



REAd - Revista Eletrônica de
Administração

ISSN: 1980-4164

ea_read@ufrgs.br

Universidade Federal do Rio Grande do
Sul
Brasil

Becker Franzen, Marcelo; Porto Bellini, Carlo Gabriel

ARTE OU PRÁTICA EM TESTE DE SOFTWARE?

REAd - Revista Eletrônica de Administração, vol. 11, núm. 3, mayo-junio, 2005, pp. 1-25

Universidade Federal do Rio Grande do Sul

Porto Alegre, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=401137446003>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica

Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal

Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

ARTE OU PRÁTICA EM TESTE DE *SOFTWARE*?

Marcelo Becker Franzen¹

Rua Álvaro Nicofê, 220 – Vila Jardim
CEP 91320-020 Porto Alegre/RS Brasil
E-mail: mbfranzen@yahoo.com.br

Carlo Gabriel Porto Bellini²

Avenida Ipiranga, 3377/901 – Santana
CEP 90610-001 Porto Alegre/RS Brasil
E-mail: cgpbellini@ea.ufrgs.br

¹ Universidade do Vale do Rio dos Sinos – UNISINOS

Informática – Análise de Sistemas
CEP 93022-000 São Leopoldo/RS Brasil

² Universidade Federal do Rio Grande do Sul – UFRGS

Escola de Administração – EA
CEP 90010-460 Porto Alegre/RS Brasil

Resumo

Há uma crescente preocupação na indústria em melhorar a qualidade de produtos e serviços, e o setor de informática não está alheio a isto. Entidades reguladoras governamentais e organizações privadas desenvolvem normas de qualidade refletindo melhores práticas da indústria (*estado da prática*) e descobertas científicas (*estado da arte*), dada a crescente demanda por sistemas alinhados a compromissos de projeto e necessidades do cliente; entre as proposições, encontram-se as de teste de *software*, objeto do presente artigo. Aqui, apresenta-se, comparativamente a recomendações da literatura, um panorama das práticas de teste de *software* em um dos principais pólos industriais brasileiros de produtos e serviços de informática, localizado na Região Sul do país. Em 2003, 14 das 20 empresas que integravam o referido pólo foram estudadas em estudo de caso auxiliado por entrevistas em profundidade junto a alguns de seus profissionais de teste de *software*, e os resultados dão conta de que há uma correlação entre o tempo em que as empresas atuam no mercado e a conformidade de seus procedimentos de teste de *software* às recomendações teóricas. Questionamentos

importantes sobre a indústria brasileira de *software* são derivados, bem como sobre a primazia a ser dada à arte ou à prática neste campo.

Palavras-chave: teste de *software*, pólos de TI, indústria brasileira de *software*, estado da arte, estado da prática

Abstract

There is an increasing concern in the industry on improving product and service quality, and the software sector is aware of that. Governmental agencies and private organizations develop quality standards reflecting best practices (*state of the practice*) and scientific findings (*state of the art*) due to the increasing demand for systems aligned to project requirements and customer needs; software testing, the central topic of the present article, is one of the areas targeted for standardization (be it in the process itself or in managerial issues). Here we contrast the literature on software testing with actual practices – as perceived by experts – of a major cluster of software companies in Southern Brazil. In 2003, 14 out of the 20 companies constituting the cluster were investigated under a case study approach supported by in-depth interviews with some of the companies' software testing professionals, and findings suggest that there is a correlation between a company's time in the market and the adherence of its software testing procedures to the theoretical exhortations. Important topics are raised concerning the Brazilian software industry, as well as about the priority that should be given to the art or to the practice in the field.

Keywords: software testing, IT clusters, Brazilian software industry, state of the art, state of the practice

1 Introdução

Investimentos e incentivos destinados à melhoria da qualidade de produtos e serviços das empresas brasileiras de desenvolvimento de *software* estão em conformidade com o aumento verificado nos últimos anos relativamente ao número de empresas brasileiras, das mais diversas áreas, que buscam certificações reconhecidas, como aquelas conferidas pela ISO – organização não governamental com o objetivo de promover o desenvolvimento de padrões e atividades correlatas para facilitar o intercâmbio internacional de produtos e serviços e desenvolver a cooperação nas esferas intelectual, científica tecnológica e

econômica. Especificamente na área de *software*, outra forma com que se trabalham critérios de qualidade é dada pelos modelos CMM (modelos de maturidade de competências), desenvolvidos pelo Software Engineering Institute da Carnegie Mellon University (Pittsburgh, EUA).

Pode-se argumentar que a qualidade de um produto final depende, em grande parte, dos métodos, técnicas e ferramentas que o desenvolvedor utiliza. Entre os procedimentos que buscam a qualidade de um produto, os testes são aplicados para que ele chegue ao mercado com o menor índice de defeitos possível, observando-se que haja uma relação custo-benefício que não comprometa o seu preço final (Peters e Pedrycz, 2001; Leite, 2001; Beizer, 1990). As atividades de teste de *software* (TS) envolvem etapas definidas (planejamento, projeto de casos de teste, execução e avaliação dos resultados – Maldonado e Fabbri, 2001b) e devem ser realizadas sob o rigor da técnica e planejadas de acordo com métodos específicos para cada caso (Paula Filho, 2001). TS é, de fato, aspecto crucial do desenvolvimento de *software* (Huq, 2000), sobretudo quando se entende que o alinhamento estratégico de sistemas de informação (sistemas que incluem um componente *software* – Stamelos *et al.*, 2003) é um dos maiores desafios dos executivos (Chan *et al.*, 1997).

O presente trabalho versa sobre TS no contexto específico de um dos pólos mais destacados da indústria brasileira de informática, localizado na Região Sul do país e já discutido por outros autores (*e.g.*, Pereira e Bellini, 2002; Bellini e Pedrozo, 2001a; Bellini e Pedrozo, 2001b), doravante denominado “Pólo”. O Pólo foi idealizado para reunir empresas da área de informática em um espaço físico próximo e unir esforços para promover níveis de qualificação dessas empresas. Ele é composto por uma unidade de desenvolvimento tecnológico – que se divide em incubadora tecnológica e condomínio de empresas, com o papel de apoio à formação e ao crescimento de empresas que estejam entrando no mercado – e por um parque tecnológico – este visando a abrigar empresas que já estejam consolidadas no mercado, que desenvolvam *software* ou que prestem serviços na área de informática, tendo como principal objetivo qualificar e ampliar as atividades dessas empresas.

Com o intuito de estimar o nível de conformidade dos procedimentos de TS das empresas do Pólo com as recomendações da literatura, desenvolveu-se um estudo de caso junto a 14 de suas 20 empresas durante o ano de 2003. O estudo foi apoiado por entrevistas em profundidade (roteiro desenvolvido e validado pelos autores) com profissionais responsáveis por TS em cada empresa. Os resultados da pesquisa indicam diferenças de conformidade entre as empresas estudadas; de especial nota é o fato de que as empresas mais

experientes no mercado parecem estar mais comprometidas com as atividades de TS propostas pela literatura, sugerindo, entre outras coisas, que o nível de maturidade de processo de *software* e a institucionalização da reputação da empresa junto à clientela tenham relação com este fato.

