ArqCheck: Uma abordagem para inspeção de documentos arquiteturais baseada em *checklist*

Rafael Ferreira Barcelos **Guilherme Horta Travassos**

Universidade Federal do Rio de Janeiro – COPPE/Sistemas
Caixa Postal 68.511 – CEP 21.941-972 – Rio de Janeiro – RJ – Brasil

{barcelos, ght}@cos.ufrj.br

Abstract. *Software architecture, usually represented by an architectural document, is extremely important to stakeholders since it is used in several moments throughout the software development process. Therefore, due to it importance, the reviewing of architectural documents becomes a fundamental activity for the success of the software project and for the improvement of the software quality. The architectural evaluation approaches described on the technical literature have some limitations related to their use that make difficult to apply them in industrial environments. This work proposes ArqCheck, an approach to inspect architectural documents that aims at minimizing these limitations. Moreover, it describes some results from a feasibility study conducted to evaluate this approach.*

Resumo. *A arquitetura de um software, representada através do documento arquitetural, é de grande importância para os stakeholders por ser utilizada em diversos momentos no processo de desenvolvimento do software. Portanto, a sua revisão se torna uma atividade relevante para o sucesso do projeto e para a melhoria da qualidade do software. As abordagens de avaliação arquitetural existentes apresentam limitações que dificultam sua aplicação em um contexto industrial. Esse trabalho propõe ArqCheck, uma abordagem para inspeção de documentos arquiteturais, que objetiva minimizar essas limitações. Além disso, ele descreve alguns resultados de um estudo de viabilidade dessa abordagem.*

1. Introdução

Arquitetura de software está sendo cada vez mais utilizada em projetos de desenvolvimento de software para representar as soluções computacionais. Isso ocorre devido às facilidades que ela oferece como, por exemplo, a possibilidade em abstrair informações, o que diminui a complexidade e facilita o entendimento da solução.

Segundo Bass *et al.* [2003], arquitetura de software representa a estrutura, ou conjunto de estruturas, que compreende os elementos de software, suas propriedades externamente visíveis e seus relacionamentos. Essa estrutura é utilizada principalmente como ferramenta para comunicar a solução projetada entre os diversos stakeholders que participam do processo de desenvolvimento do software [Garlan 2000]. Para que essa comunicação seja possível, essa estrutura deve ser descrita através de uma representação, conhecida como documento arquitetural.

Esse documento é de grande importância para os stakeholders visto que ele contém informações que são utilizadas por diversos tipos de stakeholders e por ser utilizado em diversos momentos no processo de desenvolvimento do software. Devido à essa importância, a sua revisão se torna uma atividade relevante para o sucesso do projeto e para a melhoria da qualidade do software. Esta afirmação é fortalecida se for considerado que (1) a sua revisão impede a propagação de defeitos arquiteturais para os

demais artefatos, como diagramas de projeto e código fonte, e (2) o custo de correção desses defeitos é bem menor se for realizada durante os primeiros estágios do projeto [Boehm 1981; Boehm e Basili 2001]. Contudo, como avaliar se o documento arquitetural descreve uma arquitetura que atenda aos requisitos especificados?

Existem diversas abordagens de avaliação descritas na literatura que objetivam avaliar as representações arquiteturais e, por conseqüência, a arquitetura nelas representadas. Barcelos e Travassos [2006] apresentam os resultados obtidos por uma revisão sistemática realizada com o objetivo de identificar as principais abordagens de avaliação arquitetural existentes.

A partir dos dados obtidos por essa revisão, foi possível identificar e caracterizar 20 abordagens de avaliação. Uma análise dessas abordagens permitiu identificar algumas limitações que dificultam a sua aplicação em contexto industrial [Barcelos e Travassos 2006].

As abordagens identificadas possuem limitações principalmente devido ao fato de terem sido criadas e aplicadas em contextos com grande disponibilidade de recursos como, por exemplo, em contextos acadêmicos. Devido a isso, elas (a) dependem muito do conhecimento dos avaliadores, o que torna a avaliação muito subjetiva, (b) apresentam um elevado custo de aplicação por necessitarem de grande quantidade de pessoas ou da aplicação de complexas técnicas de avaliação, (c) não realizam uma avaliação completa da arquitetura pois avaliam somente um número restrito de características de qualidade e (d) podem ser aplicadas somente em contextos limitados por dependerem do uso de determinadas técnicas ou de abordagens de representação arquitetural específicas.

Portanto, com a expectativa de minimizar essas limitações e levando em consideração os resultados que se tem obtido com a aplicação de técnicas de inspeção na revisão de artefatos de software [Shull *et al.* 2000; Conradi *et al.* 2003], este artigo apresenta ArqCheck, uma abordagem para inspeção de documentos arquiteturais baseada em *checklist*.

O *checklist* que compõe ArqCheck foi criado com base no conhecimento normalmente utilizado durante as atividades de especificação arquitetural e com base em propriedades presentes na família de técnicas de leitura OORT´s (*Object Oriented Reading Techniques)* [Travassos *et al.* 2002]. Além disso, os autores avaliaram essa abordagem através de estudos experimentais, que indicaram a viabilidade em utilizá-la para a identificação de defeitos em documentos arquiteturais.

Visando descrever esta abordagem, este artigo foi dividido em três seções além desta. Na seção 2, a abordagem de inspeção proposta é descrita. Na seção 3, são descritos como a abordagem está sendo avaliada. E na seção 4, o artigo é concluído.

2. ArqCheck: Abordagem para inspeção de documentos arquiteturais

Inspeção de software é um método rigoroso e formal executado visando revisar artefatos de software. Essa prática tem se apresentado como uma abordagem eficiente para encontrar defeitos, reduzir o retrabalho e aumentar a qualidade dos produtos [Boehm e Basili 2001]. Por esse motivo, os autores acreditam que uma abordagem de inspeção permita a avaliação de documentos arquiteturais e possibilite identificar defeitos arquiteturais relacionados ao atendimento dos requisitos do software.

Para definir ArqCheck, os autores se basearam nas características técnicas, identificadas por Laitenberger e DeBaud [1998], que compõem uma abordagem de inspeção. Essas características levaram a execução das seguintes tarefas:

- Determinar o artefato de software que deve ser avaliado pela abordagem e o tipo de informação que o compõem;

- Identificar os perfis das pessoas que devem ser envolvidas nessa inspeção, assim como o papel que desempenham durante a avaliação;
- Definir os tipos de defeitos que a abordagem busca identificar;
- Definir a técnica de avaliação utilizada para detectar os defeitos do artefato, e ;
- Especificar o processo a ser executado para atingir o objetivo final da avaliação.

A seguir, visando caracterizar a abordagem proposta, os resultados obtidos com a realização dessas tarefas são descritos.

2.1 Artefato de software avaliado

ArqCheck objetiva a melhoria da qualidade da arquitetura de um software através da identificação de defeitos no seu documento arquitetural.

