

RELATÓRIO TÉCNICO

RT – ES 687/06

Estudos Primários e Secundários apoando a busca por Evidência em Engenharia de Software

Sômulo Nogueira Mafra

(somulo@cos.ufrj.br)

Guilherme Horta Travassos

(ght@cos.ufrj.br)



Programa de Engenharia de Sistemas e Computação

COPPE/UFRJ

Rio de Janeiro, Março de 2006

Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software

Neste relatório, é descrito como a Engenharia de Software pode se beneficiar da utilização de uma abordagem baseada em evidência. Nesse cenário, a condução de estudos experimentais, e a posterior compilação, agregação e sumarização dos resultados desses estudos, possibilita caracterizar uma determinada tecnologia em uso. O resultado dessa caracterização pode auxiliar a aquisição e a aplicação de tecnologias no desenvolvimento industrial de software, além de direcionar os esforços de pesquisa na área.

1. Introdução

Durante os últimos 20 anos, sistemas de software têm exercido um papel essencial e crítico em nossa sociedade (MAFRA, 2004). Nós dependemos das características e serviços oferecidos por sistemas computadorizados. Se alguns sistemas de uso global deixarem de funcionar, aproximadamente 40% da população mundial sofrerão as conseqüências desse problema (REED, 2000).

No entanto, em despeito dessa importância estratégica exercida pelo software, de acordo com Alan Davis (*apud* GLASS 2002), a atual indústria de software encontra-se no mesmo estado da arte que a indústria farmacêutica esteve durante o século XIX. A argumentação de Davis sustenta-se na percepção de que freqüentemente a indústria de software é acometida por diversos anúncios de “cura” para os mais variados problemas, assim como ocorria na indústria farmacêutica de dois séculos atrás.

Em vista desse quadro de relativa imaturidade, engenheiros de software freqüentemente são confrontados com as seguintes questões, quando da adoção de tecnologias:

- Em qual tecnologia investir, quando todas elas prometem aprimorar a produtividade e a qualidade no desenvolvimento (SHULL, 1998)?
- Como saber o custo de implantação de uma determinada tecnologia?
- Como saber o retorno de investimento proporcionado pela implantação de tal tecnologia?
- Sob quais circunstâncias a adoção de tal tecnologia pode ser recomendada?

Devido à carência de fontes confiáveis que possibilitem responder de forma satisfatória a essas questões, freqüentemente a decisão equivocada é tomada, e engenheiros de software acabam investindo tempo e dinheiro em tecnologias que falham em melhorar a qualidade no desenvolvimento de software (SHULL, 1998).

Entretanto, as respostas para essas questões poderiam ser obtidas de forma razoável caso a Engenharia de Software fizesse um uso intenso e sistemático de uma abordagem baseada em evidência.

Resultados positivos obtidos em outras disciplinas reforçam essa hipótese. Segundo KITCHENHAM *et al.* (2004a), a pesquisa em Medicina avançou consideravelmente durante a última década, como resultado da adoção de abordagens baseadas em evidência. SACKETT *et al.* (2000) apontaram que a quantidade de artigos médicos sobre práticas baseadas em evidência cresceu de uma publicação em 1992 para aproximadamente mil em 1998.

Além disso, o interesse internacional despertado resultou no desenvolvimento de seis revistas médicas especializadas no assunto. Como consequência, o sucesso obtido na Medicina levou outras disciplinas a adotarem abordagens semelhantes, incluindo Psiquiatria¹, Enfermagem², Política Social³ e Educação⁴.

Porém, não obstante o sucesso alcançado em outras disciplinas, é importante destacar-se porque a adoção de uma abordagem baseada em evidência pode ser benéfica para a Engenharia de Software. Nesse sentido, a necessidade de adoção de uma abordagem baseada em evidência na Engenharia de Software pode ser discutida sob os pontos de vista de seus principais *stakeholders*, entre eles:

- **Organização de Desenvolvimento de Software.** Necessita da pesquisa e do desenvolvimento de tecnologias que lhe permita aumentar a qualidade de seus produtos e diminuir custos, de forma a aumentar sua margem de lucro e alcançar ou sustentar uma vantagem competitiva.
- **Engenheiro de Software.** Exige tecnologias que facilitem seu trabalho possibilitando o aumento de sua produtividade.
- **Pesquisador.** Necessita de um acúmulo de evidência experimental que lhe permita o direcionamento de esforços em tópicos de pesquisa pertinentes à indústria ou a outros grupos interessados.

¹ www.med.nagoya-cu.ac.jp/psych.dir/ebpcenter.htm, acessado em 04/01/2006.

² www.york.ac.uk/healthsciences/centres/evidence/cebn.htm, acessado em 04/01/2006.

³ www.evidencenetwork.org, acessado em 04/01/2006.

⁴ cem.dur.ac.uk/ebeuk/EBEN.htm, acessado em 04/01/2006.

- **Usuário de Software.** Almeja produtos de qualidade que satisfaçam suas necessidades diárias a um custo razoável de aquisição e de operação.
- **Público Geral.** Tem o direito de esperar que seus governantes administrem de forma responsável a verba pública utilizada para desenvolver e manter sistemas de software, e ponham em prática controles para minimizar os riscos de tais sistemas causarem danos significativos à população (KITCHENHAM *et al.*, 2004a).

No contexto da Engenharia de Software, a abordagem baseada em evidência permitiria a caracterização de uma determinada tecnologia em uso. Através dessa caracterização seria possível determinar com níveis razoáveis de segurança o que funciona e o que não funciona sob quais circunstâncias.

Nesse sentido, para atender a essa finalidade, a Engenharia de Software Baseada em Evidência deve prover meios pelos quais melhores evidências provenientes da pesquisa possam ser integradas com experiência prática e valores humanos no processo de tomada de decisão considerando o desenvolvimento e a manutenção do software, conforme definição formulada por KITCHENHAM *et al.* (2004a). Como resultado:

- 1) Organizações de desenvolvimento de software seriam contempladas com um embasamento científico que apoiasse a tomada de decisão no que se refere à aquisição de novas tecnologias alinhadas às suas necessidades. Dessa forma, engenheiros de software poderiam ter à sua disposição tecnologias que contribuíssem para o aumento de produtividade, resultando em eventuais diminuições de custos de desenvolvimento e melhorias de qualidade nos produtos.
- 2) Usuários de software poderiam ter à sua disposição produtos de alta qualidade, desenvolvidos com tecnologias adequadas, que atendessem às suas necessidades. De forma similar, o público em geral poderia ser beneficiado com o funcionamento de produtos confiáveis que não representassem eventuais riscos à sua segurança e bem-estar, nem desperdício de dinheiro público por parte dos governantes.

Em vista disso, para atingir um nível adequado de evidência a respeito da caracterização de uma determinada tecnologia em uso, a Engenharia de Software Baseada em Evidência deve fazer uso basicamente de dois tipos de estudos: estudos primários e estudos secundários.

Por estudos primários, entende-se a condução de estudos que visem a caracterizar uma determinada tecnologia em uso dentro de um contexto específico.

Nessa categoria encontram-se os estudos experimentais, entre os quais experimentos, estudos de caso e surveys (WÖHLIN *et al.*, 2000).