O presente artigo apresenta (1) um panorama geral sobre TS e seus principais enfoques, (2) aspectos relacionados à qualidade do desenvolvimento de *software*, (3) o estudo de caso realizado junto ao Pólo, e (4) conclusões, limitações e recomendações de pesquisa.

2 Teste de *Software*

Citações sobre teste de *software* (TS) são encontradas na literatura desde 1950. A partir dos anos 1970, as atividades de TS eram executadas pelos próprios programadores ou analistas de sistemas. Nos anos seguintes, os usuários passaram a determinar quando um sistema poderia ser entregue à produção. As conseqüências de se tratar TS desta maneira podiam ser percebidas na forma de uma eterna dependência dos clientes pelas empresas que desenvolviam sistemas, na tarefa de corrigirem erros (Rios, 2002). Hoje, sabe-se que a maioria das empresas de desenvolvimento de *software* lida com TS de uma maneira que pode ser caracterizada como amadora, em que as técnicas são empregadas de forma incorreta, imprecisa ou incompleta (Hetzel, 1987; Peters e Pedrycz, 2001).

Myers (1979) define TS como o processo de executar um programa com o objetivo de encontrar erros. Com isto, Myers determina que as atividades de teste partam do pressuposto de que o *software* contém erros e de que esses devem ser descobertos, enfrentando a proposição, até então defendida, de que um teste bem-sucedido era aquele que não encontrava erros. Outro objetivo traçado por Myers (1979) e ratificado por Pressman (1995) diz que um bom caso de teste é aquele que tem uma elevada probabilidade de revelar um erro ainda não descoberto; ou seja, quando se define um caso de teste, devem ser contempladas todas as possibilidades onde possa haver erros em um *software* (Myers, 1979; Pressman, 1995). Por fim, Pyhäjärvi e Rautiainen (2004) mencionam que TS também representa uma importante ferramenta na busca pela satisfação do cliente – um dos principais objetivos das pesquisas em sistemas de informação (Woodroof e Burg, 2003; Zviran e Erlich, 2003) – e pela melhoria do processo de desenvolvimento.

2.1 Ciclo de Vida de *Software*

Independentemente da complexidade do *software* a ser produzido, Maldonado *et al.* (2001) indicam que o seu processo apresenta três fases distintas:

- *definição* – definem-se as informações a serem processadas, funções e desempenho desejados, interfaces a serem estabelecidas, restrições de projeto e critérios de validação;
- *desenvolvimento* – definem-se as estruturas de dados, a arquitetura do *software* e como as atividades de teste devem ser conduzidas; e
- *manutenção* – focam-se aspectos ligados às modificações ocorridas no *software*, sejam elas decorrentes de erros, alterações ou melhoramentos efetuados a partir de uma nova necessidade do usuário.

Outros aspectos além dos anteriormente citados devem ser tidos em conta quando da implantação de um processo de desenvolvimento de *software*. Esses aspectos estão ligados às características específicas de cada organização e das pessoas envolvidas no processo de produção, e estão presentes na norma NBR ISO/IEC 12207 – que relaciona os aspectos intrínsecos e ligados à cultura ao chamado *processo de adaptação*, que objetiva auxiliar na adaptação da norma às peculiaridades de cada organização (Machado, 2001). Além do processo de adaptação, a norma NBR ISO/IEC 12207 apresenta outros três macroprocessos principais que contribuem para a maximização da qualidade do produto final: *processos fundamentais*, *de apoio* e *organizacionais*. Apesar de os processos organizacionais tratarem de aspectos extremamente importantes de gestão, buscando a melhoria contínua da estrutura e dos processos organizacionais, o assunto não será tratado no presente trabalho, por não haverem sido localizados na literatura aspectos diretamente relacionados a TS.

2.2 Processos Fundamentais

Inserido nos processos fundamentais, há um subprocesso denominado *processo de desenvolvimento*, onde estão embutidas atividades diversas, como projeto, integração e atividades relacionadas à análise de requisitos, e nele se aborda o TS com destaque para a adoção de uma série de métodos, técnicas e atividades para se assegurar que o código seja eficiente e esteja livre de erros (Gimenez, 2001). Outro aspecto ressaltado por Gimenez (2001) é a necessidade de o desenvolvedor adotar um plano de integração dos módulos de um *software*, no qual haja testes que certifiquem que o comportamento desse *software*, após a integração, esteja de acordo com os requisitos estabelecidos na documentação de especificação de requisitos do *software*. Concomitantemente, outros procedimentos de TS

devem ser adotados, como teste de cobertura, teste operacional e teste de qualificação, que verificam e validam os requisitos previamente estabelecidos.

2.3 Processos de Apoio

Entre os processos de apoio, que buscam auxiliar outro processo para contribuir para a qualidade do projeto, há um processo chamado *verificação e validação de software*, em que o *software* deve ser exposto a atividades de teste que minimizem a ocorrência de erros. Tais atividades são consideradas elementos críticos para a garantia da qualidade e devem ser planejadas durante todo o desenvolvimento (Maldonado e Fabbri, 2001a).

2.4 Fases

As atividades de TS ainda podem ser divididas de acordo com a fase do desenvolvimento e a abordagem empregada, seja procedimental ou orientada a objeto (OO), conforme Beizer (1990) e Maldonado e Fabbri (2001b). As fases são:

- *teste de unidade* – aplicado a sub-rotinas e funções (na abordagem procedimental) e a métodos (na abordagem OO);
- *teste de integração* – aplicado à relação entre dois ou mais subsistemas (na abordagem procedimental) e à classe, *cluster*, componente e subclasse (na abordagem OO); e
- *teste de sistema* – na abordagem procedimental e na OO, é aplicado para verificação de se o sistema está de acordo com as especificações.

2.5 Teste Estático e Teste Dinâmico

Na literatura, foram encontradas várias classificações para TS, entre elas destacando-se a condição de execução em que o *software* se encontra. Assim, os testes podem ser classificados em estáticos e dinâmicos.

O *teste estático* é aplicado quando o *software* não está em execução, e compreende todas as técnicas baseadas em inspeções de código (Haug *et al.*, 2001; Peters e Pedrycz, 2001). As técnicas mais comuns são as inspeções de código, *walkthroughs* e testes de mesa, quando o código é analisado por uma equipe de pessoas ou uma única pessoa com o objetivo de encontrar erros. A análise estática ou teste estático, segundo Peters e Pedrycz (2001), divide-se em duas classes:

- *técnicas formais* – fazem parte das técnicas formais as provas de correção e execução simbólica; nas *provas de correção*, o código é comparado com a sua especificação, e, na *execução simbólica*, os valores numéricos são substituídos por símbolos, procurando-se simular a execução do programa no computador; e

- *técnicas informais* – são divididas em inspeções e revisões de código, onde uma pessoa ou uma equipe de pessoas analisa um projeto ou código.

