Produção de objetos de aprendizagem para o ensino de teste de software[1]

Eduardo Amaral, Paulo Henrique da Silva, Fabiane B. Vavassori Benitti

Universidade do Vale do Itajaí (UNIVALI)
Caixa Postal 360 –88303-202 – Itajaí – SC – Brazil

{eduarduamaral, phs.paulohenriquesilva}@gmail.com,
fabiane.benitti@univali.br

Resumo. *O teste de software, visto como uma das principais formas de avaliar a qualidade do software, tem suas técnicas pouco aplicadas pelas empresas, sendo a falta de profissionais habilitados à implantação destas técnicas apontada como um dos principais fatore, evidenciando a necessidade de melhorar o ensino relacionado a testes de software. Uma tecnologia que pode auxiliar neste contexto refere-se aos objetos de aprendizagem, os quais permitem trabalhar um determinado conteúdo com interatividade, em diferentes níveis de granularidade de forma a potencializar o seu reuso. Este artigo descreve o planejamento e construção de um conjunto de objetos de aprendizagem de teste de software, buscando sua aplicação em diferentes currículos e contextos de ensino, como cursos de graduação e treinamentos empresariais.*

1. Introdução

O teste de software é uma das principais ferramentas para manutenção da qualidade do software. Contudo, percebe-se que a grande maioria das empresas não o pratica, devido a falta de profissionais qualificados para aplicação de testes no processo de desenvolvimento (DIAS NETO et al, 2006), o que ocorre em função da pouca ênfase dada ao assunto nas grades curriculares dos cursos de Ciência da Computação e Engenharia de Software (CHEN, 2004). Apesar de muitas pessoas entenderem que uma introdução breve aos testes na ementa de engenharia de software seria suficiente, esta abordagem é inadequada (CHEN, 2004).

A partir da necessidade de apoiar o ensino de teste de software de forma flexível a se adequar a diferentes contextos de ensino (treinamento empresarial e cursos de graduação), bem como considerar variados currículos, propõe-se neste artigo o planejamento e construção de um conjunto de objetos de aprendizagem focados na área de teste de software. Alguns pesquisadores já desenvolveram objetos de aprendizagem para área de teste de software ((BARBOSA; MALDONADO, 2006); (BARBOSA; MALDONADO; SOUZA, 2008)). Contudo, os trabalhos se atêm a metodologia de desenvolvimento e ao desenvolvimento de objeto para um contexto específico. Este artigo aborda um conjunto de objetos para a área de teste e o planejamento de sua granularidade.

Para viabilizar o reuso dos objetos de aprendizagem em diferentes contextos,

[1] Este artigo segue o template dos eventos da SBC (Sociedade Brasileira de Computação), disponível em http://www.sbc.org.br/en/index.php?option=com_jdownloads&task=view.download&catid=32&cid=38&Itemid=195

um fator chave e difícil de equacionar refere-se a granularidade (WILEY, 2000). Embora objetos de aprendizagem maiores sejam mais fáceis de administrar, são mais difíceis de re-contextualizar para outros cenários de aprendizagem diferentes daqueles para os quais foram inicialmente previstos. Objetos de aprendizagem menores podem ser mais precisamente definidos, são mais fáceis de re-contextualizar, mas demandam esforço para organizar com vistas a facilitar sua localização (TAROUCO, FABRE, TAMUSIUNAS, 2003).

2. Planejamento dos Objetos de Aprendizagem

Um dos desafios encontrados foi identificar o que deve ser ensinado referente a teste de software, definindo uma granularidade que propicie o reuso e minimize o esforço de produção, bem como considerar tanto o contexto acadêmico quanto empresarial. Para tanto, a metodologia envolveu a realização de pesquisas documentais, considerando: (i) currículos de referência nacionais (CEEinf, 1999; SBC, 2005) e internacionais (IEEE, 2004; ACM, AIS, IEEE-CS, 2006); (ii) planos de ensino das disciplinas de cursos de graduação voltados à informática; (iii) guia de implementação do MR-MPS em organizações do tipo Fábrica de Teste (SOFTEX, 2009); e (iv) conhecimentos exigidos dos candidatos ao Certificado Brasileiro de Teste de Software (ALATS, 2011). O mapeamento resultante dos itens i e ii estão detalhados em (BENITTI; ALBANO, 2012).

A partir da pesquisa documental realizada foi possível definir os objetos de aprendizagem e a granularidade dos mesmos (Tabela 1). Importante observar que a etapa do planejamento incorporou-se além dos tópicos, o nível de conhecimento (conforme taxonomia de Bloom e constante em IEEE (2004)), bem como a origem de cada tópico e seus objetivos de aprendizagem.