Por estudos secundários, entende-se a condução de estudos que visem a identificar, avaliar e interpretar todos os resultados relevantes a um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa (KITCHENHAM, 2004b). Revisões sistemáticas (KITCHENHAM, 2004b), (BIOLCHINI *et al.*, 2005) são tipos de estudos secundários.

Nesse sentido, resultados obtidos por diversos estudos primários correlatos atuam como fonte de informação a ser investigada por estudos secundários. Além disso, conforme apontado por BIOLCHINI *et al.* (2005), vale ressaltar que estudos secundários não podem ser considerados uma abordagem alternativa para a produção primária de evidências, representada pelos estudos primários; estudos secundários são complementares a estudos primários. A precisão e a confiabilidade proporcionadas pelos estudos secundários contribuem para a melhoria e para o direcionamento de novos tópicos de pesquisa, a serem investigados por estudos primários, num ciclo iterativo.

Nas próximas seções são discutidas as principais abordagens no que se referem a estudos primários e a estudos secundários.

2. Estudos Primários

O sucesso do desenvolvimento de software depende da interação entre diversas variáveis, como ambiente de trabalho agradável, experiência de pessoal, utilização de processos e procedimentos de qualidade, e apoio ferramental adequado. Segundo KITCHENHAM *et al.* (2004a), a maior parte das técnicas em Engenharia de Software exerce impacto direto apenas em uma parte específica do ciclo de vida de desenvolvimento. Como consequência, torna-se difícil o isolamento do efeito individual de uma determinada tecnologia sobre a qualidade final do produto. Por exemplo, o sucesso da utilização de uma dada técnica de modelagem de projeto dependerá consideravelmente da qualidade da análise de requisitos realizada anteriormente, além das restrições impostas por plataformas de software e hardware utilizadas, linguagens de programação, cronograma e orçamento do projeto.

Em vista desse cenário de diversas variáveis presentes no desenvolvimento de software, e da complexidade no relacionamento entre essas variáveis, uma importante questão a ser levantada é: como identificar, isolar e avaliar a contribuição individual da aplicação de uma determinada tecnologia para a qualidade final do produto num cenário de desenvolvimento industrial de software?

A utilização de experimentação em Engenharia de Software pode atender satisfatoriamente a essa necessidade, à medida que fornece mecanismos adequados para a identificação e o entendimento do relacionamento entre as diferentes variáveis envolvidas num determinado contexto. Como consequência dessa investigação experimental, o entendimento de pesquisadores a respeito da Engenharia de Software pode ser aprimorado.

Além disso, o entendimento satisfatório de um determinado problema é um pré-requisito necessário para a busca de oportunidades de melhoria. Por essa razão, novas tecnologias e sugestões não devem ser apenas sugeridas, publicadas ou apresentadas para a venda, mas pelo menos devem ser comparadas com as tecnologias já existentes (TRAVASSOS *et al.* 2002).

A utilização de experimentação em Engenharia de Software justifica-se também pela necessidade de se avaliar aspectos relacionados ao comportamento humano no desenvolvimento de software. Segundo WÖHLIN *et al.* (2000), o desenvolvimento de software é fortemente dependente da criatividade e também da ingenuidade de pessoas trabalhando na área. Como exemplo, estudos experimentais têm tradicionalmente sido utilizados em Ciências Sociais e Psicologia envolvendo, sobretudo, o comportamento humano (APA, 2005), (PLESS, 2005), (XLAB, 2005).

2.1 Histórico de Experimentação em Engenharia de Software

A discussão pela necessidade de experimentação em Engenharia de Software não é recente. Essa necessidade foi discutida pela primeira vez em meados da década de 1980 (BASILI *et al.*, 1986). Desde então, diversos trabalhos têm sido publicados ao longo dos anos a respeito de experimentação em Engenharia de Software (POTTS, 1993), (FENTON *et al.*, 1994), (GLASS, 1994), (KITCHENHAM *et al.*, 1995), (BASILI, 1996), (TICHY, 1998), (WÖHLIN *et al.*, 2000), (JURISTO e MORENO, 2002).

Entretanto, apesar de todos os esforços, a pesquisa em Engenharia de Software ainda carece de evidência experimental conforme observado em TICHY *et al.* (1995), ZELKOWITZ e WALLACE (1998), PFLEEGER (1999) e KITCHENHAM *et al.* (2004a). Na próxima subseção são discutidos os principais métodos para a investigação experimental.

2.2 Métodos de Investigação Experimental

No que diz respeito à condução de estudos experimentais, dependendo do propósito da avaliação, e das condições para a condução do estudo, três tipos de investigação podem ser conduzidos (WÖHLIN *et al.* 2000):

- **Survey.** Um survey é freqüentemente uma investigação realizada em retrospecto, quando, por exemplo, uma determinada tecnologia tem sido utilizada durante um certo período de tempo (PFLEEGER, 1994). Dessa forma, a condução de um survey permitiria capturar um “retrato instantâneo” da atual situação. Os principais meios utilizados para a coleta de dados são entrevistas ou questionários. A coleta de dados é realizada utilizando-se uma amostra representativa da população sob estudo. Os resultados do survey são então analisados de forma a extrair-se conclusões que possam ser generalizadas à população da qual a amostra foi tomada. Surveys são discutidos em ROBSON (1993) e BABBIE (1990).
- **Estudo de Caso.** São estudos conduzidos com o propósito de se investigar uma entidade ou um fenômeno dentro de um espaço de tempo específico. Estudos de caso são usados principalmente para a monitoração de atributos presentes em projetos, atividades ou atribuições. Durante a condução de um estudo de caso, dados são coletados e, baseado na coleta desses dados, análises estatísticas são conduzidas de forma a permitir-se avaliar um determinado atributo ou o relacionamento entre diferentes atributos. Uma vantagem na condução de estudos de caso é a facilidade de planejamento. Entretanto, uma desvantagem é a dificuldade em generalizar-se os resultados obtidos. Geralmente, os resultados de um estudo de caso apontam os efeitos em uma situação em particular, não podendo ser generalizados para cada situação presente em outros contextos. Estudos de caso são discutidos em ROBSON (1993), STAKE (1995), PFLEEGER (1994), YIN (1994) e WÖHLIN *et al.* (2000).
- **Experimento.** Experimentos são conduzidos quando deseja-se obter um maior controle da situação, ao manipular-se as variáveis envolvidas no estudo de forma direta, sistemática e precisa. Experimentos são normalmente conduzidos em ambientes de laboratório, os quais proporcionam um nível relativamente alto de controle sobre as variáveis envolvidas no estudo. Em um experimento, os participantes são atribuídos a diferentes tratamentos de forma aleatória. O objetivo é manipular uma ou

mais variáveis e controlar todas as outras variáveis num valor fixo. O efeito da manipulação das variáveis é então medido e, baseado nessa medição, análises estatísticas são conduzidas. A condução de experimentos reais é rara em Engenharia de Software, devido à dificuldade de se alocar os participantes do estudo a diferentes tratamentos de forma aleatória. Nessas situações, tais estudos denominam-se quasi-experimentos.

A principal diferença entre um estudo de caso e um experimento refere-se ao nível de controle exercido sobre as variáveis do estudo. Num estudo de caso, o nível de controle é menor do que num experimento. Por isso, freqüentemente um estudo de caso caracteriza-se por ser um estudo de observação, enquanto um experimento caracteriza-se por ser um estudo controlado (ZELKOWITZ e WALLACE, 1998).