Já os *testes dinâmicos* contemplam técnicas que objetivam analisar a estrutura e a funcionalidade de um *software* quando em execução (Haug *et al.*, 2001). São exemplos os testes funcionais, ou de caixa preta, e os testes estruturais, ou de caixa branca.

2.5.1 Teste Funcional (Caixa Preta) e Teste Estrutural (Caixa Branca)

Peters e Pedrycz (2001) definem dois grandes grupos usados como divisores das atividades de TS: os testes funcionais, ou de caixa preta, e os testes estruturais, ou de caixa branca – as duas abordagens mais documentadas sobre TS (Scott *et al.*, 2003). Em função de não ser objetivo deste trabalho uma discussão profunda sobre cada técnica ou método de TS existente na literatura, os mesmos são resumidamente descritos a seguir.

O *teste estrutural* ou *teste de caixa branca* objetiva testar a estrutura interna do *software* e desenvolver casos de teste baseados na estrutura lógica interna do *software* (Peters e Pedrycz, 2001), fazendo com que cada caminho lógico seja testado pelo menos uma vez (Jacobson *et al.*, 1997). Esses testes podem ser aplicados desde o início do processo de desenvolvimento. São exemplos: *teste de caminho básico* ou *abrangência de instrução*, *abrangência de ramificação/condição*, *abrangência de caminho*, *teste de caminho em grafos com loop* e *teste de estrutura de controle*. Pressman (1995) apresenta três variações de testes de estrutura de controle: *testes de condição*, *de fluxo de dados* e *de laços*.

O *teste funcional* ou *teste de caixa preta* atua como complemento ao anterior (Pressman, 1995). Por exemplo, nos testes finais junto ao usuário, os testes de caixa preta são utilizados para verificação da conformidade do que foi implementado aos respectivos requisitos funcionais (Paula Filho, 2001), ou quando não se dispõe do código para teste. Esses testes não se preocupam com a estrutura interna do *software*, mas com o objetivo principal de cada unidade – ou seja, com suas entradas e saídas (Jacobson *et al.*, 1997). Os testes de caixa preta tendem a ser aplicados nas etapas finais (Pressman, 1995), sendo exemplos: *particionamento de equivalência* (divisão dos dados de entrada em classes onde haja semelhança entre os dados utilizados para o caso de teste – Paula Filho, 2001; Pressman, 1995), *análise de valor-*

limite ou condição-limite (valores extremos – Pressman, 1995; Haug *et al.*, 2001), *teste exaustivo* (todas as entradas possíveis – Peters e Pedrycz, 2001), *teste de comparação* (Pressman, 1995), *teste orientado por sintaxe* (aplicável a sistemas cujas especificações são determinadas por uma determinada gramática – Peters e Pedrycz, 2001), *teste baseado em tabela de decisão* (Peters e Pedrycz, 2001) e *grafo de causa-efeito* (Pressman, 1995; Peters e Pedrycz, 2001).

2.6 Outros Tipos de Teste de *Software*

Peters e Pedrycz (2001) apresentam outros dois tipos de TS, para os quais não há um enquadramento claro nas categorias acima descritas. A literatura sobre TS é rica e proporciona variadas interpretações sobre classificações possíveis de técnicas ou métodos de TS. Uma mesma técnica pode ser classificada como dinâmica e estrutural, não invalidando a classificação de outra técnica que seja estrutural, mas esteja classificada como estática. Isto acarreta um problema adicional para se obter uma forma única de classificação das técnicas encontradas na literatura. Visto que o presente trabalho não pretende descrever todas as técnicas conhecidas – mas sim, verificar seu emprego na prática –, os autores ficarão restritos a citá-las sem a preocupação, deste ponto em diante, por uma classificação mais apurada e sistemática. Assim (cf. Peters e Pedrycz, 2001):

- *teste de desempenho* – avalia aspectos relativamente ao tempo de execução, tempo de resposta e utilização de dispositivos;
- *teste de fadiga* – visa a determinar pontos fortes e limitações do *software*; e
- *teste de regressão* – execução de alguns testes sempre que houver modificações.

Peters e Pedrycz (2001) apresentam outras técnicas de teste funcional e estrutural, igualmente abordadas por Pressman (1995), Jacobson *et al.* (1997), Haug *et al.* (2001), Myers (1979) e Beizer (1990), entre outros. O leitor interessado pode buscar uma síntese dos testes mais comuns na indústria de desenvolvimento de *software* em Scott *et al.* (2003), bem como encontrar uma comparação substantiva entre abordagens tradicionais (*planejadas*) e outras mais contemporâneas (*ágeis*) a TS em Pyhäjärvi e Rautiainen (2004).

3 Qualidade

A qualidade está na raiz das motivações para TS (Scott *et al.*, 2003) e é definida por Campos (1992) como sendo relativa a um produto ou serviço que trate de forma plena, segura e no tempo apropriado as necessidades do cliente. De modo equivalente, Schmauch (1994) diz que a qualidade de um produto é atingida quando se atendem as necessidades dos consumidores. Tais definições têm em comum a relação entre o produto/serviço oferecido e o produto/serviço percebido pelo cliente (De Campos e Campos, 2001). Uma maneira de a empresa alcançar qualidade em um produto ou serviço é por meio da qualidade do seu desenvolvimento (Schmauch, 1994) – e aqui o TS ganha destaque.

A qualidade pode ser buscada, por exemplo, com um sistema de qualidade proposto por uma série de padrões ISO. Um *sistema de qualidade* compreende procedimentos, processos, pessoas, ferramentas de gestão e instrumentos que contribuam para a qualidade de produtos e serviços, e a série de padrões ISO estabelece requisitos básicos de sistemas de qualidade para uma empresa superar as expectativas da clientela (Schmauch, 1994).

Uma forma mais específica para buscar qualidade em processos de *software* é dada pelos modelos CMM (Capability Maturity Model) formulados pelo Software Engineering Institute da Carnegie Mellon University (Pittsburgh, EUA). Esses modelos propõem níveis de maturidade para o enquadramento das empresas relativamente a aspectos envolvidos em projetos de *software*, como o desenvolvimento, a aquisição e a gestão dos profissionais envolvidos. Os cinco níveis de maturidade de uma empresa desenvolvedora de *software*, segundo a abordagem CMM original, são (cf. Paulk, 2000):

- *inicial* (nível 1) – há poucos processos definidos, e o sucesso depende basicamente de esforços individuais e “heróicos”;
- *repetível* (nível 2) – a execução de novos projetos é baseada na experiência com projetos antigos, havendo controles gerenciais básicos de *software*;
- *definido* (nível 3) – há padronização, documentação e integração dos processos de *software* da organização;
- *gerenciado* (nível 4) – medidas dos processos de *software* e qualidade dos produtos são coletadas no decorrer do projeto de *software*; e
- *otimizante* (nível 5) – contínuo aperfeiçoamento dos processos por toda a organização, com o objetivo de serem detectados e corrigidos problemas.

