

Aplicando Técnicas de Inspeção de Modelagem de Interações: Lições Aprendidas por Alunos de Graduação

Sebastião H. N. Santos¹, Eduardo S. Vieira¹, Arthur C. Serra¹, Luís Rivero¹

¹Departamento de Informática – Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brazil

{sebastiaohns, edusvieirap, arthurserra10, luisriverocabrejoso}@gmail.com

Abstract. *Inspection techniques are extremely important in the software development process, saving the resources of the team due to the ease of finding errors before the subsequent stages of production. In this work the MoLVERIC Check inspection technique was used to inspect the elements of the MoLIC interaction diagrams. The inspected artifacts were made from the specification of the Easy Coordination system, an academic process management system. The application of the technique was efficient in the identification and correction of the interaction diagrams, however, it was identified the need for some corrections in the items.*

Resumo. *Técnicas de Inspeção são extremamente importantes no processo de desenvolvimento de software, economizando os recursos da equipe, devido à facilidade de encontrar erros antes das fases subsequentes de produção. Neste trabalho foi usada a técnica de inspeção MoLVERIC Check, usada para inspecionar os elementos dos diagramas de interação MoLIC. Os artefatos inspecionados foram feitos a partir da especificação do sistema Coordenação Fácil, um sistema de gerenciamento de processos acadêmicos. A aplicação da técnica se mostrou eficiente na identificação e correção dos diagramas de interação, porém foi identificada a necessidade de algumas correções nos itens.*

1. Introdução

O processo de desenvolvimento de software consiste em diversas etapas bem definidas em que, dada uma entrada, é feita uma série de manipulações para resultar em uma saída esperada. Ao executar o processo em um ambiente real, é necessário descrever detalhadamente todos os passos para a obtenção do produto final. O sucesso do gerenciamento do processo requer um extenso planejamento, métricas de avaliação bem definidas e um rigoroso controle dos passos feitos.

Inspeção é um método formal, eficiente e econômico para encontrar erros, planejar e construir uma aplicação. Em [Fagan 1976] é definido o processo de inspeção de software como o inter-relacionamento de cinco elementos: seis passos de inspeção (planejamento, visão geral, preparação, encontro, retrabalho e acompanhamento), quatro papéis bem definidos de inspetor (moderador, gravador, leitor e desenvolvedor), uma definição formal do processo e dados do produto, o produto sendo inspecionado e uma infraestrutura de suporte [Ackerman et al. 1989].

As técnicas de inspeção de software se mostram eficientes para detecção de defeitos no processo de desenvolvimento e principalmente no produto, identificando as falhas tão logo elas são introduzidas, o que leva a uma economia substancial dos recursos

disponíveis, já que um defeito se torna cada vez mais custoso à medida que o desenvolvimento progride. Essa prática ajuda uma equipe de desenvolvimento de software a se tornar mais eficaz e eficiente, pois ela inter-relaciona passos do desenvolvimento mais comumente usados, os testes e a verificação formal do software em técnicas completas [Qazi et al. 2016].

Neste trabalho é descrito a execução de uma técnica de inspeção no processo de desenvolvimento de um sistema, a partir da modelagem do sistema de gerenciamento de processos acadêmicos, Coordenação Fácil, como base para a elaboração dos artefatos a serem inspecionados. O artefato foi construído seguindo as especificações do sistema e a definição da *Modeling Language for Interaction as Conversation* (MoLIC), a partir de então esse artefato foi inspecionado seguindo a técnica de inspeção MoLVERIC Check (MCheck) e, por fim, foram feitas as correções necessárias a partir dos erros encontrados. Na seção 2 é apresentada a fundamentação teórica do trabalho - definição da MoLIC e da MoLVERIC Check, na seção 3 é apresentado o domínio do sistema, o planejamento e os passos da inspeção, na seção 4 são apresentados os resultados da inspeção, na seção 5 o trabalho é concluído apresentando os resultados do processo apresentado.