Visto que não existe uma padronização para se representar arquitetura de software, a abordagem proposta procura avaliar o documento independentemente da forma como as informações são representadas. Contudo, para que ela seja aplicada é necessário que o documento arquitetural contenha algumas informações, como as definidas pelas recomendações da IEEE 1471 [IEEE 2000], que abordam a descrição arquitetural de sistemas de software.

Sendo assim, um documento arquitetural deve atender aos seguintes requisitos para possa ser avaliado por ArqCheck:

- Identificar os elementos arquiteturais que compõem a solução a ser construída, assim como a forma que esses elementos estão organizados;
- Descrever o papel de cada elemento dentro da arquitetura;
- Identificar como cada requisito relevante a nível arquitetural está sendo atendido através da arquitetura documentada. Essa identificação pode ser feita principalmente através do rastreamento de que requisito está sendo atendido e quais requisitos justificam a criação de determinado elemento arquitetural, e;
- Representar o software através de diferentes perspectivas, como por exemplo, através do uso de visões arquiteturais.

2.2 Papéis e perfis da equipe de inspeção

Para a abordagem proposta, utilizamos os papéis presentes em um processo padrão de inspeção de software: moderador, autor do documento e inspetor [Kalinowski 2004].

Para a execução dessa abordagem, é necessário alocar um indivíduo no papel de moderador e outro no de autor do documento. Em relação aos inspetores, visto que a abordagem proposta avalia o documento arquitetural sob duas perspectivas, consistência de informações e corretude em relação ao atendimento dos requisitos, a alocação é feita através da divisão dos inspetores em grupos que avaliam uma dessas perspectivas. Contudo, nenhum estudo foi realizado visando identificar o número de inspetores que deve ser utilizado para que ArqCheck seja executada de forma eficiente.

2.3 Taxonomia de defeitos

Defeitos em artefatos de software podem ser classificados através de diversas taxonomias. Essas taxonomias auxiliam os inspetores na identificação e categorização dos defeitos encontrados durante a atividade de detecção.

A taxonomia que utilizamos como base é a definida por Shull [1998], devidamente interpretada para o contexto de Arquitetura de Software (Tabela 1).

Tabela 1 - Taxonomia de defeitos

Tipos de defeitos	Descrição
Omissão	1. Quando um elemento arquitetural necessário para o atendimento a um requisito

	não foi definido;
Ambigüidade	2. Quando a forma como os elementos arquiteturais ou suas responsabilidades foram definidos dificulta ou impossibilita o atendimento a um requisito de qualidade. 3. Quando elementos descritos em visões distintas possuem o mesmo nome, mas responsabilidades diferentes (Homônimo);
Inconsistência	4. Quando elementos descritos em visões distintas possuem mesma responsabilidade, mas nomes distintos (Sinônimo); 5. Quando um elemento arquitetural presente em diagramas das demais visões não foi definido no diagrama avaliado; 6. Quando a representação não condiz com a semântica estabelecida pela abordagem de documentação. 7. Quando um elemento arquitetural é definido com responsabilidades distintas em duas ou mais visões. 8. Quando um elemento é representado de maneira diferente em duas visões.
Fato Incorreto	9. Quando um elemento não foi descrito ou representado de forma correta 10. Quando não é possível mapear um elemento arquitetural para algum elemento descrito em outra visão.
Informação Estranha	11. Quando não é possível determinar o papel de um elemento arquitetural ou de uma de suas responsabilidades no atendimento aos requisitos especificados.

2.4 Técnica de inspeção utilizada

Entre as técnicas de inspeção existentes, *checklist* é uma das que pode ser utilizada para identificar defeitos em documentos arquiteturais. O uso de *checklist* é justificado por oferecer vantagens em relação às demais técnicas de inspeção existentes:

- Técnicas *ad-hoc* não oferecem apoio ou procedimento de execução formal e sistemática da inspeção. Além do mais, os resultados obtidos dependem da capacidade, competência e experiência do inspetor [Chen *et al.* 2002].
- Técnicas de leitura são procedimentos que visam guiar individualmente os inspetores no entendimento de um artefato de software e, por conseqüência, na identificação de discrepâncias [Shull *et al.* 2000]. Abordagens de avaliação baseadas nessa técnica [Shull *et al.* 2000; Conradi *et al.* 2003] são mais eficientes na detecção de defeitos quando comparadas a outras técnicas de inspeção, como *checklists*, por exemplo. Porém, para que seja utilizada, o artefato deve ser representado em uma forma específica e padronizada, característica que ainda não pode ser atingida em documentos arquiteturais devido à imaturidade da área.

Várias abordagens baseadas em *checklist* já foram definidas visando a identificação de defeitos em documentos arquiteturais. Contudo, os questionamentos feitos por esses *checklists* são específicos ao domínio do software avaliado [Hollocker 1990; Nasa 1993], o que dificulta a sua aplicação em um contexto diferente do qual foi projetado, ou então são muito específicos à solução [Erickson *et al.* 1993], o que requer que a cada avaliação um novo *checklist* seja criado.

ArqCheck objetiva detectar defeitos arquiteturais através de um *checklist* de avaliação. A hipótese definida pelos autores é que o uso de *checklist* como técnica de inspeção e o tipo de conhecimento utilizado para definir seus itens possibilitam identificar defeitos arquiteturais relacionados ao atendimento dos requisitos do software e permitem minimizar as limitações identificadas nas abordagens de avaliação existentes [Barcelos e Travassos 2005].

Para que o uso de *checklist* possibilitasse a minimização das limitações identificadas nas abordagens de avaliação arquitetural, algumas decisões foram tomadas durante a definição dos itens de avaliação que formam o *checklist* (Tabela 2).

Além disso, os autores utilizaram conceitos presentes na família de técnicas de leitura OORT´s [Travassos *et al.* 2002] para criar esse *checklist*. Essa família de técnicas de leitura foi escolhida devido às similaridades existentes entre arquiteturas de software e documentos de projeto orientado a objetos, artefato avaliado por OORT´s.

Tabela 2 - Decisões tomadas para minimizar as limitações identificadas

Limitação identificada	Decisões tomadas
Subjetividade na avaliação	Definir um *checklist* composto por itens de avaliação que orientam o inspetor sobre o que ser avaliado, não deixando essa decisão somente a cargo de sua experiência.
Custo de aplicação	Utilizar inspeção de software por não exigir a realização de tarefas complexas, como a definição de cenários, nem uma grande quantidade de especialistas em arquitetura para que a avaliação seja realizada.
Avaliação de múltiplas características de qualidade	Definir itens no *checklist* para avaliar as características de qualidade de acordo com os requisitos de qualidade especificados pelo cliente e que possuem relevância em um contexto arquitetural
Contexto de aplicação limitado	Definir um processo de configuração que permita adaptar o *checklist* às características do documento arquitetural a ser avaliado e aos objetivos da avaliação.

Para compor esse *checklist*, vários itens de avaliação foram definidos e agrupados de acordo com seus propósitos durante a detecção de defeitos (Figura 1). Com isso, formaram-se grupos compostos por: (1) itens que avaliam a consistência do documento, (2) itens que avaliam o atendimento aos requisitos e (3) itens que avaliam a abordagem utilizada para atender aos requisitos de qualidade.