Tabela 1: Objetos de aprendizagem propostos e seus objetivos de aprendizagem

Objetos de Aprendizagem para Teste de Software		Nível de Conhecimento (Taxonomia de Bloom)	Objetivos de aprendizagem
1	Fundamentos de Teste de Software (SB; CN; CT; MB)	Compreensão	✓ Conhecer a importância dos testes de software no desenvolvimento de sistemas ✓ Identificar os significados dos diferentes termos utilizados na área de teste de software.
2	Técnicas de Teste (SB)	Compreensão	✓ Conceituar e entender as diferenças entre as duas principais técnicas de teste: caixa branca e caixa preta. ✓ Conhecer os critérios de cada técnica. ✓ Compreender porque essas técnicas e seus critérios são relevantes dentro do processo de teste.
3	Técnica Caixa Branca (SB; CN; CT; MB)	Aplicação	✓ Conhecer os critérios da técnica caixa branca ✓ Construir o grafo de fluxo de controle a partir do código fonte ✓ Calcular a complexidade ciclomática a partir do grafo de fluxo de controle

4	Técnica Caixa Branca: Critérios Baseados no Fluxo de Controle (SB; CN; CT; MB)	Aplicação	✓ Conhecer os critérios baseados no fluxo de controle ✓ Aplicar os critérios baseados no fluxo de controle para criação de casos de teste
5	Técnica Caixa Branca: Critérios Baseados na Complexidade (SB; CN; CT; MB)	Aplicação	✓ Usar a complexidade ciclomática para determinar o número de casos de teste ✓ Definir os caminhos linearmente independentes no Gráfico de Fluxo de Controle
6	Técnica Caixa Branca: Critérios Baseados no Fluxo de Dados (SB; CN; CT; MB)	Aplicação	✓ Conhecer os critérios baseados no fluxo de dados ✓ Construir o grafo Def-Uso a partir do Gráfico de Fluxo de Controle ✓ Aplicar critérios baseados no fluxo de dados para definição de casos de teste
7	Técnica Caixa Preta: Particionamento por equivalência e análise de valor limite (SB; CN; CT; MB)	Aplicação	✓ Aplicar as técnicas de particionamento por equivalência e análise de valor limite ✓ Saber identificar casos de teste a partir da aplicação das técnicas
8	Técnica de Caixa Preta: tabela de decisão e grafo causa e efeito (SB; CN; CT; MB)	Aplicação	✓ Aplicar as técnicas de tabela de decisão e grafo de causa e efeito ✓ Saber identificar casos de teste a partir da aplicação das técnicas
9	Técnica de Caixa Preta: critérios baseados em casos de uso (SB; CN; CT; MB)	Aplicação	✓ Entender os casos de uso como um artefato útil tanto para modelagem do sistema como à atividade de teste ✓ Conhecer e utilizar o critério de teste baseado em casos de uso
10	Visão Geral dos Níveis e Tipos de Teste (SB; CN; CT; MB)	Compreensão	✓ Entender como os testes podem ser abordados em cada fase do desenvolvimento do software (níveis de teste) ✓ Descobrir como alguns testes podem ser organizados conforme um objetivo específico (tipos de teste) ✓ Conhecer algumas possibilidades para automatização de testes através de ferramentas específicas
11	Níveis de Teste: o teste no ciclo de vida do desenvolvimento (SB; CN; CT; MB)	Aplicação	✓ Compreender os níveis de teste de software e sua relação com o processo de desenvolvimento ✓ Saber aplicar os teste em diferentes níveis ✓ Utilizar ferramentas para automatizar alguns níveis de teste
12	Tipos de teste: testes quanto ao objetivo (SB; CN; CT; MB)	Aplicação	✓ Saber realizar testes de volume, stress, tempo e conformidade e usabilidade
13	Outros Tipos de Teste (SB; CN; CT; MB)	Aplicação	✓ Saber realizar tipos específicos de teste, a citar: segurança, instalação, confiabilidade, recuperação, configuração e compatibilidade

14	Visão geral do processo de teste (SB)	Compreensão	✓ Conhecer um processo de teste, bem como a necessidade de se estabelecer um processo adequado de testes em empresas de desenvolvimento de software.
15	Por dentro do Processo de testes (SB; CT; CN)	Aplicação	✓ Organizar um conjunto de atividades a serem realizadas durante os testes. ✓ Organizar e empregar processo de teste para um projeto de desenvolvimento de software. ✓ Utilizar artefatos para documentação e execução do processo de teste.
16	Questões quanto ao gerenciamento (SB; CT; MB)	Análise	✓ Definir e analisar indicadores dentro do processo de teste que darão suporte para realização de estimativas e análise de riscos. ✓ Conhecer ferramentas que podem auxiliar no processo de teste.
17	TDD **(**CN)	Aplicação	✓ Conhecer um processo ágil de desenvolvimento baseado em testes ✓ Saber aplicar algumas práticas relacionadas ao TDD

Legenda – SB: Swebok, CN: Currículos Nacionais, CT: Certificações, MB: MPS.BR – indica na pesquisa documental onde o tema foi encontrado.