Segundo TRAVASSOS *et al.* (2002), independente do método de investigação utilizado, qualquer estudo experimental presume um relacionamento de causa (representado por tratamentos) e efeito (representado por resultados). Além disso, os conceitos envolvidos num estudo experimental incluem variáveis dependentes e independentes, objeto(s), participantes, contexto, hipóteses nula e alternativa(s), e o projeto do estudo.

Além disso, para um estudo experimental ter uma importância científica ou industrial, seus resultados devem ser obrigatoriamente válidos. Isso significa que os resultados de um estudo devem ser considerados segundo quatro tipos de validade: de conclusão, interna, externa, e de construção, que apresentam respectivamente, os aspectos a respeito da análise estatística, o relacionamento tratamento-resultado, a generalização dos resultados a uma população maior, e a relação entre a teoria e a observação (COOK e CAMPBELL, 1979).

Uma observação importante, é que os diferentes métodos de investigação não são ortogonais entre si. Um estudo experimental completo deve ser realizado considerando todos os métodos descritos. Segundo WÖHLIN *et al.* (2000), um estudo experimental completo poderia, por exemplo, adquirir a informação inicial utilizando um survey, elaborar uma teoria através de um experimento controlado e verificar a teoria proposta em condições reais por meio de um estudo de caso.

Na próxima subseção é descrito um processo de experimentação definido para a Engenharia de Software.

2.3 Processo de Experimentação em Engenharia de Software

No que diz respeito à condução de estudos experimentais, WÖHLIN *et al.* (2000) definiram um processo composto por cinco principais etapas: Definição, Planejamento,

Execução, Análise e Interpretação, e Apresentação e Empacotamento. AMARAL (2003) estendeu esse processo de experimentação substituindo a etapa de Apresentação e Empacotamento por um processo de empacotamento conduzido em paralelo ao processo de experimentação, conforme ilustrado na Figura 1.

No processo de experimentação definido em WÖHLIN *et al.* (2000), e estendido em AMARAL (2003), a etapa de Definição é a primeira atividade, onde o estudo experimental é expresso em termos dos problemas e objetivos. Durante a etapa de Planejamento, o projeto do estudo é determinado, a instrumentação a ser utilizada é definida, e os aspectos da validade dos resultados são analisados, resultando na elaboração do Plano do Estudo Experimental.

O Plano de Estudo Experimental desempenha um papel crucial no contexto de um estudo. A elaboração do plano deve contemplar aspectos que venham a minimizar a probabilidade de ocorrência de riscos durante a etapa de Execução. Dependendo da magnitude dos riscos, há uma possibilidade considerável do estudo ser invalidado.

Dessa forma, o processo de experimentação contém um ponto de controle (explicitado pela figura do 1° losango contendo um símbolo de interrogação na Figura 1) onde o plano do estudo deve ser avaliado e, dependendo da decisão tomada, opta-se pelo re-planejamento ou pelo prosseguimento do estudo.

Em alguns casos, o plano do estudo deve ser revisado, de preferência por pesquisadores que não possuam interesse direto nos resultados do estudo, para minimizar a presença de viés; em outras situações, onde a consequência da ocorrência de riscos é mais significativa, a condução de um estudo piloto é recomendada.

Uma vez aprovado o Plano do Estudo Experimental, e decidida pela condução do estudo, a etapa de Execução segue conforme o planejado. Durante essa etapa, os dados experimentais são coletados, de forma a serem analisados e avaliados durante a etapa de Análise e Interpretação. A etapa de Análise e Interpretação pode ser re-executada, caso haja necessidade de se avaliar os dados coletados durante a etapa de Execução de uma forma não prevista inicialmente durante o Planejamento.

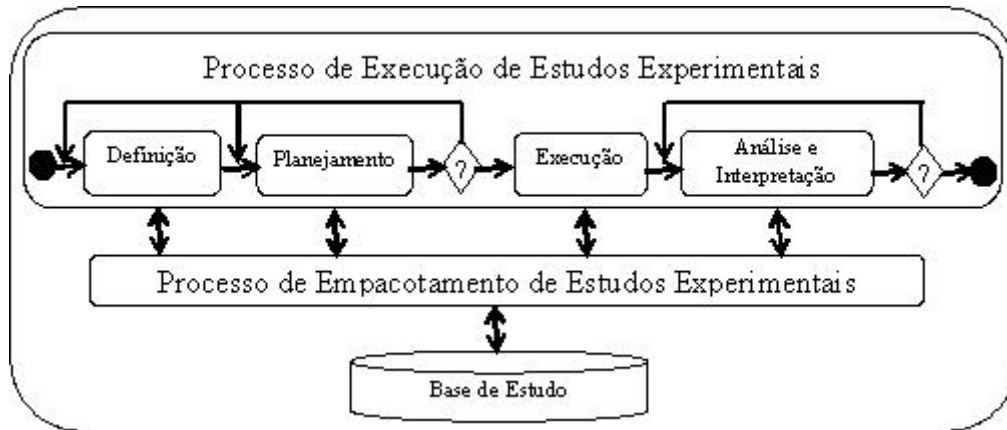


Figura 1 - Processo de Experimentação, definido em WÖHLIN et al. (2000) e estendido por AMARAL (2003).

Uma vantagem da evolução proposta por AMARAL (2003) refere-se à possibilidade de se capturar lições aprendidas ainda durante as fases iniciais do estudo. Dessa forma, essas lições aprendidas poderiam ser documentadas logo no início do estudo, evitando-se eventuais perdas de informações, caso fossem documentadas somente ao final do processo de experimentação, conforme a proposta original de WÖHLIN *et al.* (2000).

Na próxima seção, é discutida a importância da condução de estudos secundários, representados por revisões sistemáticas, para a busca pela evidência em Engenharia de Software.

3. Estudos Secundários

Apesar de ser uma condição requerida para a caracterização adequada de uma determinada tecnologia, somente a condução de estudos experimentais não é suficiente para esse propósito.

Segundo SHULL *et al.* (2004), nenhum estudo sobre uma determinada tecnologia pode ser considerado definitivo. Os resultados de qualquer estudo não podem simplesmente ser generalizados para todos os ambientes, uma vez que pode haver fontes de variação entre diferentes ambientes. Dessa forma, estudos experimentais devem ser repetidos em diferentes contextos, com o objetivo de se obter resultados mais precisos e confiáveis a respeito de uma tecnologia em uso.

Nesse sentido, para integrar-se os resultados provenientes de diversos estudos experimentais correlatos, estudos secundários devem ser conduzidos. Na próxima subseção, destacamos as principais diferenças entre revisões de literatura informais, tradicionalmente utilizadas em Engenharia de Software, e revisões sistemáticas.

3.1 Revisões Sistemáticas versus Revisões Informais

A ciência é uma atividade cooperativa e social, e o conhecimento científico é o resultado do processo cumulativo dessa cooperação (BIOLCHINI *et al.*, 2005). Nesse sentido, a revisão de literatura é o meio pelo qual o pesquisador pode identificar o conhecimento científico existente em uma determinada área, de forma a planejar sua pesquisa, evitando a duplicação de esforços e a repetição de erros passados.

Entretanto, a menos que essa revisão de literatura seja conduzida de uma forma confiável e abrangente, seus resultados possuirão pouco valor científico. Uma revisão de literatura conduzida sem um protocolo de revisão pré-estabelecido, pode ser dirigida por interesses pessoais de seus pesquisadores, levando a resultados pouco confiáveis.

No que diz respeito à Engenharia de Software, a pesquisa conduzida ainda realiza pouca aplicação de métodos científicos no que se refere à identificação de conhecimento na literatura. Frequentemente, revisões de literatura são conduzidas informalmente, sem um planejamento e critérios de seleção estabelecidos a priori. Como consequência, muitas dessas revisões caracterizam-se por serem:

- **Pouco abrangentes.** Numa revisão informal de literatura, a escolha das fontes dos estudos a serem analisados, e a busca por estudos nessas fontes, é feita de forma *ad hoc*, sem um planejamento prévio. Como consequência, a revisão de literatura pode se caracterizar por ser pouco abrangente, haja vista que:
 1. Fontes importantes de estudos podem ser negligenciadas durante a condução da revisão de literatura.
 2. Como geralmente não há uma utilização de uma estratégia de busca dentro das fontes selecionadas, importantes estudos podem não ser capturados pela revisão informal de literatura.
- **Não passíveis de repetição.** Por não apresentarem um protocolo de revisão estabelecido a priori, as revisões informais de literatura não são passíveis de repetição.
- **Pouco confiáveis.** A possibilidade de repetição faz com que a revisão seja passível de auditagem, haja vista que os resultados documentados podem ser verificados de acordo com os resultados obtidos pela repetição da execução do protocolo.
- **Dependentes dos revisores.** Sem a possibilidade de repetição, as revisões informais de literatura são frequentemente dependentes dos revisores que a

conduziram, aumentando possivelmente o viés de seus resultados (MAFRA e TRAVASSOS, 2005a).

Nesse sentido, o desenvolvimento de uma abordagem sistemática de revisão visa a estabelecer um processo formal para conduzir este tipo de investigação, evitando a introdução de eventuais vieses da revisão de literatura informal.

Dessa forma, uma revisão sistemática atua como um meio para identificar, avaliar e interpretar toda pesquisa disponível e relevante sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse.

Em vista disso, a condução de uma revisão sistemática supostamente apresenta uma avaliação justa do tópico de pesquisa, à medida que utiliza uma metodologia de revisão rigorosa, confiável e passível de auditoria (KITCHENHAM, 2004b). Além disso, uma revisão sistemática deve obrigatoriamente conter o protocolo de busca executado de forma a permitir que a revisão seja repetida por outros pesquisadores interessados.

Uma importante diferença entre revisões informais de literatura e revisões sistemáticas refere-se à presença de variações, ou seja, resultados inconsistentes entre os estudos analisados. Segundo BIOLCHINI *et al.* (2005), para a condução de revisões informais de literatura, a presença de variações entre estudos tende a representar uma fonte de ruído, ou seja, um fator negativo para o entendimento e julgamento. No caso de revisões sistemáticas, a presença de ruídos é vista como um fator estimulante de pesquisa.

Dessa forma, pesquisadores conduzindo revisões sistemáticas devem despender cada esforço na identificação e relato de pesquisas que apóiam e que não apóiam suas hipóteses (KITCHENHAM, 2004b). Caso os estudos identificados apresentem resultados consistentes, a revisão sistemática provê indícios de que o fenômeno é robusto e generalizável a outros contextos. Caso os resultados dos estudos sejam inconsistentes, as fontes de variação desses resultados podem ser estudadas.

Outra diferença significativa refere-se aos diferentes propósitos das revisões informais de literatura e das revisões sistemáticas. Uma revisão sistemática não é simplesmente uma revisão de literatura conduzida conforme um planejamento. A revisão de literatura é parte integrante de uma revisão sistemática, que possui objetivos maiores. Nesse sentido, uma revisão de literatura atua como um meio pelo qual um determinado propósito é atendido, ou seja, uma revisão de literatura permite a coleta de dados que possam ser analisados posteriormente com o objetivo de gerar-se evidências na área, que é o propósito das revisões sistemáticas.

Além desses aspectos, uma revisão sistemática não consiste em um simples rearranjo dos dados já conhecidos ou publicados. É também um novo tipo de abordagem metodológica para fazer pesquisa, com uma finalidade de integrar resultados experimentais. Conseqüentemente, a condução de revisões sistemáticas enfatiza a descoberta de princípios gerais, em um nível mais elevado de abstração conceitual no campo de pesquisa, além de incentivar o diagnóstico e a análise das inconsistências externas encontradas ao comparar estudos individuais com resultados contrastantes entre si. Nesse contexto, a condução de revisões sistemáticas ajuda a elucidar novos aspectos na área de investigação, direcionando esforços em pesquisa na área (BIOLCHINI *et al.* 2005).

Na próxima subseção, são discutidos os principais benefícios proporcionados pela condução de revisões sistemáticas, do ponto de vista de seus principais *stakeholders*.

3.2 Benefícios da Utilização de Revisões Sistemáticas

A condução de revisões sistemáticas na Engenharia de Software poderia vir a beneficiar diferentes *stakeholders*, entre eles:

- **Estudantes:** devido ao alto nível de abrangência referente à obtenção de estudos primários proporcionado pela condução de revisões sistemáticas, estudantes seriam contemplados com uma quantidade maior de informações acerca do tópico sendo pesquisado. Com isso, os estudantes poderiam identificar diversas oportunidades de pesquisa relacionadas como pertinentes por outros grupos de pesquisa ou pela indústria de software. Além disso, o caráter metodológico da revisão sistemática, que requer a elaboração e a execução de um protocolo de pesquisa, auxiliaria o estudante a manter o foco na pesquisa sendo conduzida, evitando eventuais desvios proporcionados pela leitura de outros artigos interessantes, mas não relacionados ao escopo estabelecido para a pesquisa.
- **Orientadores:** A execução do protocolo de revisão sistemática possibilita a obtenção de diversos artigos sob um determinado tema. Esses artigos deverão passar por um processo de avaliação, considerando os critérios de inclusão e exclusão estabelecidos no protocolo. Dessa forma, o orientador poderia monitorar a pesquisa sendo conduzida por seus alunos, ao verificar periodicamente os valores de algumas métricas, como por exemplo, a quantidade de estudos avaliados, selecionados e resumidos. Além disso, a possibilidade de se obter diversos estudos primários poderia fornecer

subsídios para o orientador sobre eventuais riscos associados ao prosseguimento da pesquisa em um determinado tópico de interesse.

- **Comunidade Acadêmica:** a condução de revisões sistemáticas possibilitaria à comunidade acadêmica dispor da caracterização experimental de diversas tecnologias em uso. A caracterização dessas tecnologias poderia ser continuamente incrementada através da repetição desses estudos experimentais em diferentes contextos proporcionando, dessa forma, o acúmulo de conhecimento esperado para o estabelecimento de qualquer ramo científico.
- **Indústria de Software:** a indústria de software poderia se beneficiar da condução de revisões sistemáticas ao ser contemplada com resultados experimentais que apontassem o que funciona e o que não funciona na aplicação de uma determinada tecnologia sob quais circunstâncias. Essas informações obtidas experimentalmente poderiam ser utilizadas como um apoio para a tomada de decisão no que se refere principalmente à aquisição de uma tecnologia.

Na próxima subseção, são descritos como a necessidade e os conceitos de revisão sistemática foram discutidos ao longo dos anos.

3.3 Histórico da Revisão Sistemática

A carência de aplicação de métodos científicos no que se refere à revisão de literatura não é uma constatação exclusiva da área de Engenharia de Software. Durante muito tempo, a área médica esteve repleta de revisões que não utilizavam métodos para identificar, avaliar e sintetizar a informação existente na literatura (COCHRANE, 2003). No final da década de 80, estudos conduzidos para avaliar a qualidade das publicações médicas chamaram a atenção para a baixa qualidade científica das publicações da área de saúde (COCHRANE, 2003). MULROW (1987) avaliou cinquenta artigos publicados em quatro grandes revistas médicas entre os anos de 1985 e 1986 e comprovou a baixa utilização de métodos científicos nessas revisões.

Para reverter esse quadro, durante os anos 70s, e início dos anos 80s, psicólogos e cientistas sociais tiveram suas atenções voltadas para a elaboração de diretrizes que sistematizassem os passos requeridos para a condução de uma revisão de literatura. O objetivo era tentar minimizar os vieses e erros de futuras revisões a serem conduzidas.

Desde então, o reconhecimento da necessidade da condução de revisões de forma sistemática e formal em Medicina tem crescido rapidamente. Este fato pode ser comprovado pela grande quantidade de revisões formais publicadas a cada ano na área médica (NHSCRD, 2003). Tais revisões, denominadas de revisões sistemáticas, são revisões rigorosas da literatura à procura de indícios que possam levar à identificação de evidências sobre um tema de pesquisa ou tópico na área em questão.

A importância da condução de revisões sistemáticas é ressaltada em KITCHENHAM *et al.* (2004a). No final dos anos 80s e início dos anos 90s, estudos apontaram que, de um lado, a falha em organizar a pesquisa médica em revisões sistemáticas podia custar vidas (COCHRANE, 1989). Por outro lado, julgamentos clínicos de alguns especialistas foram considerados inadequados, quando comparados aos resultados provenientes de revisões sistemáticas (ANTMAN *et al.*, 1992).

Na próxima subseção, é destacado o estado atual de evidência observado na Engenharia de Software.

3.4 Estado Atual de Evidência na Engenharia de Software

Foi o trabalho de KITCHENHAM *et al.* (2004a) o primeiro a estabelecer um paralelo entre Medicina e Engenharia de Software, no que diz respeito à abordagem baseada em evidência, e a recomendar o uso de revisões sistemáticas na pesquisa em Engenharia de Software. Naquele momento, ao avaliar o grau de evidência obtido na atual pesquisa em Engenharia de Software, KITCHENHAM *et al.* (2004a) havia identificado que essa pesquisa era caracterizada por ser:

- **Limitada.** Apesar de muitos grupos de pesquisa conduzirem valiosos estudos experimentais, parte considerável desses trabalhos visa unicamente a publicações individuais ou a dissertações de pós-graduação sobre as áreas de interesse que sejam mais convenientes ao grupo de pesquisa.
- **Fragmentada.** A limitação do escopo de áreas de interesse de cada grupo de pesquisa resulta numa despreocupação com um propósito mais abrangente e universal de pesquisa.
- **Dispersa.** A falta de uma padronização para a condução de revisões sistemáticas em Engenharia de Software, somada à falta de uma cultura acadêmica que privilegie repetições de estudos e a aplicação de meta-análise, contribui para tornar a pesquisa em Engenharia de Software dispersa.

Como resultado dessa observação, KITCHENHAM (2004b) propôs um processo para a condução de revisões sistemáticas em Engenharia de Software, discutido na próxima subseção.

3.5 Processo de Revisão Sistemática em Engenharia de Software

O processo para a condução de revisões sistemáticas definido em KITCHENHAM (2004b) envolve três principais etapas, entre elas: Planejamento da Revisão, Condução da Revisão, e Publicação dos Resultados, conforme ilustrado na Figura 2.

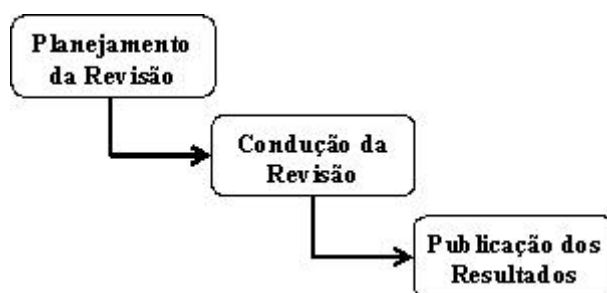


Figura 2 - Processo para a condução de revisões sistemáticas definido em KITCHENHAM (2004b).

Durante a etapa de Planejamento da Revisão, os objetivos da pesquisa são listados e o protocolo de revisão é definido. Durante a etapa de Condução da Revisão, as fontes para a revisão sistemática são selecionadas, os estudos primários são identificados, selecionados e avaliados de acordo com os critérios de inclusão e de exclusão, e de qualidades estabelecidos durante o protocolo da revisão. Após a seleção dos estudos, os dados dos estudos são extraídos e sintetizados para serem finalmente publicados durante a etapa de Publicação dos Resultados.

Apesar da aparência seqüencial, é importante destacar que algumas etapas do processo são iterativas. Em particular, algumas atividades são iniciadas durante o desenvolvimento do protocolo e re-executadas à medida que a revisão é conduzida. Como exemplo, cita-se a escolha dos critérios de inclusão e exclusão dos estudos primários. Inicialmente esses critérios são definidos na etapa de planejamento, quando o protocolo é definido, mas refinados à medida que os critérios de qualidade são definidos. Nos Apêndices I – IV são apresentados Templates que complementam a proposta encontrada em Biolchini et al. (2005). Ainda no Apêndice V é apresentado um protocolo exemplo, relacionado a pesquisa descrita em Mafra (2006).

Na próxima subseção, é descrito o estágio atual da Engenharia de Software no que se refere à condução de revisões sistemáticas.

3.6 Estágio Atual de Revisões Sistemáticas em Engenharia de Software

De 2004 para cá, um considerável avanço foi obtido no que diz respeito à definição de modelos de protocolos e à condução de revisões sistemáticas em Engenharia de Software. Mesmo com todas as dificuldades ainda existentes no contexto da Engenharia de Software para se executar Revisões Sistemáticas (Mia net al., 2005), alguns resultados interessantes já podem ser identificados.

Em KITCHENHAM (2004b) são apresentadas diretrizes adaptadas de outras áreas de pesquisa para a realização de revisões sistemáticas em Engenharia de Software. BIOLCHINI *et al.* (2005) vão além, ao definirem um processo para a condução de revisões sistemáticas mais detalhado, considerando o impacto do tipo de questões nos procedimentos de revisões e na condução de meta-análise.

No que se refere à condução de revisões sistemáticas, algumas instituições de pesquisa se destacaram no cenário mundial, entre elas a Universidade de Keele do Reino Unido, a Universidade de Auckland, da Nova Zelândia, o National ICT, da Austrália, a COPPE/UFRJ, do Brasil, a Universidade de Trondheim e o Laboratório de Pesquisa Simula, ambos da Noruega. Dentre as revisões sistemáticas conduzidas, destacam-se:

- JASPERSON *et al.* (2002): revisão sistemática conduzida na área de tecnologia de informação, onde o objetivo principal foi possibilitar o ganho de entendimento sobre a área de forma a apontar oportunidades de pesquisa.
- CABRAL e TRAVASSOS (2004): Revisão sistemática conduzida com o objetivo de identificar iniciativas existentes para revisão e verificação de modelos de processo de software.
- DIAS NETO e TRAVASSOS (2004): O objetivo da revisão foi identificar iniciativas de planejamento de testes de software.
- CONTE *et al.* (2004): esta revisão sistemática buscou caracterizar quais processos de desenvolvimento têm sido utilizados na indústria para desenvolver aplicações Web.