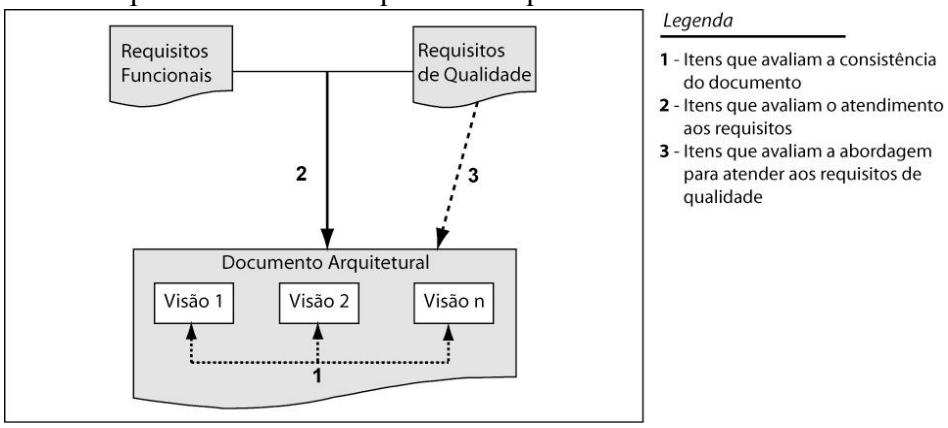


Figura 1 - Grupos de itens de avaliação que compõem o *checklist* proposto

2.4.1 Itens que avaliam a consistência do documento

O papel das visões arquiteturais é representar a arquitetura sob diferentes perspectivas, portanto, para avaliar se elas estão representando a mesma arquitetura, itens foram definidos para avaliar a consistência das informações de uma visão arquitetural e também a consistência entre diferentes visões (Tabela 3).

A avaliação realizada por esses itens consiste principalmente em analisar os diagramas gráficos que compõem o documento arquitetural. Visto que esses diagramas são específicos à abordagem de documentação utilizada, ou seja, dependem da semântica atribuída aos símbolos gráficos que descrevem a arquitetura, houve necessidade em se criar também itens de avaliação específicos à abordagem de representação gráfica utilizada.

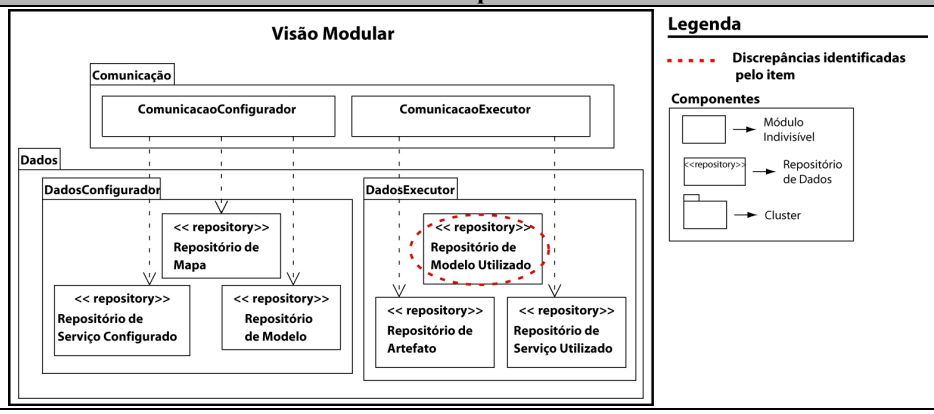
Os itens que compõem esse grupo foram definidos principalmente com base em conceitos relacionados à documentação de visões arquiteturais extraídos de [Clements *et al.* 2004]. Portanto, esses itens avaliam a consistência entre quatro tipos de visão: modular (2 itens), dinâmica (5 itens), de alocação (3 itens) e de contexto geral (2 itens). Além disso foram criados mais dois itens que avaliam conceitos presentes em todas as visões, totalizando assim 14 itens que compõem esse grupo.

A avaliação realizada por eles é de extrema relevância para a melhoria da qualidade do documento arquitetural pois buscam identificar defeitos que poderiam dificultar a implementação do software a partir da arquitetura. Além disso, eles

permitem uma avaliação direcionada, pois não fica a cargo do inspetor determinar o que deve ser avaliado.

Tabela 3 - Exemplo de item que avalia consistência do documento

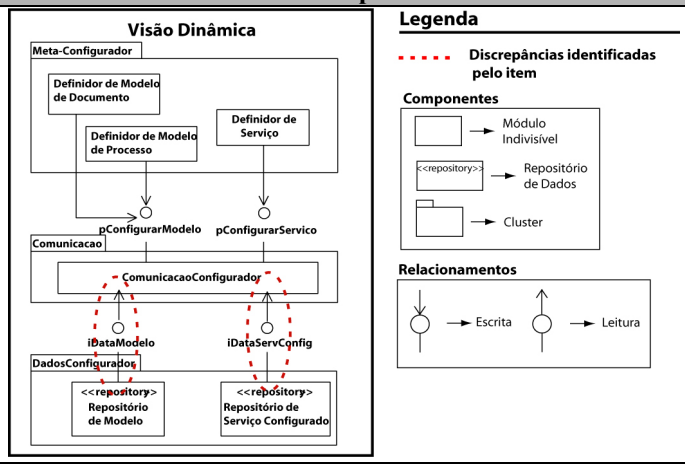
Item de avaliação de consistência	
Descrição do item	Ao analisar todos os diagramas, foi identificado algum elemento arquitetural que não possua relacionamentos, ficando isolado dos demais?
Resposta Esperada	Não
Tipo de Defeito	Informação Estranha ou Fato Incorreto
Objetivo	Identificar elementos inseridos que foram inseridos na arquitetura sem necessidade ou então identificar erros no projeto da arquitetura
Exemplo	



2.4.2 Itens que avaliam o atendimento aos requisitos

Tabela 4 - Exemplo de item que avalia o atendimento aos requisitos

Item de avaliação do atendimento aos requisitos	
Requisito Avaliado	A infra-estrutura deve permitir a persistência de todos os modelos e artefatos por ela manipulados
Descrição do Item	As responsabilidades dos elementos arquiteturais estão condizentes com os requisitos que eles atendem?
Resposta Esperada	Sim
Tipo de Defeito	Fato Incorreto
Observação	O diagrama abaixo está representando que os dados no repositório podem ser somente acessados e não alterados, o que vai de encontro com o requisito especificado.
Exemplo	