Vencido o primeiro desafio de tratar a granularidade de um conjunto de objetos de aprendizagem focados na área de teste de software e, tendo-se definido os objetos, bem como a definição do nível e objetivos de aprendizagem, o próximo passo é projetar detalhadamente cada objeto.

Para esta etapa de análise e projeto do objeto de aprendizagem foi utilizada uma abordagem baseada na matriz de design instrucional proposta por Filatro (2008). Através dessa matriz é possível organizar de maneira abrangente os objetivos, papéis, ferramentas, conteúdos, atividades, avaliações e os ambientes necessários, conforme é possível visualizar na Figura 1. A matriz de design instrucional é caracterizada por orientar a equipe de desenvolvimento e design, podendo ser considerada como um mapa do conteúdo que será abordado (FILATRO, 2008).

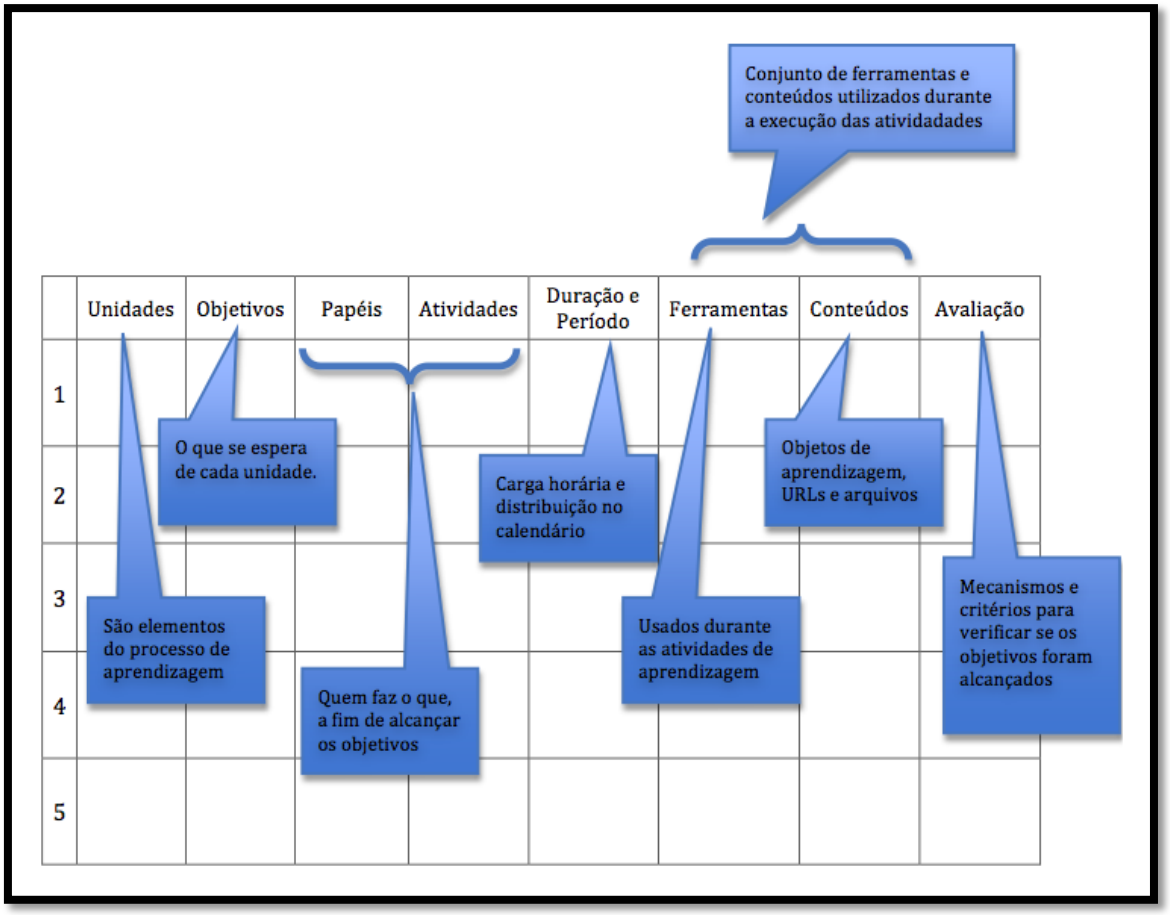


Figura 1 - Matriz de design instrucional
Fonte: FILATRO (2008).

Contudo, para o projeto dos objetos foi realizada uma adaptação da matriz, conforme detalhado na tabela 2.

Tabela 2- Adaptação de elementos da matriz de design instrucional

Elemento da matriz original (Filatro, 2008)	**Aplicação de cada elemento na matriz instanciada**
Unidades	Cada Unidade representa um objeto de aprendizagem, definido por: Identificador (AO <<número>>), título e Nível de aprendizagem (conforme Taxonomia de Bloom).
Pré-requisitos	Elemento adicionado que visa descrever os conhecimentos prévios que o aluno deve ter para obter maior aproveitamento do conteúdo do objeto proposto.
Objetivos	Elemento utilizado conforme matriz original
Papéis e Atividades	Elementos suprimidos, pois neste projeto não há uma equipe vinculada.
Duração e período	Elemento suprimido.
Ferramentas	Descritas de forma vinculada a cada tópico apontado no Conteúdo Programático.

Conteúdos	Descrito utilizando estrutura de tópicos, vinculando abordagem proposta e ferramentas a serem utilizadas.
Avaliação	Descrita no como tópico juntamente ligado ao conteúdo.
Leitura sugerida e próximos passos	Elemento adicionado visando fornecer referência para aprofundamento do estudo, bem como diretriz para outros conteúdos relacionados.

A tabela 3 ilustra um exemplo de matriz instrucional desenvolvida para o projeto de um objeto de aprendizagem. O projeto leva em consideração a pesquisa realizada na etapa inicial do projeto, vinculando conteúdos encontrados nas diferentes fontes consultadas (diretrizes curriculares internacionais e nacionais, currículos nacionais, MPS.BR, e conteúdo abordado em certificações.

Tabela 3 – Matriz instrucional do OA1

Identificador: OA1	**Título: Fundamentos de Teste de Software**
Nível de aprendizagem: Compreensão	**Objetivo:** - Conhecer a importância dos testes de software no desenvolvimento de sistemas - Identificar os significados dos diferentes termos utilizados na área de teste de software.
Pré-Requisitos: -	
Conteúdo Programático	

Estrutura de tópicos:

- Por que realizar testes no software?
- Conceitos de Teste de Software
- Terminologia relacionada à testes
- Você sabia?
- Verificação e Validação
- Entendendo o Processo de Teste
- Exercícios
- Leituras sugeridas/próximos passos

➢ Por que realizar testes no software? – Despertar o interesse do aluno ao conteúdo de teste de software

Abordagem: Apresentar personagem testador questionando o aluno sobre a importância de testar o software. O usuário pode interagir informando se considera ou não importante. Por fim, os personagens falam da importância do teste e apresentam o objetivo do objeto.
Ferramentas: Personagens estáticos com balões de fala.

➢ Conceito de teste

Abordagem: Apresentar definições citadas por autores da área, sugerir um conceito, mas também deixar em aberto para que o aluno, a partir dessas informações, elabore seu conceito.
Ferramentas: Personagens estáticos com balões de fala e vídeo.

➢ Terminologia relacionada à testes

Abordagem: Como a área de testes define sutis diferenças entre termos que em geral poderiam ser considerados sinônimos, esses conceitos são apresentados em forma de tópicos, segundo definição do IEEE. Os termos a serem abordados são engano, defeito, erro e falha. Um exemplo ilustrando em uma situação real o emprego de cada termo deve ser apresentado.
Ferramentas: Personagens estáticos com balões de fala, imagens e elemento interativo.

➢Você sabia? – Por que *bug* (inseto) é sinônimo de erro de software?

Abordagem: Apresentar, como um fato curioso, os relatos sobre insetos que causavam problemas nos antigos computadores, e por esse motivo, acabaram sendo associados à erro de programação. O quadro também pode citar erros famosos, casos de falhas em software e suas

consequências.
Ferramentas: Personagens estáticos com balões de fala, imagens e elemento interativo.

➢ Verificação e Validação– Conceituar e distinguir estes conceitos
Abordagem: Apresentar uma animação de uma fábrica de software e identificar no processo a validação e verificação através do diálogo entre personagens que representam um testador e um analista de testes. Além da cena em que os conceitos são contextualizados, os conceitos também são claramente apresentados através de texto explicativo.
Ferramentas: Personagens estáticos com balões de fala, animação e elemento interativo.

➢Entendendo o processo de teste
Abordagem: Apresentar brevemente um conceito sobre processo de teste.
Apresentação dos personagens incorporando papéis do processo de teste, cada um explicando sua função no processo.
Ferramentas: Personagens estáticos com balões de fala e elemento interativo.

➢ Exercícios
Abordagem: Serão elaborados 9 exercícios em nível de compreensão, conforme a taxonomia de Bloom.
Ferramentas: Questões de múltipla escolha, exercícios de completar frases com conceitos e ligar colunas.

➢Leituras sugeridas e próximos passos
- Leituras:
  - ALMEIDA, Carla. Introdução ao Teste de Software. Disponível em: <http://www.linhadecodigo.com.br/artigo/2775/introducao-ao-teste-de-software.aspx>. Acessado em março de 2012.
  - DELAMARO, Márcio Eduardo; JINO, Mario; MALDONADO, José Carlos. Introdução ao teste de software. Rio de Janeiro: Elsevier, 2007.
  - DEVMEDIA. Introdução a Teste de Software. Disponível em: <http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>. Acessado em março de 2012.
  - MALDONADO, José Carlos; FABBRI, Sandra Camargo Pinto Ferraz. Qualidade de software: teoria e prática. São Paulo: Prentice Hall, 2001.
  - MOREIRA, Trayahú; RIOS, Emerson. Teste de Software. 2 e.d. São Paulo: Alta Books, 2006.
  - NETO, Michelle. Vídeo Aula1 – Conceitos iniciais em teste de Software. Disponível em: <http://www.youtube.com/watch?v=7kec8BT8gU4>. Acessado em março de 2012.
- Próximos passos:
  - Você conheceu os conceitos básicos sobre teste de software, viu que existem testes estáticos que podem ser executados mesmo antes da codificação do programa e aprendeu o conceito de processo de teste. Mas isso é só o começo. Você pode dar continuidade aos seus estudos explorando diversos aspectos do teste de software, por exemplo:
    - Estudar as técnicas de teste caixa branca e caixa preta;
    - Estudar os níveis de teste: unidade, integração e sistema.

Esta estrutura preliminar de detalhamento foi elaborada para os 17 objetos de aprendizagem previstos (Tabela 1) e orientam na etapa de produção.

3. Produção

Para construção dos objetos de aprendizagem foi utilizada como ferramenta de autoria o Articulate Studio® 09. O Articulate funciona como um plug-in do Microsoft PowerPoint, sendo assim um ambiente já conhecido, o que facilitou a sua utilização. A ferramenta permitiu elaborar uma padronização de layouts para formatação estética dos objetos. Além disso, o Articulate possui ferramentas para criação de diversos tipos de questionários, narrações, captura das ações na tela para criação de vídeo-aulas e,

principalmente, permite o empacotamento dos objetos segundo padrão SCORM. Desta forma, pode-se distribuir o objeto em formato compatível com os principais LMS (Learning Management System) disponíveis atualmente.

Na sequencia as Figuras 2 a 10 apresentam alguns exemplos de telas com o intuito de evidenciar o design, recursos e conteúdos abordados nos objetos de aprendizagem produzidos.

Exemplo do OA1 – Terminologia relacionada à testes

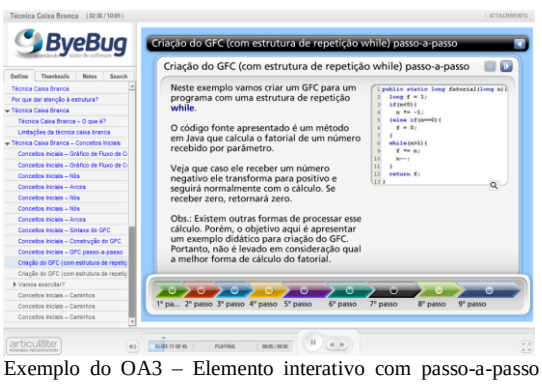
Exemplo do OA1 – Entendendo o processo de teste

Figura 2. Exemplos do Objeto de Aprendizagem 1 - Fundamentos de Teste de Software

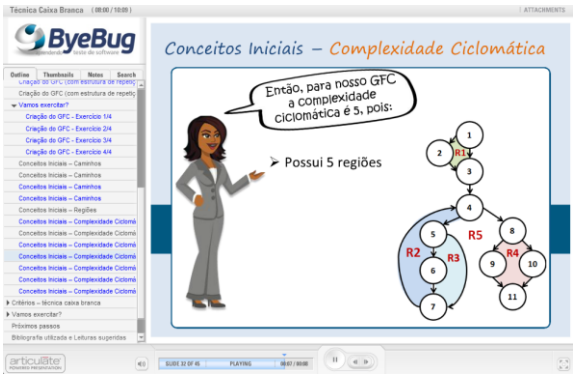
Exemplo do OA2 – Elemento interativo para teste de caixa preta

