Standard for Software Test Documentation, IEEE Press., 2007.
- Poon, L. P., Tse, T.H., Kuo, F.C., (2010) "Contributions of tester experience and a Checklist guideline to the identification of categories and choices for Software testing" In: Software Quality Journal
- Shull, F., Carver, J., Travassos, G. H., "An empirical methodology for introducing software processes." ACM SIGSOFT Software Engineering Notes, 2001.
- Shull, F., Rus, I., Basili, V., "How perspective based reading can improve requirements inspections", IEEE Computer, v. 33, n. 7, pp 73-79, 2000.