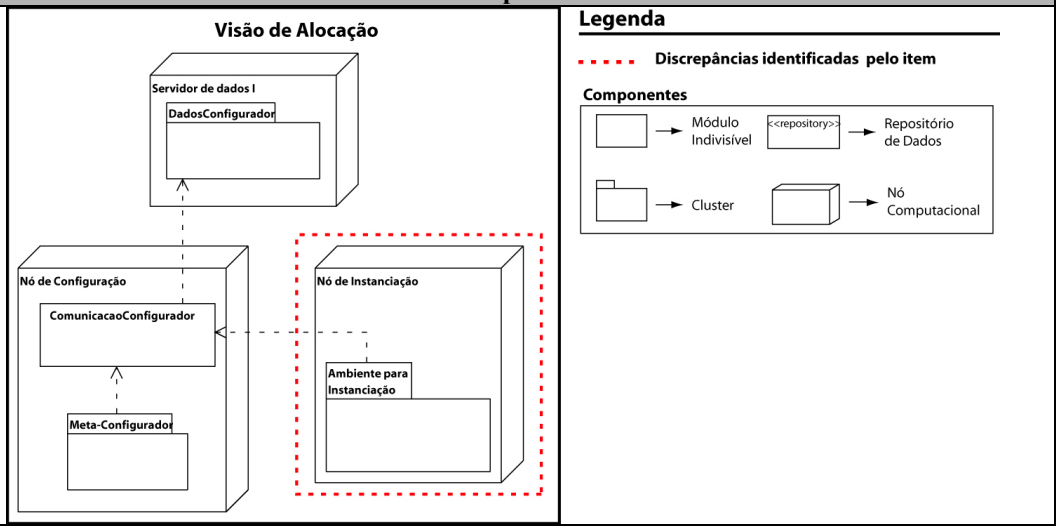


Os itens de avaliação que pertencem a esse grupo permitem analisar: se todas as funcionalidades especificadas foram atendidas pela arquitetura (2 itens) e se todos os elementos arquiteturais foram definidos com base nos requisitos especificados (4 itens).

2.4.3 Itens que avaliam a abordagem utilizada para atender aos requisitos de qualidade

Visto que avaliar o atendimento a requisitos de qualidade é de grande importância para a garantia da qualidade do documento arquitetural [Bass *et al.* 2003], itens de avaliação foram criados visando avaliar se a abordagem usada pelos arquitetos realmente atende aos requisitos de qualidade especificados (Tabela 5).

Tabela 5 - Exemplo de item que avalia a abordagem utilizada para atender a um requisito de qualidade

<table>
<tr><th colspan="2">Item de avaliação do atendimento aos requisitos</th></tr>
<tr><td>Requisito de desempenho avaliado</td><td>A infra-estrutura deve permitir a instanciação de um Ambiente para Execução em menos de 10 segundos</td></tr>
<tr><td>Cenário de qualidade</td><td>
<table>
<tr><th colspan="2">Cenário de Desempenho</th></tr>
<tr><td>Fonte do estímulo</td><td>Usuário Engenheiro</td></tr>
<tr><td>Estímulo</td><td>Instanciar Ambiente para Execução</td></tr>
<tr><td>Contexto do sistema</td><td>Durante a execução normal da infra-estrutura</td></tr>
<tr><td>Artefato</td><td>Sistema</td></tr>
<tr><td>Resposta</td><td>Ambiente para Execução instanciado</td></tr>
<tr><td>Medida de resposta</td><td>Menos de 10 segundos</td></tr>
</table>
</td></tr>
<tr><td>Guia de Identificação de Contexto</td><td>De acordo com os requisitos de desempenho selecionados, identifique as seqüências de execução do sistema (Seqüência), assim como os elementos arquiteturais (Elementos Participantes) que a compõem. Essa identificação pode ser feita através da análise do estímulo, do artefato e da resposta descritos nos cenários de desempenho.</td></tr>
<tr><td>Descrição do Item</td><td>Todos os Elementos Participantes foram alocados em um mesmo nó computacional visando aumentar o desempenho da Seqüência?</td></tr>
<tr><td>Resposta Esperada</td><td>Sim</td></tr>
<tr><td>Tipo de Defeito</td><td>Fato Incorreto</td></tr>
<tr><td>Observação</td><td>Para a execução da seqüência de instanciação, os elementos que participam dessa seqüência estão alocados em três nós computacionais diferentes, o que pode atrapalhar o atendimento ao requisito de desempenho especificado.</td></tr>
<tr><th colspan="2">Exemplo</th></tr>
<tr><td colspan="2"> 
</td></tr>
</table>

Para criar os itens que avaliam o atendimento a esses requisitos, informações provenientes de práticas usadas durante o projeto arquitetural foram utilizadas como base. A principal fonte de conhecimento utilizada na criação dos itens de avaliação do *checklist* proposto foram as táticas arquiteturais [Bass *et al.* 2003].

Táticas arquiteturais consistem em um conjunto de diretrizes arquiteturais baseados em boas práticas, que auxiliam no atendimento a requisitos de qualidade que influenciam o projeto de uma arquitetura [Tvedt *et al.* 2002].

A identificação dos tipos de requisito de qualidade que deveriam ser avaliados pelos itens do *checklist* foi obtida ao analisar a taxonomia de classificação de requisitos de qualidade descrita por Bass *et al.* [2003], que identifica os requisitos de qualidade relevantes a nível arquitetural. Com base nessa taxonomia, 22 itens de avaliação foram criados com o propósito de avaliar os requisitos de: Desempenho (5 itens), Disponibilidade (3 itens), Modificabilidade (5 itens), Segurança (5 itens), Testabilidade (2 itens) e Usabilidade (2 itens).

A principal contribuição desses itens e na forma como foram agrupados está na possibilidade de avaliar o atendimento a um ou mais requisitos de qualidade.

2.5 Processo de inspeção utilizado

Abordagens de inspeção são normalmente executadas a partir de um processo que identifica as atividades necessárias para detectar defeitos no artefato e que define os papéis que os indivíduos devem exercer durante a realização dessas atividades.

O processo utilizado por ArqCheck consiste em uma adaptação do processo tradicional de inspeção definido por Fagan [1976]. Devido às características do checklist proposto, as adaptações realizadas consistem principalmente na introdução de sub-atividades durante a etapa de planejamento e de detecção (Figura 2).