Exemplo do OA2 – Elemento interativo para teste de caixa branca

Figura 3. Exemplos do Objeto de Aprendizagem 2 - Técnicas de Teste

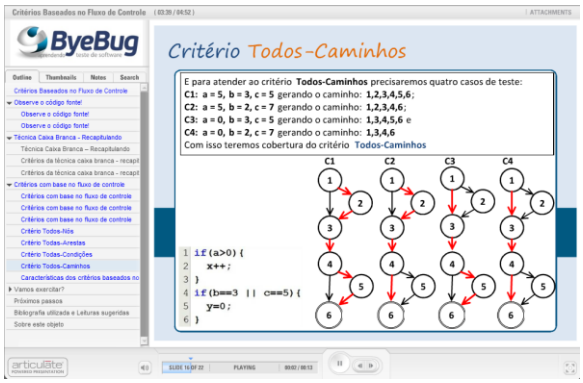


Exemplo do OA3 – Elemento interativo com passo-a-passo para criação de um GFC

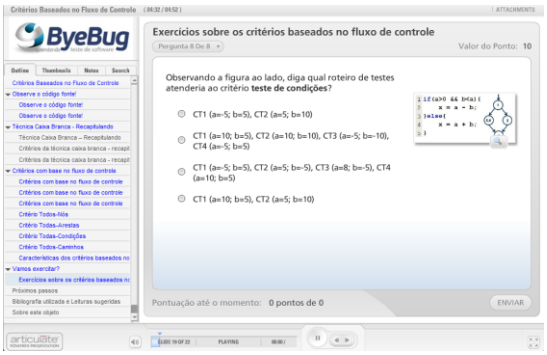


Exemplo do OA3 – Cálculo da complexidade ciclomática

Figura 4. Exemplos do Objeto de Aprendizagem 3 - Técnica Caixa Branca



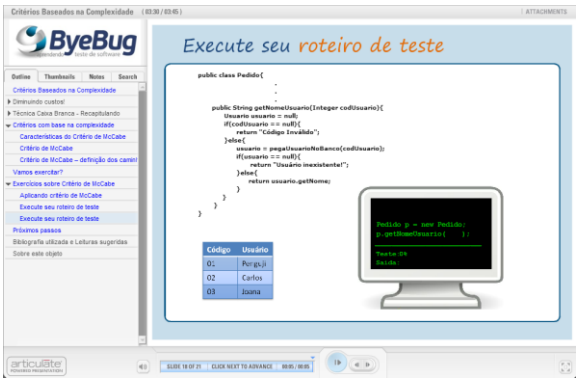
Exemplo do OA4 – Critério Todos-Caminhos



Exemplo do OA4 – Exercício critério teste de condições

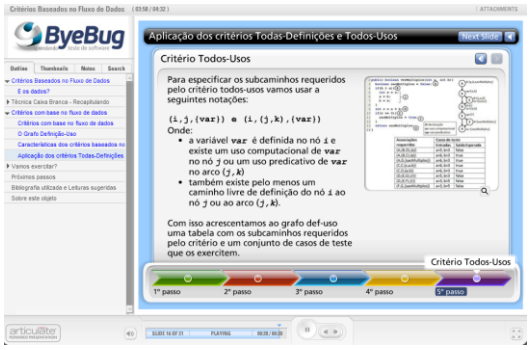
Figura 5. Exemplos do Objeto de Aprendizagem 4 - Técnica Caixa Branca: Critérios Baseados no Fluxo de Controle

Exemplo do OA5 – Animação ilustrando o critério de McCabe



Exemplo do OA5 – Aplicando o critério de McCabe

Figura 6. Exemplos do Objeto de Aprendizagem 5 - Técnica Caixa Branca: Critérios Baseados na Complexidade

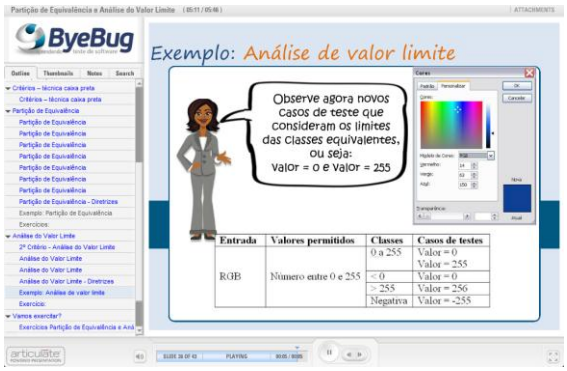


Exemplo do OA6 – Elemento interativo passo-a-passo da aplicação do critério Todas-Definições e Todos- Usos