4 Estudo de Caso

Assim como Hetzel (1987) e Peters e Pedrycz (2001) afirmam que os procedimentos de TS são empregados na maioria das empresas de desenvolvimento de *software* de forma amadora, também se questiona sobre o assunto em se tratando dos pólos de informática brasileiros. Caso de especial interesse neste trabalho é representado por um desses pólos, localizado na Região Sul do país e já caracterizado como possuindo elementos de um legítimo *cluster* estratégico para o desenvolvimento regional (Pereira e Bellini, 2002; Bellini e Pedrozo, 2001a; Bellini e Pedrozo, 2001b); doravante, referir-se-á a esse grupo de empresas como *Pólo*.

Supõe-se que haja, no *Pólo*, uma preocupação elevada com TS, dada a sua condição de referência nacional em qualidade de produtos e serviços de informática. É de interesse dos autores, portanto, confrontar as recomendações teóricas sobre TS ao que efetivamente vem sendo praticado pelas empresas do *Pólo*, de modo que sejam geradas percepções fundamentais sobre a importância dos desenvolvimentos teóricos (*estado da arte*) comparativamente a práticas correntes (*estado da prática*) de expoentes da indústria. Deste modo, pode-se levantar questionamento seminal acerca de quão viáveis são os desenvolvimentos acadêmicos na área, bem como investigar se há práticas desconhecidas ou negligenciadas pelos estudos teóricos.

Os autores utilizaram o método de estudo de caso, dado que este viabiliza um conhecimento profundo da realidade de interesse (Palvia *et al.*, 2003; Hussey e Hussey, 1997; Benbasat *et al.*, 1987), não requer que todas as principais variáveis sejam conhecidas *a priori* (Boudreau *et al.*, 2001; Benbasat *et al.*, 1987) e aborda elementos complexos para investigação futura (Stake, 2000); todas essas características alinham-se ao formato da pesquisa proposta. O estudo de caso contemplou as seguintes atividades até o ano de 2003:

- levantamento bibliográfico sobre teste e qualidade de *software*;
- estudo preliminar das empresas que constituem o *Pólo*, separando-as em dois grupos;
- construção e validação (face e conteúdo) de instrumento de coleta de dados (roteiro de entrevistas) para entrevistas em profundidade sobre práticas de TS junto às empresas do *Pólo*;
- aplicação do instrumento a um ou dois profissionais de TS em cada empresa;
- análise de conteúdo (cf. Bardin, 1977) das entrevistas realizadas, observando-se critérios de (cf. Lima Filho apud Borges, 2000) *aglutinação* (reunião de itens semelhantes), *desmembramento* (separação de termos com mesma grafia/pronúncia e significado contextual

múltiplo), *determinância* (desconsideração de itens irrelevantes) e *representatividade* (atenção aos itens mais freqüentes e enfatizados pelo entrevistado); e

- cruzamento das entrevistas com o referencial teórico sobre TS, gerando-se percepções sobre o alinhamento entre a teoria e a prática.

4.1 Unidades de Análise

O processo de seleção das empresas para o estudo foi desenvolvido em pesquisa preliminar no *site* do Pólo, sendo coletadas informações sobre as empresas que constituem a incubadora, o condomínio e o parque tecnológico. Para corroborar as informações obtidas no *site*, um dos autores solicitou, junto à secretaria da unidade de desenvolvimento tecnológico do Pólo, a relação das suas empresas. Com isto, identificaram-se nove empresas na incubadora, uma no condomínio e dez no parque tecnológico, e um contato inicial com as mesmas foi então realizado para avaliar a possibilidade de sua participação no estudo.

Com o primeiro contato, constatou-se que oito das nove empresas da incubadora, nove das dez empresas do parque e a única empresa do condomínio possuíam alguma atividade relacionada a desenvolvimento de *software*, o que as tornava passíveis de investigação. Em um segundo momento, um dos autores novamente contactou as empresas pré-selecionadas para agendar entrevistas em profundidade (do agendamento à execução, o processo consumiu três semanas). No total, 14 empresas foram abordadas, pois duas empresas da incubadora não se dispuseram a participar e outras duas tinham dupla presença no Pólo.

As empresas foram classificadas em dois grupos, conforme o seu tempo de atuação no mercado. Este critério dividiu-as em empresas com um tempo de atuação superior a sete anos (*Grupo A*) e empresas com tempo de atuação inferior ou igual a isto (*Grupo B*). Relativamente ao critério adotado, o tempo de atuação, este não difere substancialmente da classificação feita pelo próprio Pólo. O Quadro 1 resume a classificação.

Grupo	Critério	Qtde. Empresas
A	atuação no mercado há mais de 7 anos	9
B	atuação no mercado há 7 anos ou menos	5

Quadro 1: composição dos grupos de empresas

4.2 Instrumentação para Coleta de Dados

Com a divisão das empresas do Pólo em dois grupos, os autores elaboraram um instrumento de coleta de dados (roteiro de entrevistas) formado por 29 questões, objetivando extrair informações sobre o entendimento dos entrevistados quanto ao Pólo, às propostas ISO e CMM, qualidade, e quanto a processo e teste de *software*. Para cada questão, associou-se uma variável a ser estudada em análise de conteúdo posterior, objetivando reduzir a possibilidade de perguntas ambíguas ou que abordassem mais de uma variável de interesse simultaneamente (o que, no entanto, poderia ocorrer dentro de cada resposta, dada a prerrogativa das entrevistas em profundidade de permitir ao entrevistado discorrer livremente sobre o tema em questão); devido a limitações de espaço, as variáveis e suas definições não serão aqui apresentadas. Adicionalmente às 29 perguntas, uma tabela com 38 procedimentos específicos de TS foi apresentada aos entrevistados para preenchimento relativamente ao conhecimento e aplicação de cada procedimento em sua empresa.

Desenvolvido o instrumento de coleta de dados, fez-se necessário validá-lo para haver maior credibilidade quanto às suposições de pesquisa (Straub, 1989). O instrumento foi submetido a dois processos de validação: face e conteúdo (cf. Hoppen *et al.*, 1996). Após as correções necessárias e precedendo a aplicação final, o instrumento passou por pré-teste junto a uma empresa de cada grupo, mostrando-se adequado aos propósitos de pesquisa.

4.3 Coleta de Dados

A seguir, apresentam-se variáveis selecionadas, que forneceram os principais resultados deste artigo, evitando-se, por motivos de espaço, a discussão de muitas situações particulares resolvidas durante a análise de dados – por exemplo, valores específicos dos indicadores envolvidos na mensuração de cada variável. Nos quadros abaixo, os números entre parênteses indicam a quantidade de respostas cujo conteúdo é mencionado.

4.3.1 Documentação de Desenvolvimento

Esta variável buscou saber do entrevistado como ele avaliava a documentação envolvida no desenvolvimento de *software* em sua empresa. Não se abordou em profundidade quão completa era a documentação relativamente ao que a literatura recomenda, devido à gama de informações necessárias para levantamento desta situação. O Quadro 2 compara os grupos de empresas neste quesito. Apesar de um dos entrevistados do Grupo B afirmar que em seus projetos novos a documentação era completa, optou-se por classificá-la como incompleta,

pois, para sistemas antigos (anteriores à adoção de uma política voltada à documentação), o entrevistado relatou não haver documentação alguma.

Grupo	Prática
A	completa (6); incompleta (3)
B	completa (2); incompleta (3)

Quadro 2: documentação de desenvolvimento

4.3.2 Documentação de Teste de *Software*

Buscando levantar a percepção do entrevistado a respeito da documentação adotada em TS, esta variável reuniu as impressões dispostas no Quadro 3. Três entrevistados do Grupo A e dois do Grupo B argumentaram que o motivo da falta de documentação estava relacionado a problemas de tempo.

Grupo	Prática
A	completa (4); incompleta (5)
B	completa (1); incompleta (4)

Quadro 3: documentação de TS

4.3.3 Referencial Teórico

A variável objetivou averiguar qual referencial teórico era utilizado pelo entrevistado para auxílio à aplicação de algum procedimento de TS em sua empresa. Quatro empresas do Grupo A e outras quatro do Grupo B indicaram não utilizar livros com informações acerca da aplicação de alguma prática de TS; contudo, alguns entrevistados utilizavam apostilas de atividades de formação de que participaram (Quadro 4).

Grupo	Prática
A	não (4) ; sim (1); palestras (3); experiência anterior (3); cursos (2); Internet (1); seminários (1)
B	não (4); palestras (1)

Quadro 4: referencial teórico de TS

4.3.4 Planejamento

A variável resgatou a existência, ou não, de planejamento (plano de teste) precedente ao TS (Quadro 5). Um dos entrevistados do Grupo A relatou que sua empresa possuía um planejamento completo, mas, quando inquirido relativamente ao tempo destinado aos testes, este não existia. Outro entrevistado do Grupo A confirmou a inexistência de planejamento, relatando: “hoje, o que temos é um cara heróico, em uma busca desenfreada por erros”; mas ressaltou haver interesse em adotar algum planejamento em curto prazo.

Relativamente às empresas do Grupo B, há uma incidência menor de planejamento para auxílio à aplicação de TS, e não foi possível fazer uma classificação satisfatória da resposta de um dos entrevistados deste grupo – desconsiderando-se, assim, a sua resposta.

Grupo	Prática
A	sim/completo (4); sim/incompleto (3); não (2)
B	sim/completo (1); sim/incompleto (2); não (1); sem resposta (1)

Quadro 5: planejamento de TS

4.3.5 Equipe

Esta variável objetivou averiguar a existência de equipes responsáveis pelas atividades de TS e a quantidade de pessoas que as formam. Inicialmente, um dos autores pretendia proceder a uma classificação comparativamente à divisão de equipes definida por Peters e Pedrycz (2001) para as revisões estruturadas e inspeções formais; mas isto não foi possível, pois não foram obtidas informações suficientes para a definição do papel de cada pessoa na equipe. Observou-se (Quadro 6) que todas as empresas do Grupo A possuíam equipes para TS (não necessariamente alocadas somente para esta função), enquanto nenhuma empresa do Grupo B as possuía.

Grupo	Prática
A	cinco pessoas (2); quatro pessoas (1); três pessoas (1); não se sabe quantas pessoas (1); por produto – cada produto requer um número diferente de pessoas (1)
B	desenvolvedor testa (2); desenvolvedor reveza com outro desenvolvedor (2); contrata-se testador (1)

Quadro 6: equipe de TS

4.3.6 Expectativa

A partir da definição de Myers (1979) de que o objetivo de TS é encontrar erros, buscou-se apreender o entendimento do entrevistado sobre o objetivo dos testes realizados em suas empresas. As respostas do Grupo B mostraram-se bastante interessantes no sentido de não haver um acordo sobre os objetivos de TS no desenvolvimento de *software*. Contudo, as empresas do Grupo B apresentaram um panorama positivo sobre esta definição, conforme indica o Quadro 7. Uma das empresas do Grupo A, que respondeu “encontrar erros”, complementou: “encontrar o que hoje é a nossa média de erros, que é de 75 erros a cada mil horas construídas, isso é o que eu espero encontrar (...) em cima disso é que eu planejo o tempo de reconstrução do código (...) se eu encontro mais que isso, nossa expectativa muda, porque muda a métrica para aquele projeto (...)”. Na percepção dos autores, isto demonstra um controle apurado dos processos envolvidos nos testes. Cabe ressaltar que dois critérios encontrados como valores para esta variável – cumprir prazos e não encontrar erros – são polêmicos, pois podem produzir resultados que não levam à qualidade do *software* (Myers, 1979).

Grupo	Prática
A	encontrar erros (6); não encontrar erros (1); alcançar funcionalidade total do <i>software</i> (1); cumprir prazos (1)
B	encontrar erros (1); cumprir prazos (1); validar requisitos (1); verificar funcionalidade (1); diminuir problemas no cliente (1)

Quadro 7: expectativa quanto a TS

4.3.7 Aproveitamento

Os resultados obtidos a partir de uma atividade de TS não devem ser desprezados, pois ajudam no estabelecimento de padrões de teste, histórico de erros mais comuns, auxílio na eliminação de erros de implementação de módulos semelhantes e – como relatado por um entrevistado – apresentação, ao cliente, da incidência de erros no *software* sendo adquirido, objetivando-se transmitir segurança sobre o produto final. De acordo com os relatos (Quadro 8), acredita-se ser um dos fatores determinantes para a inexistência de padrões de documentação em duas empresas do Grupo B e uma do Grupo A a falta de tempo para a organização desse material.

Grupo	Prática
A	armazenados e consultados por programadores (2) armazenados e consultados por programadores e <i>help desk</i> (1) armazenados em sistema próprio (1) armazenados em sistema próprio e utilizados; relatórios podem ser fornecidos ao cliente (1) armazenados (1) armazenados mas não utilizados; falta repositório de erros (1) armazenados e utilizados para melhoria da qualidade (1) não utilizados (1)
B	não armazenados (2) armazenados (3)

Quadro 8: aproveitamento de TS

4.3.8 Procedimentos Específicos

Conforme já mencionado, uma tabela com 38 procedimentos específicos de TS foi apresentada aos entrevistados para que indicassem o conhecimento e – caso positivo – o uso dos mesmos em suas empresas (Quadro 9). Eventuais conflitos de terminologia foram solucionados pelo entrevistador no momento das entrevistas, com apoio de material conceitual. Ao final do preenchimento da tabela, permitia-se ao entrevistado fazer observações livres com respeito a possíveis outros procedimentos não previstos na pesquisa.

Os resultados são extensos e estão disponíveis em maior detalhe junto aos autores. Para a análise dos procedimentos conhecidos e utilizados pelas empresas, adotou-se como critério de corte o entendimento empírico dos autores (profissionais do desenvolvimento de *software*) de que procedimentos citados por ao menos sete empresas do Grupo A e quatro do Grupo B constituiriam a situação desejável – dado o papel destacado do Pólo na indústria nacional de informática. Assim, níveis elevados de conhecimento e utilização de procedimentos de TS e níveis baixos de desconhecimento e inobservância dos mesmos eram esperados – supondo-se, é claro, que as recomendações teóricas devessem ser rigidamente implementadas em empresas de referência. O Quadro 9 apresenta os 38 procedimentos, destacando, em negrito, as situações desejáveis.

As classificações dos Quadros 10 e 11 indicam o conhecimento dos entrevistados sobre os procedimentos de TS sugeridos, enquanto os Quadros 12 e 13 apresentam a ocorrência desses procedimentos – tanto quanto relatada – nas empresas.

Procedimento	Grupo A				Grupo B			
	Conhece		Utiliza		Conhece		Utiliza	
	Sim	Não	Sim	Não	Sim	Não	Sim	Não
<i>checklist</i>	8	1	9	0	4	1	5	0
abordagem de grafo de causa-efeito	6	3	2	7	4	1	0	5
abrangência de caminho	7	2	3	6	2	3	2	3
abrangência de instrução	7	2	2	7	1	4	2	3
abrangência de ramificação	7	2	3	6	1	4	2	3
abrangência de ramificação/condição	7	2	3	6	2	3	3	2
análise de valor limite/condições limite	8	1	8	1	3	2	3	2
inspeções de código	6	3	5	4	4	1	3	2
particionamento de equivalência	3	6	2	7	1	4	1	4
revisões estruturadas ou de código	7	2	7	2	4	1	4	1
teste alfa	6	3	4	5	3	2	3	2
teste baseado em erros	8	1	8	1	4	1	5	0
teste baseado em tabela de decisão	6	3	4	5	3	2	1	4
teste beta	8	1	7	2	4	1	3	2
teste com base em máquinas de estado finito	2	7	0*	8*	1	4	1	4
teste de aceitação/validação	8	1	9	0	5	0	5	0
teste de caminho em grafos com <i>loop</i>	3	6	0	9	1	4	0	5
teste de conformidade de protocolo	4	5	7	2	2	3	1	4
teste de desempenho	8	1	9	0	5	0	5	0
teste de especificação	8	1	9	0	5	0	5	0
teste de estresse	8	1	7	2	5	0	3	2
teste de integração	8	1	9	0	5	0	5	0
teste de programas	8	1	9	0	5	0	5	0
teste de regressão	7	2	8	1	4	1	4	1
teste de segurança	7	2	7	2	4	1	3	2
teste de sistema	7	2	9	0	5	0	5	0
teste de unidade	8	1	9	0	4	1	4	1
teste de validação	8	1	9	0	5	0	5	0
teste estrutural	7	2	4	5	4	1	3	2
teste exaustivo	7	2	6	3	5	0	3	2
teste funcional/base na especificação	8	1	9	0	4	1	5	0
teste orientado por sintaxe	4	5	1	8	2	3	1	4
semeadura de erros	4	5	5	4	2	3	0	5
análise de mutantes	4	5	3	6	1	4	0	5
teste estático	7	2	8	1	3	2	1	4
teste dinâmico	7	2	8	1	3	2	3	2
estratégia de teste <i>top-down/bottom-up</i>	5	4	5	4	2	3	2	3
<i>walkthroughs</i>	5	4	5	4	2	3	1	4

* Uma empresa não respondeu.

Quadro 9: procedimentos de TS

Grupo A (≥ 7)	Grupo B (≥ 4)
<i>checklist</i>	
revisões estruturadas ou de código	
teste baseado em erros	
teste beta	
teste de aceitação/validação	
teste de desempenho	
teste de especificação	
teste de estresse	
teste de integração	
teste de programas	
teste de regressão	
teste de segurança	
teste de sistema	
teste de unidade	
teste de validação	
teste estrutural	
teste exaustivo	
teste funcional/base na especificação	
abrangência de caminho	
abrangência de instrução	
abrangência de ramificação	
abrangência de ramificação/condição	
análise de valor limite/condições limite	
teste estático	
teste dinâmico	
	abordagem de grafo de causa-efeito
	inspeções de código

Quadro 10: procedimentos de TS mais conhecidos pelas empresas

Grupo A (≥ 3)	Grupo B (≥ 2)
particionamento de equivalência	
teste alfa	
teste baseado em tabela de decisão	
teste de caminho em grafos com <i>loop</i>	
teste de conformidade de protocolo	
teste com base em máquinas de estado finito	
teste orientado por sintaxe	
semeadura de erros	
análise de mutantes	
estratégia de teste <i>top-down/bottom-up</i>	
<i>walkthroughs</i>	
abordagem de grafo de causa-efeito	
inspeções de código	
	abrangência de caminho
	abrangência de instrução
	abrangência de ramificação
	abrangência de ramificação/condição
	análise de valor limite/condições limite
	teste estático
	teste dinâmico

Quadro 11: procedimentos de TS menos conhecidos pelas empresas

Grupo A (≥ 7)	Grupo B (≥ 4)
<i>checklist</i>	
revisões estruturadas ou de código	
teste baseado em erros	
teste de aceitação/validação	
teste de desempenho	
teste de especificação	
teste de integração	
teste de programas	
teste de regressão	
teste de sistema	
teste de unidade	
teste de validação	
teste funcional/base na especificação	
análise de valor limite/condições limite	
teste beta	
teste de conformidade de protocolo	
teste de estresse	
teste de segurança	
teste estático	
teste dinâmico	

Quadro 12: procedimentos de TS mais utilizados pelas empresas

Grupo A (≥ 3)	Grupo B (≥ 2)
abordagem de grafo de causa-efeito	
abrangência de caminho	
abrangência de instrução	
abrangência de ramificação	
abrangência de ramificação/condição	
inspeções de código	
particionamento de equivalência	
teste alfa	
teste baseado em tabela de decisão	
teste com base em máquinas de estado finito	
teste de caminho em grafos com <i>loop</i>	
teste estrutural	
teste exaustivo	
teste orientado por sintaxe	
semeadura de erros	
análise de mutantes	
estratégia de teste <i>top-down/bottom-up</i>	
<i>walkthroughs</i>	
	análise de valor limite/condições limite
	teste beta
	teste de conformidade de protocolo
	teste de estresse
	teste de segurança
	teste estático
	teste dinâmico

Quadro 13: procedimentos de TS menos utilizados pelas empresas

5 Discussão

Primeiramente, é oportuno lembrar que não foi adotada a divisão de empresas oficialmente empregada pelo Pólo, havendo-se tratado as empresas como pertencentes a um grupo *A* (mais de sete anos de mercado) ou a um grupo *B* (até sete anos de atuação). Assim, relativamente ao objetivo proposto – estimar o nível de conformidade dos procedimentos de TS empregados pelas empresas do Pólo com as recomendações da literatura –, os autores acreditam haver chegado a conclusões diferentes para os dois grupos: as práticas do Grupo A parecem estar em conformidade com a literatura, enquanto as práticas do Grupo B não.

Na análise da documentação de desenvolvimento de *software*, observou-se que seis empresas do Grupo A apresentaram documentação completa. Como a documentação costuma receber pouca atenção por parte de desenvolvedores (Sanches, 2001), este aspecto é importante para caracterizar a preocupação das empresas com os procedimentos do processo de *software*. A menor atenção dessas mesmas empresas (Grupo A) quanto à documentação específica de TS, porém, é curiosa e precisa ser mais bem investigada, pois está em conflito com a documentação do processo como um todo e com o relato de que os testes são aproveitados posteriormente.

Outro aspecto importante sobre o Grupo A é o fato de este registrar uma busca efetiva por referencial teórico a respeito de TS, demonstrando interesse em expandir seus conhecimentos. De fato, este grupo apresenta uma noção conceitual sobre TS superior à do Grupo B no que diz respeito às definições de Myers (1979), Hetzel (1987) e Haug *et al.* (2001). E, por fim, o Grupo A também parece mais interessado que o Grupo B em planejar suas atividades de TS – talvez por possuir uma visão mais completa sobre o assunto – e utilizar posteriormente o aprendizado com as suas experiências. Tal comportamento está em conformidade com recomendações de Myers (1979) para fins de melhorias em projetos e processos futuros.

As empresas do Grupo B, por sua vez, compõem um cenário oposto: (1) a documentação do desenvolvimento de *software*, e de TS em particular, é usualmente incompleta; (2) não há utilização efetiva de referencial teórico (embora este talvez não seja fator determinante para a qualidade do processo); (3) o planejamento de TS, quando ocorre, costuma ser incompleto; e (4) o aproveitamento do histórico de TS não parece receber a atenção necessária (apesar de as empresas do Grupo B apresentarem documentação de resultados, elas, tomadas em conjunto, não relatam de forma precisa a utilização posterior dos registros).

Quanto aos 38 procedimentos específicos de TS discriminados no Quadro 9, o presente artigo restringe sua análise à verificação de que as empresas do Grupo A – como seria esperado após a discussão precedente – conhecem e aplicam mais procedimentos de TS do que as empresas do Grupo B; adicionalmente, nenhuma prática de TS do Grupo B deixa de ser efetiva nos processos de *software* do Grupo A como um todo – e o inverso não é verdadeiro. Além de ratificar um mais avançado alinhamento do Grupo A às proposições da literatura, pode-se supor que isto se deva ao próprio empenho do grupo em conhecer as fontes teóricas; os autores não ignoram, porém, que o conhecimento presumivelmente superior da literatura pelo Grupo A pode haver facilitado o preenchimento da tabela de práticas de TS por este grupo, e exercido efeito contrário junto ao Grupo B.

Outras análises possíveis a partir dos Quadros 10, 11, 12 e 13 dão conta de objetivos específicos de determinados grupos de procedimentos de TS, mas, por motivos de espaço e abrangência, não serão aqui discutidas. Por exemplo, evidências sobre a aplicação ou a inobservância de procedimentos como inspeções de código, revisões estruturadas e *walkthroughs* podem lançar luz sobre a presença e a efetividade de equipes formais de TS em cada grupo de empresas – e enriquecer a discussão sobre o alinhamento entre a prática e a teoria. Tal empreendimento encontra-se em elaboração e será divulgado oportunamente.

O fato de se haver concluído sobre inconformidade do Grupo B às recomendações teóricas de TS priorizadas neste estudo não necessariamente implica que suas empresas ignoram ou desconsideram práticas para um desenvolvimento adequado de sistemas. De fato, os procedimentos de TS investigados nas entrevistas representam escolhas arbitrárias dos autores – embora dirigidas por experiência profissional e abrangente revisão da literatura. Fatores como de viés nas escolhas teóricas, ruído de comunicação durante as entrevistas, assimetria de conhecimento entre pesquisadores e profissionais entrevistados e estratégias de divulgação de práticas de TS por estes últimos podem haver exercido influência sobre as conclusões formuladas. Do mesmo modo, em sendo o Pólo um dos expoentes da indústria brasileira de informática em termos de qualidade de produtos e serviços, também se pode imaginar que a inobservância de práticas consagradas por algumas de suas empresas talvez indique, inclusive, inadequação das próprias recomendações teóricas – mas o presente estudo não coletou evidências de que seja este o caso. Por fim, os autores ressaltam que a distância entre uma determinada recomendação teórica e a sua prática na indústria pode ser plenamente devida a idiossincrasias de aplicação.

5.1 Limitações

Estão relacionadas, abaixo, algumas limitações importantes desta pesquisa:

- acesso a apenas um ou dois entrevistados em cada empresa;
- recusa de duas empresas em participarem da pesquisa;
- alto grau de subjetividade inerente a estudos de caso apoiados por entrevistas em profundidade, não obstante o rigor de aplicação perseguido em todas as etapas metodológicas;
- impossibilidade de haverem sido observadas diretamente as atividades de TS nas empresas; e
- priorização de determinados procedimentos de TS nas entrevistas.

5.2 Sugestões para Pesquisas Futuras

Os autores julgam interessantes as seguintes linhas de estudo afins ao presente trabalho:

- levantar indicadores de qualidade de *software* nas empresas do Pólo, verificando-se possíveis efeitos das atividades de TS – realizadas ou não pelas empresas – sobre produtos e serviços (cf. Huq, 2000);
- realizar pesquisas com maior envolvimento do pesquisador com as atividades de TS mencionadas como usuais nas empresas, a fim de ser verificado o grau de realismo das percepções coletadas nas entrevistas;
- verificar a real importância de TS nos processos das empresas do Pólo relativamente às suas atividades de mercado (desenvolvimento de produtos padronizados/personalizados, implantação de sistemas, serviços de consultoria e treinamento, etc.);
- realizar estudos semelhantes junto a outros grupos de empresas de informática no Brasil, de modo a serem comparados resultados de pesquisa e de modo mais preciso se estimar a conformidade da indústria brasileira como um todo com as recomendações teóricas sobre TS; e
- elaborar conceitualmente a relação entre TS e alinhamento estratégico de sistemas de informação.