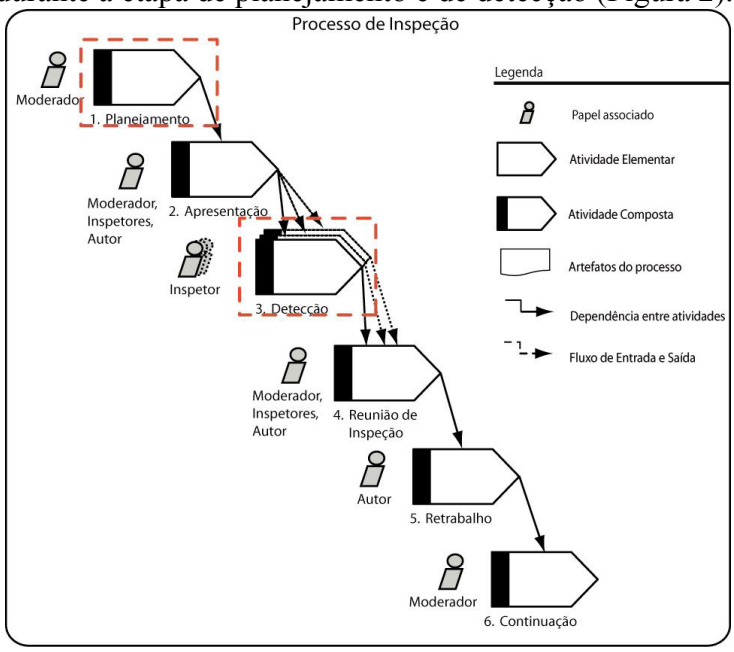


Figura 2- Atividades do processo de inspeção alteradas, representadas de acordo com a notação definida em [Villela 2004]

2.5.1 Planejamento

Nessa atividade, o moderador da inspeção define o contexto em que a inspeção vai ser realizada. Para isso, ele identifica os participantes, o artefato a ser avaliado e distribui o material aos inspetores. Contudo, devido à necessidade de configurar o *checklist* antes de realizar a avaliação do documento arquitetural, sub-atividades foram adicionadas para permitir a realização dessa tarefa (Figura 3).

A configuração do *checklist* consiste em selecionar os itens que são aplicáveis ao documento arquitetural a ser avaliado. Itens aplicáveis são itens específicos que são utilizados na inspeção de acordo com a abordagem de representação arquitetural utilizada e de acordo com os tipos de requisitos de qualidade que se deseja avaliar.

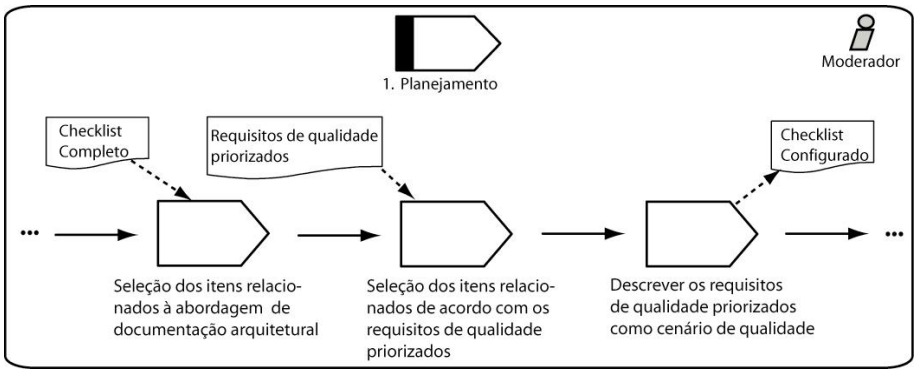


Figura 3 – Atividades de configuração

2.5.2 Detecção

Nessa atividade, os inspetores individualmente revisam o artefato identificando os defeitos. Devido às características do *checklist* que compõem a abordagem proposta, sub-atividades, que determinam a ordem de execução dos itens de avaliação, foram definidas visando auxiliar o inspetor na identificação de defeitos (Figura 4).

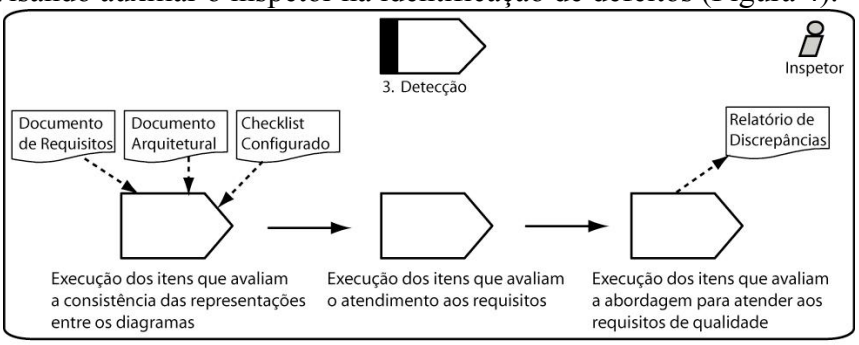


Figura 4 – Atividades de detecção

O motivo em estabelecer essa ordem está relacionado à ocorrência de inconsistências nas representações que podem influenciar na identificação de possíveis inconsistências relacionadas ao atendimento dos requisitos, por exemplo.

3. Avaliação da abordagem proposta

A abordagem proposta foi criada a partir de diversos tipos de conhecimento disponíveis na literatura técnica. Contudo, quando uma tecnologia é desenvolvida com base em experiências pessoais, observações gerais ou boas práticas publicadas, não se pode assegurar que ela obtenha os resultados esperados [Babar *et al.* 2004]. Portanto, os autores optaram por avaliar essa abordagem através de estudos experimentais.

Além do mais, o fato de resultados de pesquisas realizadas pelo MCT/SEPIN, no contexto das indústrias de software brasileiras, mostrarem um aumento pela procura por abordagens que permitam tanto a melhoria de seus processos quanto a de seus produtos [MCT/Sepin 2005], motivou os autores a, além de avaliar essa abordagem, viabilizar a sua transferência para as indústrias de software brasileiras. Para isso, foi utilizada uma metodologia descrita em [Shull *et al.* 2001] que visa o desenvolvimento de novas tecnologias e busca minimizar os riscos de introduzir essa tecnologia em um contexto industrial. Visando atingir esse objetivo, essa metodologia define uma série de estudos experimentais que devem ser realizados.

O nosso objetivo em utilizar essa metodologia é possibilitar a aplicação de ArqCheck em um contexto industrial. Contudo, a abordagem proposta encontra-se em fase de definição, tendo sido avaliada em somente um estudo experimental: o estudo de

viabilidade sugerido pela metodologia de Shull *et al.* [2001].

3.1 Estudo de viabilidade

3.1.1 Descrição

O principal objetivo desse estudo é identificar se ArqCheck realmente faz o que ela se propõe a fazer e se é viável continuar a despender recursos para desenvolvê-la. Com isso, procurou-se avaliar principalmente se o *checklist* que compõem a abordagem permite que um inspetor identifique defeitos em um documento arquitetural (Tabela 6).

Tabela 6 - Objetivo do estudo seguindo o método GQM [Basili *et al.* 1994]

Analisar o *checklist* para avaliar documentos arquiteturais
Com o propósito de caracterizar
Com respeito à clareza e aplicabilidade em identificar defeitos
Do ponto de vista de inspetores de artefatos de software
No contexto de uma inspeção de um documento arquitetural cujos defeitos são conhecidos

Para a realização desse estudo, foi utilizado o documento que representa a arquitetura do Ambiente de Experimentação eSEE [Neto *et al.* 2004], construído pelos membros do grupo de pesquisa ESE/COPPE/UFRJ.

Esse documento foi criado de acordo com as recomendações da IEEE 1471 e, portanto ele atende aos requisitos definidos como essenciais para que a abordagem proposta possa ser aplicada. Em relação a corretude desse documento, a versão utilizada possuía 18 defeitos conhecidos, que foram identificados anteriormente no processo de desenvolvimento de forma *ad-hoc*.