Exemplo do OA6 – Exercício de critério baseado no fluxo de dados

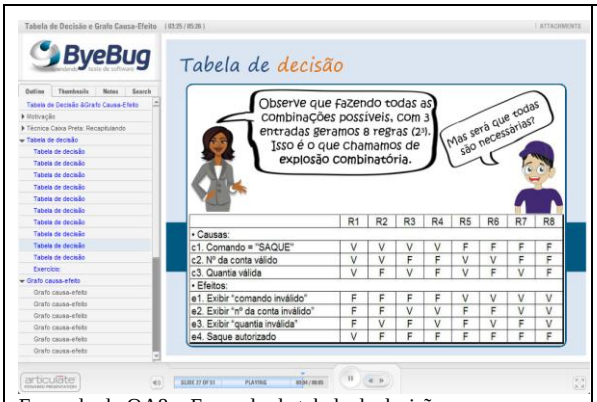
Figura 7. Exemplos do Objeto de Aprendizagem 6 - Técnica Caixa Branca: Critérios Baseados no Fluxo de Dados

Exemplo do OA7 – Elemento interativo com as diretrizes para particionamento por equivalência

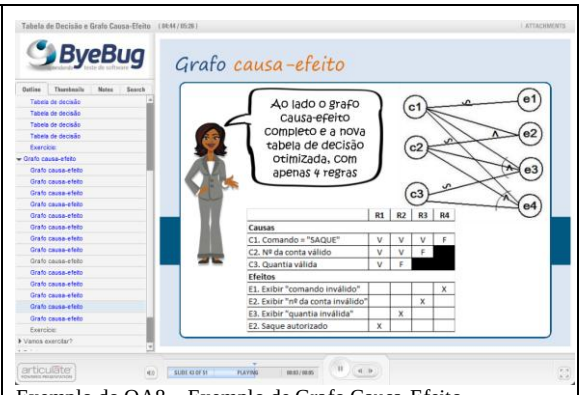


Exemplo do OA7 – Exemplo de Análise de valor limite

Figura 8. Exemplos do Objeto de Aprendizagem 7 - Técnica Caixa Preta: Particionamento por equivalência e análise de valor limite

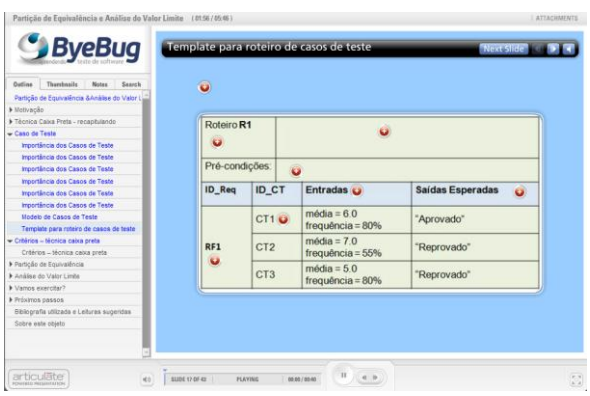


Exemplo do OA8 – Exemplo de tabela de decisão

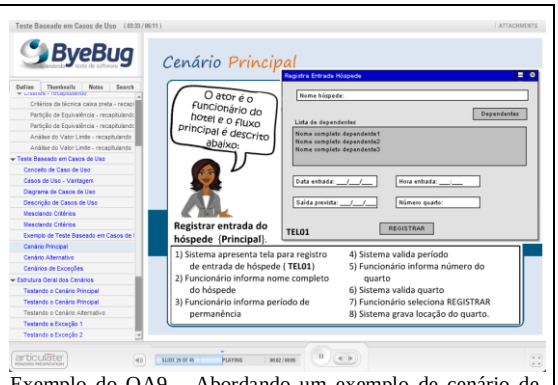


Exemplo do OA8 – Exemplo de Grafo Causa-Efeito

Figura 9. Exemplos do Objeto de Aprendizagem 8 - Técnica de Caixa Preta: tabela de decisão e grafo causa e efeito



Exemplo do OA9 – Elemento interativo apresentando template para um roteiro de testes



Exemplo do OA9 – Abordando um exemplo de cenário de caso de uso para teste

Figura 10. Exemplos do Objeto de Aprendizagem 9 - Técnica de Caixa Preta: critérios baseados em casos de uso

4. Conclusões

A meta principal deste trabalho consiste em planejar e desenvolver OA para o ensino de teste de software. Para tanto, o planejamento envolveu o estudo da granularidade adequada para os objetos, agrupando tópicos e conteúdos sugeridos por diretrizes curriculares nacionais e internacionais, planos de ensino de disciplinas relativas ao teste de software de cursos superiores, conteúdo exigido para certificação CBTS e conteúdo relativo a teste de software no guia de implementação do MPS-BR. Este estudo inicial gerou a proposta de elaboração de 17 objetos de aprendizagem, abrangendo diferentes níveis da taxonomia de Bloom.