2. Fundamentação Teórica

Com a popularização de computadores pessoais e dispositivos móveis foi levada aos desenvolvedores de *software* a demanda por produtos que atendam as necessidades dos usuários, as quais tem se tornado mais íntimas e únicas. A qualidade buscada passou do relacionamento do usuário com a arquitetura para sua experiência pessoal com o sistema usado. Para entregar a qualidade buscada pelo usuário, a área da Interface Humano-Computador (IHC) investiga formas de construir sistemas interativos com alta qualidade de uso, considerando as necessidades e preferências do usuário. O uso de modelos, sistema de tarefas e cenários são comuns para basear o processo de análise e desenvolvimento, representando comportamentos, reações, objetivos dos usuários ao usarem os sistemas e ajudando os designers a tomarem as decisões adequadas para atingir maior grau de aceitação e qualidade.

MoLIC (*Modeling Language for Interaction as Conversation*) é uma linguagem desenvolvida por Paula et al. (2005). Criada para representar a interação, ou conversa, que o usuário tem com o sistema para atingir seus objetivos. Ela é composta de quatro artefatos: diagrama de objetivos, esquema de sinais conceituais, diagrama de interação e a especificação de sinais situacionais [de Paula et al. 2005]. De forma simplificada, o diagrama de objetivos indica o que o usuário deve fazer com a aplicação, o esquema de sinais conceituais define e organiza os conceitos envolvidos no sistema, o diagrama de interação representa como os objetivos serão alcançados durante a interação e a especificação de sinais situacionais apresenta detalhes dos sinais usados no diagrama de interação. O principal artefato da linguagem MoLIC é o diagrama de interação, onde é diretamente representada a comunicação do usuário com o sistema, ou preposto do designer.

Um estudo de caso avaliativo identificou defeitos no diagrama de interação MoLIC [Lopes et al. 2015], levando a necessidade de desenvolver uma técnica específica para inspecioná-lo. Assim foi desenvolvida a MoLVERIC Cards (MCards), a qual usa elementos de *gamificação* [Camilo 2018] para encontrar defeitos em diagramas MoLIC e prevenir que eles se propaguem. Após um estudo de viabilidade da MCards [Damian 2016] fo-

ram encontradas defeitos que a técnica não abrangia, logo foi criada uma outra técnica de inspeção, agora baseada na MCards, a MoLVERIC Check (MCheck). É uma técnica de inspeção baseada em *checklist*, cujo os itens de verificação foram baseados no MCards com a adição de mais alguns itens. Os itens de verificação da MCards estão agrupados em dimensões que representam os elementos do diagrama MoLIC, nessas dimensões existem um ou mais itens de verificação para os tipos de defeitos definidos por [Travassos et al. 1999].

3. Aplicação da Técnica de Inspeção MoLVERIC Check

3.1. Definição do Sistema

A Coordenação de Ciência da Computação (COCOM) gerencia os pedidos de planos de estudos, aproveitamento de cadeiras e monitoria de maneira manual usando uma grande quantidade de material, que poderia ser substituída por um sistema informatizado. Por exemplo, caso um estudante queira dar entrada no pedido de aproveitamento de uma cadeira, ele precisa preencher o formulário correspondente no site da COCOM e imprimir histórico e ementa para cada disciplina cursada e entregar na secretaria da coordenação. Dessa forma, faz-se necessário o desenvolvimento de uma aplicação para tornar essas tarefas mais eficientes tanto para estudantes quanto para os coordenadores e professores.

Nesse sentido, o sistema denominado Coordenação Fácil propõe-se e responsabiliza-se pela submissão de planos de estudos para a coordenação, pedidos de aproveitamento de cadeiras e monitoria para professores para a coordenação de Ciência da Computação. No módulo de aproveitamento de cadeiras, o estudante anexará o histórico atualizado e as ementas das cadeiras cursadas, selecionará a cadeira cursada e a cadeira que pretende pedir aproveitamento e então submeterá o pedido para a coordenação para deferimento ou não. Após esse processo o estudante poderá acompanhar o status do pedido no painel inicial. De forma semelhante, no módulo de Planos de Estudos, o estudante anexará o histórico atualizado que será comparado com a grade curricular anteriormente cadastrada pelo coordenador. Então o sistema listará todas as cadeiras e períodos restantes para o estudante, o qual selecionará as cadeiras que deseja cursar em determinado período. Após esse processo o estudante poderá acompanhar o status do pedido no painel inicial.