- DYBÅ *et al.* (2005): revisão conduzida com o objetivo de realizar uma avaliação quantitativa do poder estatístico da pesquisa em Engenharia de Software Experimental.
- MENDES (2005): a autora conduziu uma revisão sistemática com o objetivo de investigar o rigor apregoado na pesquisa em Engenharia Web. A revisão sistemática identificou 173 artigos dos quais, segundo os critérios definidos pela autora, apenas 5% podem se considerados rigorosos do ponto de vista da metodologia de pesquisa adotada.
- MAFRA e TRAVASSOS (2005): revisão sistemática conduzida com o objetivo de caracterizar o apoio fornecido por técnicas de leitura durante a revisão de requisitos de software descritos em linguagem natural. A análise dos resultados dos estudos obtidos possibilitou caracterizar os pontos fortes e fracos relacionados à aplicação das técnicas identificadas.
- BARCELOS e TRAVASSOS (2006): o objetivo da revisão conduzida foi identificar abordagens de avaliação para modelos arquiteturais de software.
- KITCHENHAM *et al.* (2006): os autores conduziram uma revisão sistemática visando a identificar estudos que comparassem as predições de modelos de estimativa genéricos (*cross-company models*) com as predições de modelos de estimativa específicos (*within-company models*) baseando-se na análise de dados históricos de projetos.

4. Considerações Finais

A Engenharia de Software ainda faz pouco uso de métodos científicos na condução de pesquisas na área. Tecnologias são freqüentemente desenvolvidas em laboratórios sem que um processo adequado de transferência para a indústria seja conduzido. A falta de caracterização da tecnologia em uso, decorrente da não utilização de uma abordagem experimental durante o desenvolvimento dessa tecnologia, torna a Engenharia de Software palco para especulações sobre a eficácia e a qualidade de novos produtos e processos.

Mesmo entre algumas áreas da Engenharia de Software que fazem uso intenso de estudos experimentais, como a pesquisa sobre técnicas de leitura de software, não se observa a condução de estudos secundários, responsáveis por generalizar os resultados desses estudos a um contexto maior (MAFRA e TRAVASSOS, 2005).

Entretanto, esse quadro tem mudado nos últimos anos. O reconhecimento da necessidade de condução de revisões sistemáticas, e a definição de um processo e de um modelo de protocolo a ser seguido durante a condução da revisão, têm estimulado a condução e a apresentação dos resultados de diversas revisões sistemáticas.

Com a adoção de uma abordagem baseada em evidência, caracterizada pela condução de estudos primários e secundários, a Engenharia de Software poderia atingir um elevado grau de maturidade. Dessa forma, conforme apontado por JURISTO e MORENO (2002), evoluiria-se do desenvolvimento de software baseado em especulação para o desenvolvimento de software baseado em fatos. Essa evolução permitiria, segundo as pesquisadoras, transformar o processo de construção de software de qualidade em um processo de predição.

5. Referências

- AMARAL, E. (2003), “Empacotamento de Estudos Experimentais em Engenharia de Software”. Dissertação de Mestrado. COPPE/UFRJ, Programa de Engenharia de Sistemas e Computação, Rio de Janeiro, RJ, Brasil.
- ANTMAN E., LAU, J., KUPELNICK, B., MOSTELLER, F., CHALMERS, T. (1992) “A comparison of results of meta-analysis of randomized controlled trials and recommendations of clinical experts”, JAMA-Journal of the American Medical Association, 268(2):240-248, July 1992.
- APA (2005), American Psychological Association, “Journal of Experimental Psychology: Learning, Memory, and Cognition”. In: <http://www.apa.org/journals/xlm>, acessado em 04/01/2006.
- BABBIE, E. (1990) “Survey Research Methods”, Wadsworth, ISBN 0-524-12672-3.
- BARCELOS, R., TRAVASSOS, G. (2006) “Evaluation Approaches for Software Architectural Documents: a Systematic Review”. In: Proceedings of the Ibero-American Workshop on Requirements Engineering and Software Environments (IDEAS'06), La Plata, Argentina.
- BASIL, V. R., SELBY, R. W., HUTCHENS, D. H. (1986) “Experimentation in Software Engineering”, IEEE Transactions on Software Engineering, 12 (7), pp. 733-743.
- BASIL, V. R. (1996) “The Role of Experimentation in Software Engineering: Past, Current and Future”, proceedings 18th International Conference on Software Engineering, Berlin, Germany, pp. 442-449.

- BIOLCHINI, J., MIAN, P.G., NATALI, A.C., TRAVASSOS, G.H. (2005) “Systematic Review in Software Engineering: Relevance and Utility”, Relatório Técnico ES-679/05, PESC - COPPE/UFRJ.
- CABRAL, R, TRAVASSOS, G. (2004) “Inspeção em Modelos de Processo de Software”. Monografia de Curso Final COS 722, PESC - COPPE/UFRJ.
- COCHRANE, AI. (1989) In Chalmers I, Enkin M, Keirse MJNC, eds. “Effective care in pregnancy and childbirth”. Oxford University Press, Oxford, 1989.
- COCHRANE COLLABORATION (2003), Cochrane Reviewers’ Handbook. Version 4.2.1. <http://www.cochrane.dk/cochrane/handbook/hbook.htm>, acessado em 04/01/2006.
- CONTE, T., TRAVASSOS, G.H., MENDES E. (2004), "Revisão Sistemática sobre Processos de Desenvolvimento para Aplicações Web", Monografia de final de curso COS 722, PESC - COPPE/UFRJ.
- COOK, T., CAMPBELL, D. (1979) “Quasi-Experimentation – Design and Analysis Issues for Field Settings”, Houghton Mifflin Company, 1979.
- DIAS NETO, A., TRAVASSOS, G. (2004) "Planejamento de Testes de Software". Monografia de Curso Final COS 722, PESC - COPPE/UFRJ.
- DYBÅ, T., KAMPENES, V., SJØBERG, D. (2005) “A systematic Review of Statistical Power in Software Engineering Experiments”. Journal of Information and Software Technology (2005) 1-11.
- FENTON, N., PFLEEGER, S., GLASS, R. (1994) “Science and Substance: A Challenge to Software Engineers”, IEEE Software, pp. 86-95, July, 1994.
- GLASS, R. (1994) “The Software Research Crisis”, IEEE Software, pp. 42-47, November, 1994.
- GLASS, R. “Facts and Fallacies of Software Engineering”, Addison-Wesley, 2002.
- JASPERSON, J., BUTLER, B., CARTE, T., CROES, H., SAUNDERS, C., ZHEMG, W. (2002) “Review: Power and Information Technology Research: A Metatriangulation Review”. MIS Quarterly, 26(4): 397-459, December 2002.
- JURISTO, N., MORENO, A. (2002) “Reliable Knowledge for Software Development”, IEEE Software, pp. 98-99, sep-oct, 2002.
- KITCHENHAM, B., PICKARD, L., PFLEEGER, S. (1995) “Case Studies for Method and Tool Evaluation”, IEEE Software, pp. 52-62, July, 1995.
- KITCHENHAM, B., DYBÅ, T., JORGENSEN, M. (2004a) “Evidence-based Software Engineering”, Proceedings of the 26th International Conference on Software Engineering (ICSE'04).

- KITCHENHAM, B. (2004b) "Procedures for Performing Systematic Reviews", Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia.
- KITCHENHAM, B., TRAVASSOS, G. H. (2006) "A Systematic Review of Cross- vs. Within-Company Cost Estimation Studies" 10th International Conference on Evaluation and Assessment in Software Engineering, EASE'06, 10-11 April 2006, Keele University, Staffordshire, United Kingdom.
- MAFRA, S. N. (2004) "Infra-Estrutura para Automatização de Processos de Software", Monografia de Projeto Final de Curso, DCC/IM, UFRJ, Rio de Janeiro, RJ, Brasil.
- MAFRA, S.N. (2006). "Leitura Baseada em Perspectiva: A Visão do Projetista Orientada a Objetos". Dissertação de Mestrado. Programa de Engenharia de Sistemas e Computação. COPPE/UFRJ.
- MAFRA, S. N., TRAVASSOS, G. H. (2005) "Técnicas de Leitura de Software: Uma Revisão Sistemática". In: XIX Simpósio Brasileiro de Engenharia de Software, SBES'05, Uberlândia, MG, Brasil.
- MENDES, E. (2005) "A systematic review of web engineering research", International Symposium on Empirical Software Engineering, 480-490, Nov. 17, 2005.
- MIAN, Paula ; CONTE, Tayana Uchoa ; NATALI, Ana Candida Cruz ; BIOLCHINI, Jorge ; TRAVASSOS, G. H. . Lessons Learned on Applying Systematic Reviews to Software Engineering. In: WSESE2005 - Workshop Series in Empirical Software Engineering, 2005, Oulu. Proceedings of the 3rd International Workshop GUIDELINES FOR EMPIRICAL WORK in the Workshop Series on Empirical Software Engineering 2005. Kaiserslautern : Fraunhofer Center, 2005. v. 1. p. 1-6.
- MULROW CD (1987), "The medical review article: state of the science", Ann Intern Med; 106:485-8.
- NHS Centre for Reviews and Dissemination. (2003), "Database of Abstracts of Reviews of Effectiveness". In: The Cochrane Library, Issue 1. Oxford: Update Software. Updated quarterly.
- PFLEEGER, S. (1994) "Experimental Design and Analysis in Software Engineering Part 1-5", ACM Sigsoft, Software Engineering Notes, Vol. 19, n° 4, pp. 16-20; Vol. 20, n° 1, pp. 22-26; Vol. 20, n° 2, pp. 14-16; Vol. 20, n° 3, pp. 13-15; Vol. 20, n° 4, pp. 14-17, 1994-1995.

- PFLEEGER, S. (1999) "Albert Einstein and Empirical Software Engineering". IEEE Computer 32 (10): 32-37.
- PLESS (2005), Princeton Laboratory for Experimental Social Scienc. In. <http://pless.princeton.edu/index.html>, acessado em 04/01/2006.
- POTTS, C. (1993) "Software Engineering Research Revisited", IEEE Software, pp. 19-28, September 1993.
- REED, K., (2000) "Software engineering - a new millennium?", IEEE Software, jul.-ago, 2000.
- ROBSON, C. (1993) "Real World Research: A Resource for Social Scientists and Practitioners-Researchers", Blackwell.
- SACKETT, D., STRAUS, S., RICHARDSON, W., ROSENBERG, W., HAYNES, R. (2000) "Evidence-Based Medicine: How to Practice and Teach EBM", Second Edition, Churchill Livingstone: Edinburgh, 2000.
- SHULL, F. (1998) "Developing Techniques for Using Software Documents: A Series of Empirical Studies", PhD Thesis, Department of Computer Science, University of Maryland, USA.
- SHULL, F., MENDONÇA, M., BASILI, V., CARVER, J., MALDONADO, J., FABBRI, S., TRAVASSOS, G., FERREIRA, M. (2004) "Knowledge-Sharing Issues in Experimental Software Engineering", Empirical Software Engineering, Volume 9 Issue 1-2, March, 2004.
- STAKE, R. (1995) "The Art of Case Study Research", SAGE Publications.
- TICHY, W., LUKOWICZ, P., PRECHELT, L., HEINZ, E. (1995) "Experimental Evaluation in Computer Science: A Quantitative Study", Journal of System and Software, 28(1), pp. 9-18.
- TICHY, W. F. (1998) "Should Computer Scientists Experiment More?", IEEE Computer, 31 (5), pp. 32-39.
- TRAVASSOS, G., GUROV, D., AMARAL, E. (2002) "Introdução à Engenharia de Software Experimental". In: Relatório Técnico ES-590/02-Abril, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- WÖHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M., REGNELL, B., WESSLÉN, A. (2000) "Experimentation in Software Engineering: An Introduction", The Kluwer International Series in Software Engineering, Norwell, USA, Kluwer Academic Publishers.
- XLAB (2005), UC Berkeley's Experimental Social Science Laboratory, In: <http://xlab.berkeley.edu/>, acessado em 04/01/2006.

YIN, R. (1994) “Case Study Research Design and Methods”, Sage Publications, Beverly Hills, California.

ZELKOWITZ, M., WALLACE, D. (1998) “Experimental Models for Validating technology”, IEEE Computer, 31(5), pp. 23-31.

Apêndice I – Modelo do Protocolo de Revisão

Objetivo:

Formulação da pergunta:

Intervenção:

Controle:

População:

Resultados:

Aplicação:

Critérios de seleção de fontes:

Métodos de busca de fontes:

Palavras-chave:

Listagem de fontes:

Tipo dos artigos:

Idioma dos artigos:

Critérios de inclusão e exclusão dos artigos

Critérios de qualidade dos estudos primários:

Processo de seleção dos estudos primários

Avaliação da qualidade dos estudos primários:

Estratégia de extração de informação:

Sumarização dos resultados:

Apêndice II – Formulário de Condução da Revisão

Fonte: *(fonte na qual a busca foi conduzida)*

Data de busca:

Palavras-chave utilizadas:

Strings de busca utilizadas: *(combinação de palavras-chave utilizadas)*

Lista de artigos encontrados

(Referências dos artigos encontrados pela busca)

Lista dos artigos incluídos

Nome do artigo:

Autores:

Data de publicação:

**Veículo de
publicação:**

Critérios de Inclusão e Exclusão

Critérios	Resultados

Justificativa: *(comentários do pesquisador sobre sua escolha)*

Lista dos artigos excluídos

Nome do artigo:

Autores:

Data de publicação:

**Veículo de
publicação:**

Critérios de Inclusão e Exclusão

Critérios	Resultados

Justificativa: *(comentários do pesquisador sobre sua escolha)*

Apêndice III – Formulário de Seleção de Estudos

Nome do artigo:

Autores:

Data de publicação:

Veículo de publicação:

Fonte: *(fonte na qual o artigo foi obtido)*

Situação: *(incluído ou excluído)*

Critérios de Inclusão e Exclusão

Critérios	Resultados
Os artigos devem estar disponíveis na <i>web</i> .	S ou N
Os artigos devem apresentar textos completos dos estudos em formato eletrônico.	S ou N
Os artigos devem estar descritos em inglês.	S ou N
Os artigos devem contemplar técnicas de inspeção de documentos de requisitos descritos em linguagem natural.	S ou N
Os artigos devem contemplar a execução de estudos experimentais investigando técnicas de inspeção de documentos de requisitos.	S ou N

Justificativa: *(comentários do pesquisador sobre sua escolha)*

Apêndice IV – Formulário de Extração de Dados

Nome do Artigo:

Autores:

Data de Publicação:

Veículo de Publicação:

Fonte:

Abstract:

Resumo: *(o artigo deve ser resumido pelo pesquisador.)*

Estudo

Data de execução:

Local:

Tipo: *(experimento, estudo de caso etc)*

Descrição:

Hipóteses avaliadas

Variáveis independentes

Variáveis dependentes

Participantes

Material

Projeto do estudo

Ameaças à validade

Resultados

Comentários adicionais *(comentários do pesquisador acerca do estudo)*

Referências relevantes *(lista das referências relevantes e o porquê que tais referências são relevantes)*

APENDIX V. Protocol Example: Reading Techniques

The systematic review protocol was defined based on the protocol models available in Biolchini *et al.* (2005) and Mendes e Kitchenham (2005).

Goal: The *objective* is to identify, analyze, and evaluate experimental studies concerning reading techniques for software requirements document written in natural language, *with the purpose of* characterize them *with respect to* usability, efficiency in defect detection, and effectiveness in the defect detection coverage *from the point of view of* researchers *in the context* in which such studies were performed.

Question Formularization: What are the reading techniques for software requirements documents written in natural language?

Intervention: Software requirements document reading techniques.

Control: Checklist approach and *ad hoc* reading.

Population: Software requirements document inspection processes.

Results: Software requirements document reading techniques.

Application: Software projects applying requirements inspection.

Sources Selection Criteria Definition

- Availability to consult articles on the web;
- Presence of search mechanisms using keywords;
- Non-variability in search results by using a same set of keywords;
- To be a source A-level Capes evaluated (Qualis, 2004).

Sources Search Methods: research through web search engines. So, in this review, manual search is not to be considered.

Keywords:

- requirement specification, requirement document, user requirement.
- reading technique;
- review, inspection, fault detection, defect detection;
- experimental software engineering, empirical software engineering.

Sources List: CAPES periodicals (IEEE journals and conferences, IEE journals, IEE conferences, IEEE conferences, ACM journals, ACM conferences, Kluwer journals e Elsevier journals).

Papers Type Definition: papers concerning experimental studies on reading techniques.

Papers Languages: English. The choice of English language is justified by the universality of such language. So, papers written in Portuguese are not to be considered. Experimental studies must be repeatable in different contexts in different countries. A paper in Portuguese language would clearly limit this possibility.

Papers Inclusion and Exclusion Criteria Definition:

- Papers must be available on web;
- Papers must present full texts of studies in a electronic format;
- Papers must be written in English;
- Papers must concern inspection techniques for software requirements document written in natural language;
- Papers must describe experimental studies about software requirements inspection techniques.

The scale used is nominal involving two options: *yes* or *no*.

Quality Primary Studies Criteria Definition: the quality evaluation of the studies will be based on the software engineering experimental studies evaluation criteria described in Kitchenham *et al.* (2002).

Primary Studies Selection Process:

The primary studies selection process describes five steps:

1. A researcher runs a search in every selected source in order to identify papers concerning experimental studies;
2. The found papers are obtained and their references are documented in the Found Papers List, contained in the Review Report (see Appendix A);
3. The found papers are evaluated by the researchers through the verification of inclusion and exclusion criteria established; such verification is performed by reading the abstract of the paper;
4. The included and excluded papers are documented in the Included Papers List and the Excluded Papers List, respectively, present in the Review Report (see Appendix A), in addition with the justification of their inclusion or exclusion;
5. The included papers are evaluated by reading the full text; the selected papers are documented on the Studies Selection Report (see Appendix B) and then send for evaluation by the primary studies quality evaluation process. The papers excluded are documented in the Excluded Papers List in addition with the justification for its exclusion.

If there is any divergent opinion, the researchers must reach a consensus on this matter and register it.

Primary Studies Quality Evaluation Process: the studies selected by the primary studies selection process will be evaluated by researchers according to established quality criteria. The result of the quality evaluation of each study will determine whether such study will be included or excluded in the set of studies that will have their data extracted.

Data Extraction Strategy: for each selected study by the primary studies quality evaluation process will be used a copy of the Data Extraction Report (see Appendix C).

Data Synthesis: the results will be tabulated. No kind of meta-analysis will be conducted.

The review protocol was evaluated by a specialist. It was generated two versions until its final acceptance.

References

Biolchini, J., Mian, P.G., Natali, A.C., Travassos, G.H. (2005) “Systematic Review in Software Engineering: Relevance and Utility”, Technical Report ES-679/05, PESC/COPPE/UFRJ.

Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., Emam, K., Rosenberg, J. (2002) “Preliminary Guidelines for Empirical Research in Software Engineering”, IEEE Transactions on Software Engineering, vol. 28, n° 8, August 2002.

Mendes, E., Kitchenham, B. (2005) “Protocol for Systematic Review”, Available at <http://www.cs.auckland.ac.nz/emilia/srspp.pdf>, accessed in 2005-10-05.

Qualis (2004) Qualis – Classification System for Periodicals, Annals, and Journals, <http://qualis.capes.gov.br>, accessed in 2004-12-20.

Appendix V.1 - REVIEW REPORT

Source: *(source in which the search was conducted.)*

Search date:

Keywords:

Search strings: *(combination of keywords used in the search.)*

Found Papers List

(References of papers found.)

Included Papers List

Paper name:

Authors:

Publication date:

Publication vehicle:

Inclusion and Exclusion Criteria

Criteria	Results

Justification: *(researcher's comments about his/her choice.)*

Excluded Papers List

Paper name:

Authors:

Publication date:

Publication vehicle:

Inclusion and Exclusion Criteria

Criteria	Results

Justification: *(researcher's comments about his/her choice.)*

Appendix V.2 - STUDIES SELECTION REPORT

Paper Name:

Authors:

Publication date:

Publication vehicle:

Source: *(source in which the paper was obtained.)*

Situation: *(included or excluded.)*

Inclusion and Exclusion Criteria

Criteria	Results
Papers must be available on the web	Y or N
Papers must present full texts of studies in a electronic format	Y or N
Papers must be written in English	Y or N
Papers must concern inspection techniques for software requirements documents written in natural language	Y or N
Papers must describe experimental studies about software requirements inspection techniques	Y or N

Justification: *(researcher's comments about his/her choice)*

Appendix V.3 - DATA EXTRACTION REPORT

Paper Name:

Authors:

Publication date:

Publication vehicle:

Source:

Abstract:

Summary: *(the paper has to be summarized by the researcher.)*

Study

Execution date:

Localization:

Type: *(experiment, case study, etc)*

Description:

Evaluated Hypothesis

Independent Variables

Dependent Variables

Participants

Material

Study Design

Threats to validity

Results

Additional comments

(researcher's comments about the study.)

Relevant References

(list of relevant references and why they are relevant.)