Além disso, visto que o objetivo do estudo é avaliar se os itens que compõem o *checklist* realmente detectam defeitos, defeitos adicionais foram inseridos no documento arquitetural de forma que possam ser encontrados pelos itens do *checklist*, para assim avaliar a clareza e a aplicabilidade desses itens em identificar defeitos.

A versão do *checklist* utilizada é composta por 42 itens de avaliação. Entre esses itens, existem os que são aplicáveis e os que não são aplicáveis. Os não aplicáveis são formados principalmente por itens que avaliam características do documento arquitetural que não estão sendo abordadas nesse estudo, como testabilidade por exemplo, e, portanto não deveriam fazer parte do *checklist*. Contudo, visto que um dos objetivos do estudo é avaliar a clareza dos itens, optamos por deixar esses itens.

Com isso, além das opções "Sim" e "Não", o *checklist* utilizado nesse estudo disponibiliza uma terceira opção, "NA - Não Aplicável" (Figura 5), que deve ser marcada caso o participante identifique que o item não pode ser usado para avaliar o documento arquitetural que lhe fora passado. Partimos do princípio que se o participante marcar um item aplicável como não aplicável ou vice-versa, isso poderia ser um indício de que ele não entendeu o propósito do item, o que categoriza o item como não claro.

Durante a preparação para a realização do estudo, o pesquisador aplicou ArqCheck no documento arquitetural a ser distribuído aos participantes. Essa tarefa foi realizada com o objetivo de caracterizar o *checklist* em relação aos defeitos do documento arquitetural avaliado. Essa caracterização permitiu, a partir do ponto de vista do pesquisador, (1) definir que itens do *checklist* levariam a identificar os defeitos, (2) definir se todos os defeitos conhecidos até o momento seriam identificados pelos itens, (4) avaliar se haveria novos defeitos que a inspeção *ad-hoc* não teria identificado e (5) identificar os itens que não seriam aplicáveis.

Itens de avaliação da consistência das representações entre os diagramas (específicos à abordagem de documentação arquitetural utilizada)				
Nº	**Visão Modular**	**Sim**	**Não**	**NA**
3	Os módulos internos de cada cluster foram descritos em algum diagrama da visão Modular?			
4	Todo relacionamento definido com um cluster foi devidamente mapeado para um de seus módulos internos?			
Nº	**Visão Dinâmica**	**Sim**	**Não**	**NA**
5	Toda porta/interface possui um nome, é utilizada com um único propósito e de forma única?			
6	Os fluxos de execução, descritos na visão Dinâmica, alocam todos os módulos definidos na visão Modular?			
7	Todo módulo/cluster representado na visão Dinâmica foi descrito na visão Modular?			
8	Todo fluxo entre dois elementos arquiteturais pode ser mapeado para algum relacionamento da visão Modular?			
9	Todo relacionamento, descrito na visão Modular, pode ser mapeado para algum fluxo de comunicação, de dados ou de controle da visão Dinâmica?			
Nº	**Visão de Alocação**	**Sim**	**Não**	**NA**
10	Todo módulo/cluster, representado na visão de Alocação, foi descrito na visão Modular?			
11	Toda dependência, representada na visão de Alocação, pode ser mapeada para um ou mais relacionamentos da visão Modular?			
12	Dado os módulos/clusters representados na visão de Alocação, todos os relacionamentos definidos entre eles na visão Modular também foram representados na visão de Alocação?			

Figura 5 – Parte do *checklist* utilizado durante o estudo de avaliação[1]

O estudo de viabilidade foi executado em outubro de 2005 com 4 participantes, escolhidos por conveniência. Estes participantes são alunos de pós-graduação e possuem conhecimento no domínio do problema e na aplicação de técnicas de inspeção.

Devido à falta de uma abordagem padrão para representação arquitetural, foi realizado, antes da execução do estudo, um treinamento com os participantes sobre a abordagem de representação utilizada para descrever o documento arquitetural. Esse treinamento foi realizado visando evitar problemas de entendimento em relação à simbologia utilizada, o que poderia influenciar na detecção de defeitos.

Por não estar no objetivo do estudo observar os participantes durante a execução da inspeção, os participantes realizaram o estudo em casa, mas foram informados que comentários entre eles deveriam ser evitados, para não prejudicar a validade do estudo.

Para cada participante, foram distribuídos o *checklist* de avaliação, o documento arquitetural, o documento de requisitos, o relatório de discrepâncias e o questionário de pós-experimento, utilizado para obter informações de ordem qualitativa.

3.1.2 Análise dos dados e Resultados Obtidos

Para avaliar ArqCheck em relação a sua viabilidade, os questionamentos definidos durante o planejamento do estudo foram respondidos com base nos dados coletados durante sua execução.

- ***Os itens de avaliação que compõem o checklist foram totalmente compreendidos?***

Em relação à compreensão dos itens, os pesquisadores partiram do princípio que se o participante não atribuísse o valor esperado ao item de avaliação, conforme definido durante a caracterização, isso poderia ser um indício de que ele não foi compreendido.

[1] Devido à limitação de espaço, não foi possível descrever o *checklist* de forma mais completa. Ele pode ser obtido ao entrar em contato com os autores.

Portanto, com base nos dados coletados, foi identificada para cada item a quantidade de participantes que atribuíram erroneamente valores para ele. Com isso, foi observado que 8 itens de avaliação não foram compreendidos por 3 ou mais participantes e por conseqüência precisam sofrer melhorias (Gráfico 1).

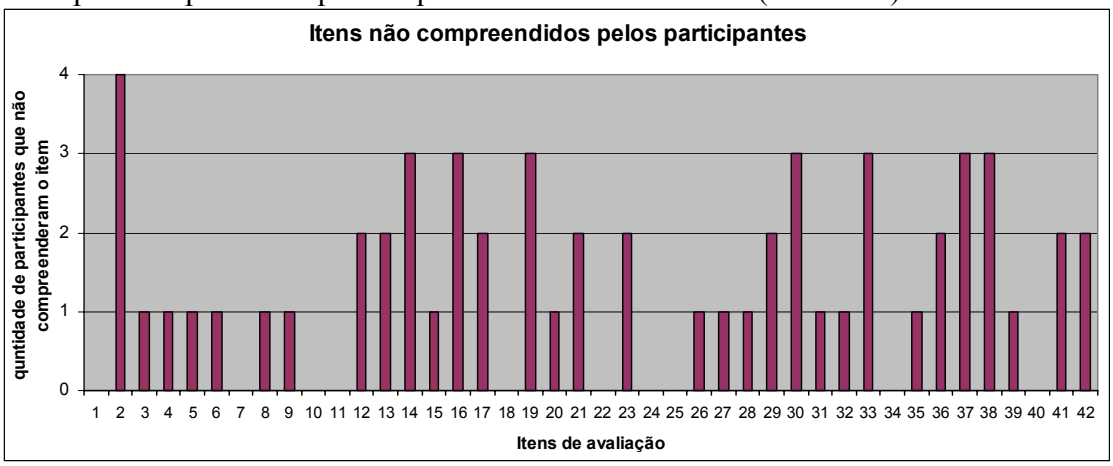


Gráfico 1 - Compreensão dos itens de avaliação

➢ ***Os itens de avaliação que compõem o checklist auxiliaram os inspetores na identificação de defeitos?***

Para identificar que itens auxiliaram na detecção de defeitos, foi realizada uma comparação entre os itens que efetivamente auxiliaram os participantes na detecção e os resultados obtidos durante caracterização realizada pelo pesquisador, possibilitando identificar os itens que deveriam detectar defeitos mas não o fizeram, e vice-versa.