Para cada um dos objetos de aprendizagem foi elaborada uma matriz instrucional para orientar o desenvolvimento dos objetos, envolvendo os objetivos de aprendizagem, conteúdos a serem abordados, ferramentas e avaliação.

Atualmente, há 9 objetos de aprendizagem concluídos ou em fase de polimento (pequenos ajustes referente a qualidade de imagens, correções ortográficas/gramaticais). Um website está em desenvolvimento para disponibilizar os objetos à comunidade.

Um novo projeto está em andamento para conclusão dos 8 objetos faltantes. Pretende-se, após concluí-los, aplicar experimentos (na forma de minicursos, com grupo experimental e de controle) com turmas de graduação e treinamento empresarial visando identificar o potencial de aprendizagem e a satisfação dos usuários ao interagirem com os objetos. Os experimentos também pretendem averiguar o reuso dos objetos em diferentes contextos.

Obteve-se também como resultado deste projeto a publicação de um resumo e apresentação de Poster no Computer on the Beach (conforme pode ser verificado em http://www.computeronthebeach.com.br/portal/programacao).

Agradecimentos

Os autores agradecem ao financiamento recebido da UNIVALI / Governo do Estado de Santa Catarina através do Artigo 171. Também agradecem a participação do acadêmico Edson Lucas Albano do Mestrado em Computação Aplicada.

Referências

ACM; AIS; IEEE-CS. (2006) “Computing Curricula 2005.” Disponível em: <www.acm.org/education/curric_vols/CC2005-March06Final.pdf>. Acesso em: 05 jun. 2011

ALATS. Manual do Candidato à Certificação Brasileira de Teste de Software. Disponível em: http://www.alats.org.br/portal/images/cbts/manual_candidatos_cbts.pdf. Acessado em: 04 nov. 2012.

BARBOSA, E. F.; MALDONADO, J. C. Establishing a Mutation Testing Educational Module based on IMA-CID. Second Workshop on Mutation Analysis, 2006.

BARBOSA, E. F.; MALDONADO, J. C.; SOUZA, S. R. S. An Experience on Applying Learning Mechanisms for Teaching Inspection and Software Testing. 21st Conference on Software Engineering Education and Training, April, 2008.

BENITTI, F.B.V.; ALBANO, E.L. Teste de Software: o que e como ensinar? In: Workshop sobre Educação em Computação, XXXII Congresso da Sociedade Brasileira de Computação, **Anais...** Curitiba: SBC, 2012.

CEEInf. Diretrizes Curriculares de Cursos da Área de Computação e Informática. Comissão de Especialistas de Ensino de Computação de Informática (CEEInf). Secretaria de Educação Superior do MEC (SESu/MEC). 1999

CHEN T. Y. Experience with teaching black-box testing in a Computer Science/Software Engineering Curriculum. **IEEE Transactions on Education**, v. 47, n. 1, fev. 2004.

DIAS NETO, A. C.; NATALI, A. C. C.; ROCHA, A. R.;TRAVASSOS, G. H. Caracterização do estado da prática das atividades de teste em um cenário de desenvolvimento de software brasileiro. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 5., 2006, Vila Velha. **Anais...**Vila Velha: SBC, 2006. p. 27-41.

FILATRO, Andrea. Design Instrucional na Prática. Prentice-Hall, 2008.

IEEE. SWEBOK - Guide to the Software Engineering Body of Knowledge, 2004 Version. Disponível em: <http://www.computer.org/>. Acesso em: 25 maio 2010.

SBC - Sociedade Brasileira da Computação. (2005) Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia da Computação. Disponível em: http://www.sbc.org.br/. Acesso em: 12 maio 2009.

SOFTEX. MPS.BR: Guia de Implementação – Parte 10: Implementação do MR-MPS em organizações do tipo Fábrica de Teste, Maio 2009. Disponível em: <www.softex.org.br>. Acessado em: 26 de maio de 2011.

TAROUCO, L. M. R.; FABRE, M. C. J. M.; TAMUSIUNAS, F. R. Reusabilidade de objetos educacionais. **Revista Novas Tecnologias na Educação**. Porto Alegre: UFRGS, 2003.

WILEY, D. A. **Connecting learning objects to instructional design theory**: a definition, a metaphor, and a taxonomy. In: The Instructional Use of Learning Objects, 2000.