Referências

- BARDIN, L. *Análise de conteúdo*. Lisboa: Edições 70, 1977.
- BEIZER, B. *Software testing techniques*. Scottsdale: Coriolis Group, 1990.
- BELLINI, C.G.P.; PEDROZO, E.A. Cluster formation in Southern Brazil: information technology driving regional development. *10th International Conference on Management of Technology*. Lausanne: IAMOT, 2001a.
- BELLINI, C.G.P.; PEDROZO, E.A. Competitividade em tecnologia da informação: perspectivas de um cluster brasileiro. *II Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento e III Seminário Catarinense de Gestão do Conhecimento e da Tecnologia*. Florianópolis: IEL, 2001b.
- BENBASAT, I.; GOLDSTEIN, D.K.; MEAD, M. The case research strategy in studies of information systems. *MIS Quarterly*, 11(3), 1987, pp. 369-386.
- BORGES, G. *Comércio eletrônico: atributos relevantes no processo de decisão de compra*. Dissertação (Mestrado em Administração). Porto Alegre: PPGA/EA/UFRGS, 2000.
- BOUDREAU, M.; GEFEN, D.; STRAUB, D.W. Validation in information systems research: a state-of-the-art assessment. *MIS Quarterly*, 25(1), 2001, pp. 1-16.
- CAMPOS, V.F. *TQC: controle de qualidade total (no estilo japonês)*. Belo Horizonte: Bloch, 1992.
- CHAN, Y.E.; HUFF, S.L.; BARCLAY, D.W.; COPELAND, D.G. Business strategic orientation, information systems strategic orientation, and strategic alignment. *Information Systems Research*, 8(2), 1997, pp. 125-150.
- DE CAMPOS, G.H.B.; CAMPOS, F.C.A. *Qualidade para domínios de aplicação e tecnologias de desenvolvimento específicas – qualidade do software educacional*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001, pp. 124-130.
- GIMENEZ, I. *Processos do ciclo de vida: processos fundamentais – processo de desenvolvimento*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001, pp. 43-47.
- HAUG, M.; OLSEN, E.W.; CONSOLINI, L. *Software quality approaches: testing, verification, and validation: software best practice I*. Berlin: Springer, 2001.
- HETZEL, W. *Guia completo ao teste de software*. Rio de Janeiro: Campus, 1987.
- HOPPEN, N.; LAPOINTE, L.; MOREAU, E. Um guia para a avaliação de artigos de pesquisa em sistemas de informação. *Revista Eletrônica de Administração*, 2(2), 1996.

- HUQ, F. Testing in the software development life-cycle: now or later. *International Journal of Project Management*, 18, 2000, pp. 243-250.
- HUSSEY, J.; HUSSEY, R. *Business research*. New York: Palgrave, 1997.
- JACOBSON, I.; GRISS, M.; JONSSON, P. *Software reuse: architecture process and organization for business success*. Harlow: Addison-Wesley, 1997.
- LEITE, J.C.S.P. *Gerenciando a qualidade de software com base em requisitos*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001, pp. 238-246.
- MACHADO, C.A.F. *Normas e modelos de maturidade – NBR ISO/IEC 12207, processos de ciclo de vida de software*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001, pp. 9-16.
- MALDONADO, J.C.; FABBRI, S.C.P.F. *Processos do ciclo de vida: processos de apoio – verificação e validação de software*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001a, pp. 66-73.
- MALDONADO, J.C.; FABBRI, S.C.P.F. *Processos do ciclo de vida: processos de apoio – teste de software*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001b, pp. 73-84.
- MALDONADO, J.C.; SANCHES, R.; FABBRI, S.C.P.F. *Processo de software*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001, pp. 5-7.
- MYERS, G.J. *The art of software testing*. New York: John Wiley & Sons, 1979.
- PALVIA, P.; MAO, E.; SALAM, A.F. *et al.* Management information systems research: what's there in a methodology? *Communications of the AIS*, 11, 2003, pp. 289-309.
- PAULA FILHO, W.P. *Engenharia de software: fundamentos, métodos e padrões*. Rio de Janeiro: LTC, 2001.
- PAULK, M.C. (Ed.). *The Capability Maturity Model: guidelines for improving the software process*. Reading: Addison-Wesley, 2000.
- PEREIRA, R.C.F.; BELLINI, C.G.P. Marketing de relacionamento no desenvolvimento de software personalizado. *XXXVII Assembléia do Conselho Latino-Americano de Escolas de Administração*. Porto Alegre: CLADEA, 2002.
- PETERS, J.F.; PEDRYCZ, W. *Engenharia de software: teoria e prática*. Rio de Janeiro: Campus, 2001.

- PRESSMAN, R.S. *Engenharia de software*. São Paulo: Makron Books, 1995.
- PYHÄJÄRVI, M.; RAUTIAINEN, K. Integrating testing and implementation into development. *Engineering Management Journal*, 16(1), 2004, pp. 33-39.
- RIOS, E. Gerência de teste num projeto de software. *XIII Congresso Internacional de Tecnologia de Software*. Curitiba: CITS, 2002.
- SANCHES, R. *Processos do ciclo de vida: processos de apoio – documentação de software*. In: DA ROCHA, A.R.C.; MALDONADO, J.C.; WEBER, K.C. (Orgs.). *Qualidade de software: teoria e prática*. São Paulo: Prentice Hall, 2001, pp. 54-59.
- SCHMAUCH, C.H. *ISO 9000 for software developers*. Milwaukee: ASQC Quality Press, 1994.
- SCOTT, E.; ZADIROV, A.; FEINBERG, S.; JAYAKODY, R. The alignment of software testing skills of IS students with industry practices – a South African perspective. *Informing Science + IT Education Joint Conference*. Pori: ISI, 2003, pp. 957-967.
- STAKE, R.E. *Case studies*. In: DENZIN, N.K.; LINCOLN, Y.S. (Eds.). *Handbook of qualitative research*. Sage, 2000.
- STAMELOS, I.; ANGELIS, L.; MORISIO, M.; SAKELLARIS, E.; BLERIS, G.L. Estimating the development cost of custom software. *Information & Management*, 40(8), 2003, pp. 729-741.
- STRAUB, D. Validating instruments in MIS research. *MIS Quarterly*, 13(2), 1989, pp. 147-169.
- WOODROOF, J.; BURG, W. Satisfaction/dissatisfaction: are users predisposed? *Information & Management*, 40, 2003, pp. 317-324.
- ZVIRAN, M.; ERLICH, Z. Measuring user satisfaction: review and implications. *Communications of the AIS*, 12, 2003, pp. 81-103.