3.2. Processo de Inspeção

Uma equipe de três integrantes desempenhou o processo de inspeção. Todos foram inspetores, responsáveis pela identificação dos erros no artefato seguindo os passos da técnica de inspeção escolhida. Um membro da equipe foi o moderador, responsável pela definição do contexto da inspeção, escolha da técnica de inspeção usada, treinamento dos membros da equipe, definição dos prazos de cada tarefa e a elaboração dos documentos resultantes da aplicação técnica.

Dentre os artefatos que a linguagem MoLIC gera foi escolhido o Diagrama de Interação para representar a conversa do usuário com o sistema Coordenação Fácil. Foram modeladas, por uma equipe paralela, as funcionalidades do módulo de aproveitamento de cadeira, em que um aluno pode submeter um pedido de aproveitamento, fazer a edição de um pedido submetido e visualizar todos os pedidos, essas funcionalidades principais do sistema são representadas no diagrama de interação como cenas, possuindo tópicos

e diálogos, em que o usuário e o preposto do designer irão se comunicar. Ainda há as funcionalidades básicas de acesso e encerramento, representados por elementos especiais da linguagem. O diagrama de interação de aproveitamento de cadeira desenvolvido é apresentado na figura 1. A técnica de inspeção MoLVERIC Check foi apontada como a mais adequada para a realização da inspeção, assim os inspetores receberam um treinamento sobre os itens de verificação de erros dos elementos do diagrama MoLIC presentes na técnica MCheck e como aplicá-la. Foram definidos os prazos para a realização das próximas etapas, encerrando a etapa de planejamento.

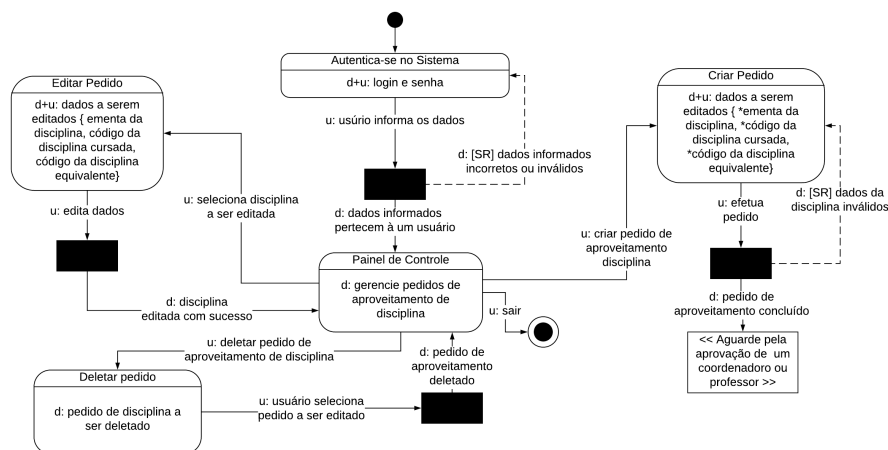


Figura 1. Artefato usado na inspeção - Diagrama de Interação Aproveitamento de Disciplina