Essa análise permitiu a identificação dos itens que o pesquisador tinha caracterizado como "identificadores de defeitos" mas não auxiliaram os participantes nessa tarefa (Gráfico 2). Com base nesses dados, esses itens foram analisados e constatou-se uma falta de clareza, o que os levou a serem evoluídos.

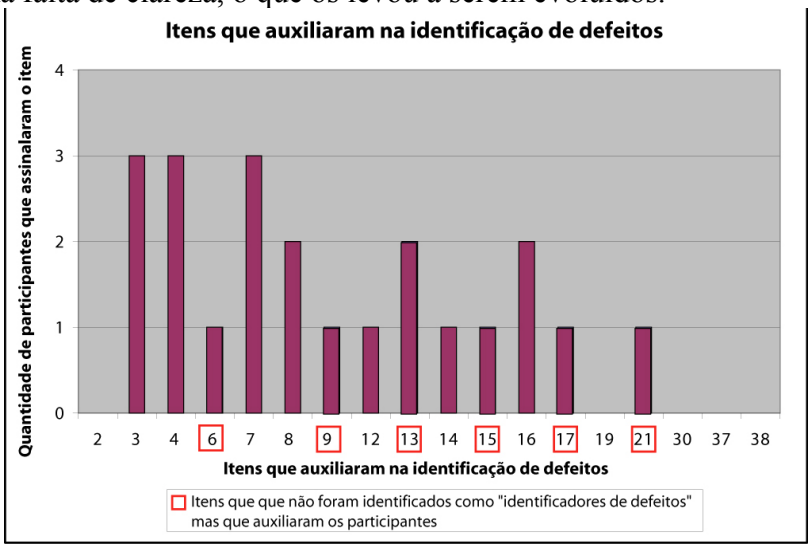


Gráfico 2 - Itens que auxiliaram na identificação de defeitos

Visto que não foi possível realizar agrupamentos entre os participantes em relação aos seus níveis de experiência, devido à pequena quantidade de participantes, a análise dos dados em relação a essa perspectiva apresenta fatores de confusão (*Counfounding factors*). Esses fatores ocorrem visto que não é possível determinar se os defeitos são identificados graças ao conhecimento contido no item de avaliação ou ao

conhecimento do inspetor obtido através de sua experiência. Contudo, mesmo com a presença desse fator, os indícios fornecidos em relação aos itens que auxiliaram na identificação de defeitos são relevantes para a evolução do *checklist*.

Além disso, foram identificados casos onde cada participante identificou determinado defeito através de itens diferentes que não tinham sido determinados como "identificadores de defeitos". Tal observação forneceu informações sobre a presença de itens redundantes, ou seja, itens diferentes, mas com o mesmo propósito. A principal conseqüência da presença desse tipo de item foi o elevado tempo de inspeção gasto pelos participantes (> 3 horas).

- ***Considerações em relação à hipótese nula do estudo***

Ao analisar os relatórios de discrepâncias devolvidos pelos participantes do estudo, foi identificado que em média cada participante identificou 30% dos defeitos. Ao analisar as porcentagens individuais dos defeitos inseridos e reais, a porcentagem de defeitos inseridos encontrados por participante é bem maior que a de defeitos reais (Gráfico 3).

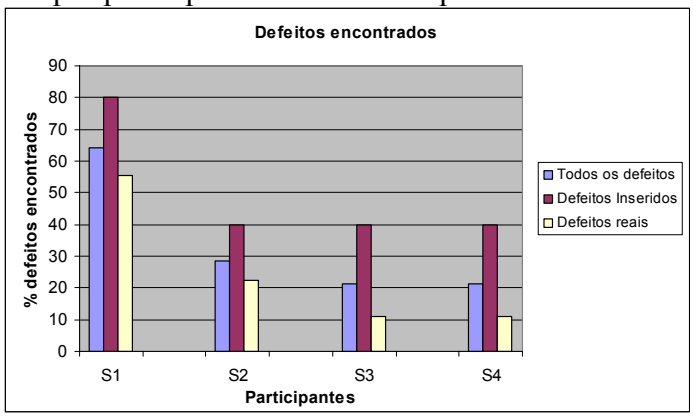


Gráfico 3 - Defeitos encontrados

Uma das razões dessa diferença de valores está no fato dos defeitos inseridos terem sido enxertados no documento com o objetivo de serem identificados, ou seja, os defeitos foram colocados de forma que haveria uma grande possibilidade dos defeitos serem identificados. Essa inserção foi realizada dessa forma com o objetivo de avaliar a clareza dos itens de avaliação.

Contudo, mesmo não sendo o objetivo desse estudo avaliar a quantidade de defeitos que o checklist auxilia a identificar, ao consolidar os relatórios de discrepâncias, identificou-se que 75% dos defeitos conhecidos foram encontrados por ArqCheck. A diferença entre a porcentagem de defeitos encontrados individualmente e a de defeitos encontrados no total, permite reforçar a idéia da alocação de vários inspetores para esse tipo de avaliação.

Esse estudo não permitiu refutar totalmente a hipótese de que a abordagem proposta não permite identificar defeitos arquiteturais, uma vez que não é possível obter resultados relevantes a nível estatístico, principalmente devido à pequena quantidade de participantes. Contudo, esses resultados preliminares fornecem alguns indícios, principalmente, se for levado em consideração que inspeções de software encontram, em média, 60% dos defeitos contidos no artefato avaliado [Boehm 1981].

3.1.3 Lições aprendidas

Com base na análise das discrepâncias identificadas e das respostas do questionário de pós-experimento, foi possível identificar:

- Os itens que necessitavam serem reescritos, por não auxiliarem na identificação de defeitos ou não estarem suficientemente compreensíveis;

- Os itens que poderiam ser excluídos do *checklist* por estar avaliando uma mesma característica várias vezes.

A partir do questionário de pós-experimento, os participantes opinaram em relação à relevância do *checklist* na identificação de defeitos arquiteturais. Como resultado, todos concordaram com o auxílio que esse *checklist* oferece para essa tarefa.