Na etapa de detecção de defeitos os inspetores analisaram o artefato seguindo cada um dos itens de verificação da MCheck, ao término da análise de todos os elementos do diagrama de interação, foi feito, individualmente, pelos inspetores uma tabela com as informações de código do item de verificação que levou a identificação do erro, tipo de erro encontrado, referência dos elementos MoLIC com defeito e o comentário do inspetor sobre o defeito encontrado e como ele deveria ser corrigido. Cada inspetor encontrou a seguinte quantidade de erros: o inspetor-1 identificou um erro de Omissão, um erro de Ambiguidade, oito erros de Fato Incorreto e um erro de Inconsistência, o inspetor-2 identificou seis erros de Omissão e dois erros de Inconsistência, o inspetor-3 identificou onze erros de Omissão, três erros de Ambiguidade, cinco erros de Fato Incorreto, dois erros de Informação Estranha e três erros de Inconsistência, totalizando uma coleção de quarenta e quatro erros sendo dezoito erros de Omissão, quatro erros de Ambiguidade, treze erros de Fato Incorreto, dois erros de Informação Estranha e cinco erros de Inconsistência.

Em seguida o moderador reuniu todos os erros encontrados em uma única tabela com os mesmos domínios apresentados acima e a identificação do inspetor que encontrou o erro. Os erros que foram encontrados por mais de um inspetor foram primeiramente agrupados, para que o moderador pudesse analisa-los, escolher o comentário que fosse mais claro, a fim de facilitar as etapas subsequentes, e eliminar os outros registros repetidos. Na discriminação dos defeitos foi reunida toda a equipe para a discussão da correção dos erros encontrados com a modelagem do artefato e os requisitos do domínio. Catorze erros foram identificados como falsos, ou seja, foram apontados como erro mas seus elementos estavam corretos, assim, foram eliminados da lista, e três erros tiveram

seu item de verificação ou tipo de defeitos alterados, finalizando a etapa de discriminação de defeitos com uma lista de vinte e nove defeitos encontrados no artefato.

4. Melhorando a Qualidade do Modelo Inspeccionado

A etapa de retrabalho ocorre após o processo de inspeção, que encontrou vinte e nove defeitos no artefato inspeccionado, com a correção dos defeitos encontrados com base na lista de discriminação. O moderador é responsável por enviar a lista de discriminação finalizada, resultado do processo de inspeção anterior, para o autor do artefato realizar as alterações necessárias para corrigir os defeitos encontrados nos diagramas MoLIC pela equipe de inspeção. Cada item da lista de discriminação é analisado pelo autor do artefato e a alteração no diagrama é realizado.

As principais mudanças feitas pelo autor do artefato, a partir da discriminação de defeitos, foram a adição de conteúdo de fala inicial e das cenas visualizar pedido e confirmar escolha, a correção do tópico da cena de “Painel de controle” para “Gerenciar pedidos de aproveitamento”, a adição de restrições da especificação do sistema que anteriormente não estavam modeladas no diagrama como a obrigatoriedade da submissão do histórico na criação e edição do pedido de aproveitamento de cadeira, a correção dos diálogos das cenas, inserindo os signos e falas adequados e adicionando falas de transição e falas de recuperação de ruptura onde foi apontado pela etapa discriminação de erros.

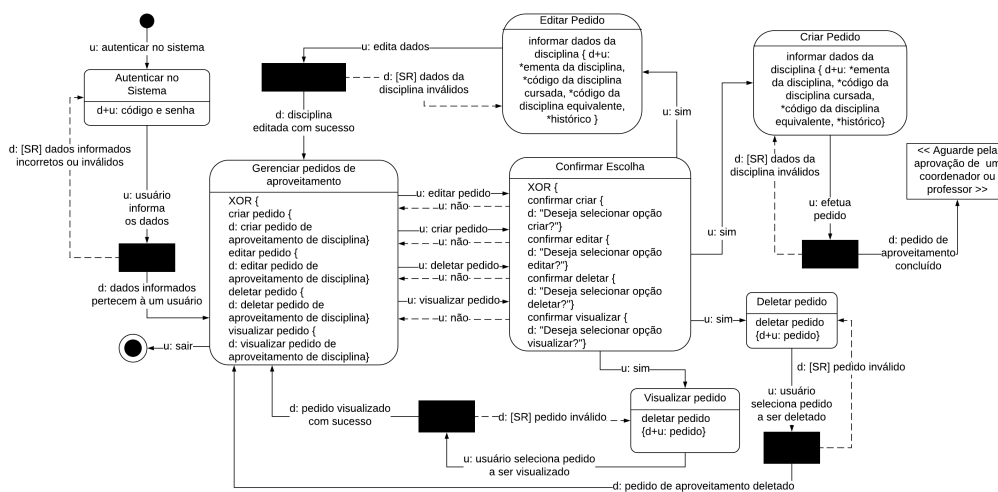


Figura 2. Diagrama de Interação Aproveitamento de Disciplina corrigido

Em seguida, o novo artefato foi novamente enviado para a equipe de inspeção para verificar se todos os itens da lista de discriminação foram atendidos. Ademais com a nova inspeção podem surgir novos defeitos, caso isso ocorra, uma nova etapa de retrabalho faz-se necessária. Assim a equipe de inspeção analisou o novo artefato e concluiu que as mudanças feitas seguiram com corretude a discriminação de defeitos, a definição do diagrama de interação MoLIC e com os requisitos do sistema, encerrando o processo de inspeção.

5. Conclusão

É evidente que o uso de técnicas de inspeção são grandes ferramentas no processo de desenvolvimento de software, facilitando a identificação de defeitos e, assim, diminuindo

os custos do projeto. O uso da linguagem MoLIC torna mais fácil o desenvolvimento de aplicações centrada na experiência do usuário, porém a linguagem ainda apresenta algumas características falhas, assim precisa-se de uma técnica de inspeção bastante robusta para suprimir suas próprias limitações, logo foi usada a técnica de inspeção MoLVERIC Check, que se mostra eficiente para a identificação de erros na construção dos diagramas e na modelagem da comunicação representada pelos mesmos.

Ressaltamos o fato da técnica de inspeção ser baseada em checklist levando a uma facilidade para ser executada. A vasta quantidade de itens de verificação, muito bem separados em domínios específicos para cada elemento do diagrama de interação MoLIC, tornam a inspeção eficiente, mas sentiu-se a falta de itens de verificação, principalmente para elementos pouco usados (como acesso ubíquo). Sendo assim a técnica ainda precisa de melhorias, mesmo que pequenas, para se tornar completa.

Embora seja relativamente rara a utilização de diagramas MoLIC no desenvolvimento de sistemas por ser, de certa forma, nova e ainda pouco consolidada, e os trabalhos relacionados sobre a MoLIC e MoLVERIC sejam escassos, os autores evidenciam que o processo de inspeção foi extremamente útil para a aprendizagem da elaboração de artefatos na linguagem MoLIC e a inspeção usando a técnica MCheck, linguagem e técnica que deviam ser abordadas e praticadas de forma mais extensa dentro do meio acadêmico e empresarial devido aos seus evidentes benefícios ao processo de desenvolvimento de aplicações centradas no usuário.

Referências

- Ackerman, A. F., Buchwald, L. S., and Lewski, F. H. (1989). Software inspections: an effective verification process. *IEEE Software*, 6(3):31–36.
- Camilo, V. (2018). Gamificação - o poder de engajar pessoas de forma divertida. [Online; accessed 17-Novembro-2018].
- Damian, A. L. (2016). Técnicas para inspeção de diagramas molic. Master's thesis. Instituto de Computação.
- de Paula, M. G., da Silva, B. S., and Barbosa, S. D. J. (2005). Using an interaction model as a resource for communication in design. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems, CHI EA '05*, pages 1713–1716, New York, NY, USA. ACM.
- Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. volume 15, pages 182–211.
- Lopes, A., Marques, A. B., Conte, T., and Barbosa, S. D. J. (2015). Estudo de viabilidade da molveric cards: Uma técnica para a inspeção de diagramas molic.
- Qazi, A., Shahzadi, S., and Humayun, M. (2016). A comparative study of software inspection techniques for quality perspective. 10:9–16.
- Travassos, G., Shull, F., Fredericks, M., and Basili, V. R. (1999). Detecting defects in object-oriented designs: Using reading techniques to increase software quality. *SIG-PLAN Not.*, 34(10):47–56.