Além do mais, com base nessas informações foi identificada a necessidade de:

- Treinamento na utilização do *checklist*, principalmente relacionado ao uso das "Guias de identificação de contexto", utilizadas durante a avaliação do atendimento aos requisitos de qualidade;
- Descrição mais detalhada sobre a taxonomia de defeitos utilizada, fornecendo assim maior auxílio ao inspetor durante a classificação do tipo de defeitos identificado;
- Definição de um glossário visando auxiliar na compreensão de conceitos utilizados pelos itens de avaliação;
- Um mecanismo que permita informar ao inspetor que a atribuição de determinado valor para um item implica na presença de um defeito no documento avaliado. Para isso, foi adicionado ao *checklist* o campo "RE- Resultado esperado";
- Alocar mais de um inspetor para participar desse tipo de avaliação.

4. Conclusão e próximos passos

Diversos estudos experimentais já comprovaram os benefícios em revisar artefatos de software, principalmente quando essa revisão é realizada nas primeiras fases do processo de desenvolvimento de um software [Shull *et al.* 2000; Conradi *et al.* 2003].

Entre esses artefatos, a arquitetura de software se destaca por ser o primeiro a descrever a solução computacional que deverá ser implementada e por ser usada como ferramenta para comunicar essa solução aos diversos *stakeholders* que estarão envolvidos no desenvolvimento do software.

Este artigo apresenta ArqCheck, uma abordagem de inspeção para documentos arquiteturais. Essa abordagem consiste em utilizar um *checklist* que foi criado com base no conhecimento normalmente utilizado durante o projeto de arquiteturas para detectar defeitos. Além disso, essa abordagem foi criada visando minimizar as limitações identificadas nas abordagens de avaliação arquitetural existentes e que dificultam a aplicação dessa prática em um contexto industrial. Através desse estudo, foi descrito também um estudo de viabilidade que permitiu evoluir alguns itens de questionamento.

Atualmente, seguindo a metodologia de Shull *et al.*, o estudo de observação foi realizado. O principal objetivo desse estudo é avaliar a eficácia e o custo/eficiência de ArqCheck na inspeção de documentos arquiteturais. Uma das principais características desse estudo está no fato do documento arquitetural utilizado ter sido criado em um contexto industrial, obtido através de um acordo definido entre a COPPE/UFRJ e uma empresa de telecomunicações de grande porte. Visto que este estudo não faz parte do contexto desse artigo, informações sobre os resultados obtidos podem ser encontrados em [Barcelos 2006].

Agradecimentos

Os autores reconhecem o apoio da FAPEAM e do projeto CNPq – eSEE (4732135/2005-0) na realização deste trabalho.

Referências

Babar, M. A., L. Zhu and R. Jeffery (2004). A framework for classifying and comparing software architecture evaluation methods. Proceedings of the Australian Software Engineering Conference, Melbourne, Australia.

Barcelos, R. F. (2006). Uma abordagem para inspeção de documentos arquiteturais baseada em checklist. Dissertação de Mestrado. Rio de Janeiro, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ.

Barcelos, R. F. and G. H. Travassos (2005). Avaliando documentos arquiteturais através de um checklist baseado em atributos de qualidade. Proceedings of Workshop de Teses e Dissertação de Engenharia de Software (WTES) - SBES, Uberlândia, MG, Brasil.

Barcelos, R. F. and G. H. Travassos (2006). Evaluation Approaches for Software Architectural Documents: a Systematic Review. Proceedings of the Ibero-American Workshop on Requirements Engineering and Software Environments (IDEAS), Buenos Aires, Argentina.

Basili, V., G. Caldieira and H. Rombach (1994). Goal Question Metrics Paradigm. Encyclopedia of Software Engineering. J. J. Marciniak, Wiley. I.

Bass, L., P. Clements and R. Kazman (2003). Software Architecture in Practice, Second Edition, Addison Wesley.

Boehm, B. W. (1981). Software Engineering Economics, Prentice-Hall.

Boehm, B. W. and V. R. Basili (2001). "Software Defect Reduction Top 10 List." IEEE Computer 34(1): 135-137.

Chen, T. Y., P. L. Poon and S. F. Tang (2002). Towards a Problem-Driven Approach to Perspective-Based Reading. Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering (HASE'02), Washington, DC, USA, IEEE Computer Society.

Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford (2004). Documenting Software Architectures, Addison-Wesley.

Conradi, R., P. Mohagheghi and T. Arif (2003). Object-Oriented Reading Techniques for Inspection of UML Models - An Industrial Experiment. Proceedings of the European Conference on Object-Oriented Programming, Darmstadt, Germany.

Erickson, R. L., N. D. Griffeth, M. Y. Lai and S. Y. Wang (1993). Software architecture review for telecommunications software improvement. IEEE International Conference on Communications.

Fagan, M. E. (1976). "Design and code inspection to reduce Errors in Program Development." IBM Systems Journal 15(3): 182-211.

Garlan, D. (2000). Software architecture: a roadmap. Proceedings of The Conference on The Future of Software Engineering, ACM Press.

Hollocker, C. P. (1990). Software Reviews and Audits Handbook. New York, John Wiley \& Sons, Inc.

IEEE (2000). IEEE Recommended Practice For Architectural Description Of Software-Intensive Systems - IEEE Standard 1471-2000, Institute of Electrical and Electronics Engineers.

Kalinowski, M. (2004). Infra-Estrutura Computacional de Apoio ao Processo de Inspeção de Software. Dissertação de Mestrado. Rio de Janeiro, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ.

Laitenberger, O. and J. DeBaud (1998). Scenarios, Quality Attributes, and Patterns: Capturing and Using their Synergistic Relationships for Product Line Architectures. Kaiserslautern, Germany, Fraunhofer Institute Experimental Software Engineering.

MCT/Sepin, Qualidade e Produtividade no Setor de Software, 2005, "Qualidade e Produtividade no Setor de Software". In: http://www.mct.gov.br/sepin/Dsi/Software/Menu_Qualidade.htm accessed in Fevereiro, 2006.

Nasa (1993). Software Formal Inspections Guidebook, NASA Goddard Space Flight Center.

Neto, A. C. D., R. F. Barcelos, W. A. Chapetta, P. S. M. Santos, S. N. Mafra and G. H. Travassos (2004). Infrastructure for Software Engineering Experiments Definition and Planning. Proceedings of the Experimental Software Engineering Latin American Workshop, Brasilia.

Shull, F., J. Carver and G. H. Travassos (2001). An Empirical Methodology for Introducing Software Processes. Proceedings of European Software Engineering Conference.

Shull, F., I. Rus and V. Basili (2000). "How perspective-based reading can improve requirements inspections." IEEE Computer 33(7): 73-79.

Shull, F. J. (1998). Developing techniques for using software documents: a series of empirical studies. Tese de Doutorado, University of Mariland.

Travassos, G. H., F. Shull, J. Carver and V. R. Basili (2002). Reading Techniques for OO Design Inspections, Programa de Engenharia de Software - COPPE/UFRJ.

Tvedt, R. T., P. Costa and M. Lindvall (2002). Does the code match the design? A process for architecture evaluation. Proceedings of the International Conference on Software Maintenance.

Villela, K. (2004). Definição e construção de ambientes de software orientados à organização. Tese de Doutorado, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ.