An Evidence-based Approach to Discovery and Assessment of Software Engineering Decisions

Priyanka Mathur
Research Scholar, Department of Computer Science, The IIS University, Jaipur, INDIA

Swati V. Chande
Professor, International School of Informatics and Management, Department of Computer Science, Jaipur, INDIA

ABSTRACT

An Evidence-based approach is using a best available evidence for making a judicious decision about a given set of problem. Evidence-based approach is an integration of individually gained expertise with the best possible evidence available from a systematic research. It started in medicine as evidence-based medicine (EBM) and is now being used in other fields such as nursing, psychology, education, library and information science also. Its basic principles are that all practical decisions made should 1) be based on research studies and 2) that these research studies are selected and interpreted according to some specific norms characteristic for Evidence Based Practice [EBP]. Software draws its roots from EBM and does Evidence Based Software Engineering [EBSE] which is potentially important because of the central place software intensive systems are starting to take in everyday life. In Evidence-based software engineering [EBSE], all the experiences are properly documented in order to inform software practice adoption decisions. In EBSE, the study factor would be the technology of interest. The technological specifications should be very detailed and not at a very high level of abstraction that is the software lifecycle and all the design methods should be properly read and documented and only then should the engineer collect evidences on it and design the software generation model. Evidence based software engineering can be applied in testing and cost estimation. Various metaheuristic search techniques are applied for searching literature and relevant evidences are gathered, these evidences are then put into practice. The results are compared with existing practices.

Keywords: Evidence based approach, contentious decisions, Evidence based software engineering, level of abstraction, testing and cost estimation, metaheuristic search techniques.

1. INTRODUCTION

Evidence means information that supports a statement. It is gathered whenever a proposition is to be proved. Evidence thus, in specific terms can be defined as "Evidence for a proposition is anything that increases the estimate of the probability of the truthfulness of the proposition."[1] A proposition here is any assumption related to the real world. Evidences are gathered for all available propositions and the best available research evidence is used to draw inferences. To bridge the gap between actual and theoretical decisions, the decisions need to be backed by sufficient evidences. Based upon the real life facts and happenings factual data is collected then inferences are drawn from the data which happens to be the key parameter for any decisions being taken. Evidence based approach [EBA] is followed for the same. The evidence, by itself, does not make any decisions, but it can help support the decision making process. In EBA a conscientious and judicious use of the best available evidence before taking any decision is being done. It integrates individual expertise with the best available external practicing approach through systematic research. EBA is carried out in three steps: first it takes into account practical expertise, secondly need of the system to be developed is evaluated and thirdly best research practices are incorporated into the study or process of decision making. The full integration of these three components into decisions enhances the opportunity for optimal outcome. Thus a lot of thought process is being carried out before reaching to any conclusion. Along with the scientific studies to support the three ways of evidences gathering are: wisdom, expert opinion, and common sense.

EBA draws its roots from the field of medicine wherein a doctor works on the theory of elimination. He gathers evidence from his patients and clinical researches and uses these pieces of knowledge for further treatment of his patients. While questioning a patient he keeps on eliminating those ailments from his mind which do not have particular set of symptoms. Thus he draws conclusion only after he has examined the patient thoroughly and through previous expertise he has eliminated all non considerable diseases. The three steps stated earlier are followed in medicinal practice as shown in The EBM Triad Fig 1. All the other fields that uses EBA also follow the same procedure.

It started in medicine as evidence-based medicine (EBM) and is now being used in other fields such as nursing, psychology, education, library and information science also. Its basic principles are that all practical decisions made should

1) be based on research studies and

2) that these research studies are selected and interpreted according to some specific norms and characteristics for Evidence Based Practice[EBP].

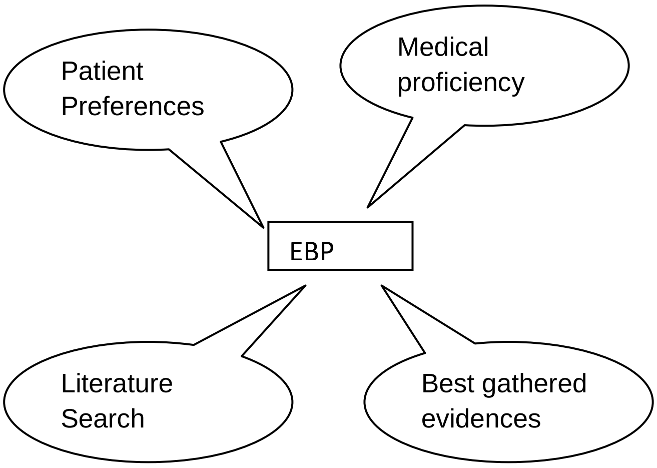

Fig 1. The EBM Practice

Research indicates that expert opinion based medical advice is not as reliable as advice based on the accumulation of results from scientific experiments [3]. Along with its applications in medicine Evidence based approach has its roots in management as well. The managerial decisions and organizational policies are formed using the best available scientific evidences. Like its counterparts in medicine, management and education EBM is a widely used tool in speech pathology. In speech pathology the practitioners consider the factors relating to individuals as per their need and values but they also refer to the clinical skills that are underpinned by knowledge gathered through published research and evidences collected over the years.

Since the rise of computing in the 1940s, the applications and uses of computers have grown at exponential rate and software plays a central role in almost all aspects of computing. The number, size, and application domains of computer programs have grown dramatically and so is the need for software engineering. “Software engineering is that form of engineering that applies the principles of Computer Science and Mathematics to achieving cost-effective solutions to software problems. It is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [IEEE 1990]” [4]. It is obvious that software engineering is required to create high quality software in an efficient and systematic manner.

Resultantly nowadays more emphasis is laid on analysis, evaluation, specification, design, and evolution of software. In today’s era the fast pace of change in computer and software technology needs new and improvised software products. This situation has created customer expectations and competitive forces that strain the developer’s ability to produce quality of software within acceptable development schedules.

Evidence based approach best suited to software engineering as a thorough documentation for every decision taken is done. It can help software developers in following a non conventional development method rather than using the conventional models. For doing so, Software project managers' decisions should be based on solid evidence rather than on common wisdom and EBSE is a methodical and scientific manner upon which managers and practitioners can base their decisions, by forming rational arguments from the evidence they have gathered through a combination of experience and related research. In this approach a unit or a part of software is examined and then gradually proceeds to the whole system. It is possible that EBSE can provide the mechanisms needed to assist practitioners to adopt appropriate technologies and to avoid inappropriate technologies. It is therefore suggested that the goal of evidence-based software engineering (EBSE) should be:

to provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software [5].

In most cases software is built with technologies for which the developer has insufficient evidence to confirm their suitability, limits, qualities, costs, and inherent risks. Thus the practitioners can have difficulty in making judicious decisions about whether to adopt a new technology because in most of the projects there is little objective evidence to confirm the suitability risks. This in turn can lead to poor decisions about technology adoption, as some researchers suggest that:

Software practitioners and managers seeking to improve the quality of their software development processes often adopt new technologies without sufficient evidence that they will be effective, while other technologies are ignored despite the evidence that they most probably will be useful. [6]

It is therefore required that conclusions drawn for any practice should be based on evidences supported by research. For generating proper evidences rigorous evaluation needs to be done which requires systematic, standardized description of target problems. Evidence base is thus built by summing up observations, careful description and measurement of problem and careful citing of specific results.

2. EVIDENCE BASED APPROACH TO DECISION MAKING

In the evidence-based approach controlled trials are conducted, to get the strongest possible evidence to support a hypothesis. This means that appropriate high quality experimental results are given more weight than those to whom studies judged to be of a low quality. In software engineering domain, the study designs conducted for evidence gathering are:

a) Systematic reviews b) Meta-analyses

c) Mapping studiesd) Narrative synthesis techniques

As its name suggests, a **systematic review** requires a systematic and methodical search of the literature in order to present an overall synthesis of results from the highest-quality studies. It is a scientific research method helping the researchers to gather all evidences available on a particular topic and its main objective is to accomplish it in an unbiased manner. **Evidence** is the cumulative data obtained from **‘primary’ studies** that is judged to be relevant to answer a particular research question. The task of aggregating data to answer the question is performed through a **secondary study** [7]**.** A rigorous ‘secondary’ study is thus required to find all relevant data which provides a base for determining what to include and what to exclude. The latest experimental finding reported in Conference Proceedings and Journal papers is the primary data. The appropriateness of the experimental method used in the experiment marks the quality of it.

In **meta-analysis** one has to take the results of several related studies and pool the data, with the effect of creating one large study which can then be analyzed. It combines the results of several studies that address to a common set of research hypotheses. Both systematic reviews and meta-analyses are important techniques for data collection and analysis. They are not considered as empirical scientific studies in themselves but they synthesize the entire body of empirical work that has been done on a topic.

Mapping study or systematic mapping study [SMS] is also done for collecting evidences. This technique is used when researcher have little evidence on a particular field. In many researches it is also used as a tool to enhance SLR based inferences.

Narrative' synthesis is another approach followed in systematic review when findings include multiple studies that are based on the use of words and text to summarize and explain the findings of the synthesis. This method adopts a textual approach which tells the crux of all the findings to a process of synthesis focusing on a wide range of questions.

3. EBA IN SOFTWARE ENGINEERING

A direct research conducted to identify the requirements of the software industry, help practitioners to make more rational decisions about technology adoption, enable better choice of development technologies and increase the acceptability of software intensive systems that interface well with the users. The procedure followed in Evidence based medicine can be summarised as shown in Fig 2:

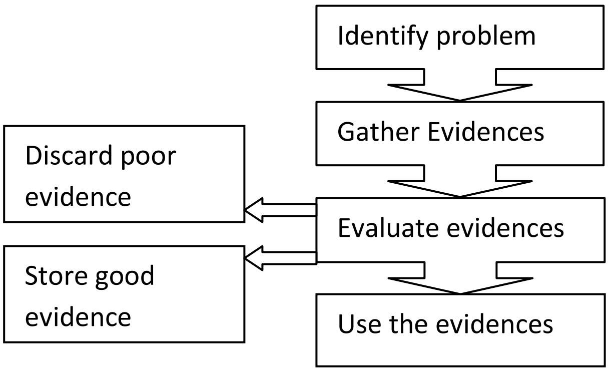

Fig 2 EBA in practice

The same can be used in software engineering since many a times software developers adopt a new technology because of their novelty or anecdotal evidence. In Evidence-based software engineering (EBSE) all the preliminary requirements of the preliminary phases of software development life cycle are fulfilled, here all the experiences are properly documented in order to inform software professionals to adopt a particular decision. In EBSE, the study factor is the technology of interest. The technological specifications should be very detailed and not at a very high level of abstraction that is the software lifecycle and all the design methods should be properly read and documented and only then should the engineer collect evidences on it and design the software generation model. Use of EBSE in the majority of relevant empirical studies, for example OO methods, Agile methods, or Cost estimation methods can be accomplished. To implement these following five steps of behavioral practices is followed; they are shown and explained in Fig 3:

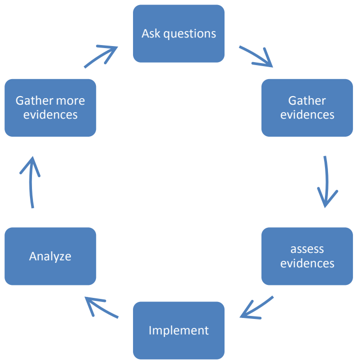

Fig 3. Steps of behaviroal practice

1. Convert a relevant problem or information need into an answerable question.

2. Search the literature for the best available evidence to answer the question

3. Critically evaluate the evidence for its legitimacy i.e. closeness to the truth, impact and applicability.

4. Incorporate the appraised evidence with practical experience to make decisions about which approach to practice.
5. Evaluate performance and seek ways to improve it.

6. Gather more evidences if required.

The first three steps require a systematic review of the literature, conducted in order to provide a balanced and objective summary that is relevant to meeting a particular need for information. Evidence-based approaches demand that, among other things, practitioners systematically track down the best evidence relating to some practice; critically appraise that evidence for validity, impact, and applicability; and carefully document it.

Based upon these practices a problem may be stated as: How can evidence or empirical based data help researchers in evaluating technologies in SE? What should be considered in order to gather relevant evidences? Based on these questions, researchers try to investigate the use of evidence as a tool to conduct robust empirical studies in SE. Mainly, the research topics investigated through the use of evidence has been identified as

- Software testing
- Software cost estimation
- Software effort estimation
- Generating test beds

A. Software testing

In software development, Testing is an important mechanism both to identify defects and assure that completed products work as specified. Software testing is a fundamental component of software quality assurance and represents a review of specification, design and coding. Even though extensive research has been done in the Testing field, it is necessary to assess the current state of research and practice, in order to provide practitioners with evidence that enable fostering its further development.

B. Software cost and effort estimation

Software is one of the most expensive components of a project, thus estimating accurate efforts required and its proportional cost of development of software is a very important and integral part of SDLC. Software cost estimation is the process of predicting the effort required to develop a software system. An accurate estimation is critical to both developers and customers. They can be used for generating request for proposals, contract negotiations, scheduling, monitoring and control. Underestimating the costs may result in underdeveloped functions, poor quality of software, and failure to complete on time. Overestimating may result in too many resources committed to the project. There are many models being practiced in software engineering which estimates the costs like the COCOMO II model developed by Barry Bohem where he used many software sizing models like Object Points, Function Points and source line of code.

The cost of software is majorly due to the human efforts inputted in it, thus most cost estimation methods focus on this aspect and give estimates in terms of person-months.

C. Generating test beds

It becomes a bit costly to test a software extensively thus test beds are generated which are an alternative to offer a good set of empirical tools to evaluate technologies in controlled environments.

Tore Dybå et al.[7] suggests that EBSE aims to improve decision making related to software development and maintenance by incorporating current best evidence from research with practical experience and human values. EBSE provides mechanisms to support various parts of Software Process Improvement. In particular, EBSE focuses on finding and appraising an appropriate technology for its suitability in a particular situation. There are several information sources that can be used. For example, getting viewpoints from the customers or the software's users, asking a colleague or an expert, using own experience, or search for research-based evidence. In research-based evidence reports, articles, and other documents that describe a study conducted and reported according to certain guidelines are included. The researchers here have used only the textual material available which does not prove to be sufficient enough.

Data sources like the Internet, Open forums and electronic databases should also be used for gathering data and seeking evidences. A large amount of data can be collected through these sources and it will then be classified under various heads and then will it be used for inference drawing.

Sackett *et al*.[8] recommend that individual doctors review the way in which they practice and teach Evidence Based Medicine in order to improve their individual performance. For EBSE, this would involve propagating successful technologies throughout a company and preventing the spread of technologies that are unsuccessful. In this paper adequate research was done in the field of medicine and the author suggests the same practices to be carried out in Software Engineering where engineers should first evaluate his decision based upon the evidences gathered and then bring it into practice. Adoption of new technology before the commencement of a new project should be based on the evidences gathered.

Though it is feasible in the field of medical science since individual doctors are responsible for reporting unanticipated side-effects of drugs, but when compared to software engineering in a competitive industry, barely any company will assist their competitors by reporting good and bad experiences with new technologies. Thus evidences should be gathered using the available literature and data gathered through primary and secondary data collection tools.

Gada kadoda et al [9] paper explores some of the practical issues associated with the use of case-based reasoning (CBR) or estimation by analogy. CBR is favored when there are are some discontinuities between effort and other independent variables. The authors thus believe there are a variety of decisions that must be made when utilizing CBR techniques. Such decisions include feature subset selection and choice of adaptation strategy. Empirical data is clubbed with experimental data to assist with the more effective use of CBR techniques for building prediction systems. This research allows estimators the freedom to utilize the features that they believe are best suited to their projects and is most appropriate to their environments. Consequently, the authors use Euclidean distance in *p*-dimensional feature space as a means of measuring similarity between cases. The categorical features are treated as either identical or completely dissimilar. adopt a simple analogy adaptation strategy of the mean of the *k* nearest neighbors', where *k* is an integer such that $0 < k < n$. When k=1 then the technique is a simple nearest neighbor method. As *k* tends towards *n* so the prediction approach tends towards merely using the sample mean. The choice of *k* is determined by the estimator.

Juristo et al [10] paper on Unit Testing Techniques states that testing is an invariable part of all systems engineering project. The human efforts that are spent on testing sometimes even exceed half the project total cost. Inadequate testing has been the source of many systems failures. Thus if test cases are designed keeping in view the real world practices then analysis of these can produce useful insights. Apart from mitigating defects in programs testing is performed in peripheral areas such as performance of software, reliability and security. Testing does not simply focuses on the technology alone but on socio-technical issues such as acceptability, usability and fitness of software. It focuses the whole systems rather than software in isolation. The authors did not attempt to create a new definition of testing in this paper, but rather to navigate this fuzzy concept by looking at what testing is and how it is meaningful from a practitioner point of view. For this study, the authors chose to use the definitions provided in the IEEE's Software Engineering Body of Knowledge—usually referred to as the Swebok. Juristo et al. refined and extended the Swebok classification system to include more detailed categories for code coverage and test set classification.

Wasif Afzal et al [11] paper uses Search-based software engineering (SBSE) as a tool for the application of optimization techniques in solving software engineering problems. This paper discusses the applicability of optimization techniques in solving software engineering problems. The growing interest in evidence based or search based search technique can be attributed to the fact that generation of software tests is generally considered as an undeniable problem, since there are many possible combinations of a program's input. The researchers here performed a SLR and came to the conclusion that metaheuristic search techniques have been applied for non-functional testing of execution time, quality of service, security, usability and safety. A variety of metaheuristic search techniques are found to be applicable for non-functional testing including simulated annealing, tabu search, genetic algorithms, ant colony methods, grammatical evolution, genetic programming and swarm intelligence methods.

Sarvnaz Karimi and Falk Scholer [12] credits systematic review as the key tool used in evidence based policy. The SLR's synthesizes available research on the topic of investigation. Though many researches hold their base by searching the literature but a SLR is carried out to agreed standards: using clear protocols in carrying out the process, focusing on specific questions, identifying as much of the relevant literature as possible, critically appraising the quality of the research included in the review; synthesizing research findings from included studies; being as objective as possible to remove bias, and, updating the review so that it remains relevant.

To get more test cases a mapping study can be done which systematically classifies the research papers and from them conclusions can be drawn. Evidence based datasets are also generated which can also be referred.

There is a severe decoupling between research in the computing field and the state of the practice in the field. The gap between empirical software engineering and software engineering practice might be lessened if more attention were paid to two important aspects of evidence. The first is that evidence from case or field studies of actual software engineering practice are essential in order to understand and inform that practice. The second is that the nature of evidence should fit the purpose to which the evidence is going to be put.

Thus a study of all the mentioned papers indicates that there is a lot of potential in the field of software planning and development using the EBSE approach. EBSE can also prove to be vital tool in cost and effort estimation for the software build using emerging technologies using Agile development, Component based software engineering and all the rest. The research paper also focuses on the use of evidence based approach for testing which includes test case generation and also determination of the best approach for testing using various techniques. The main intention of this research proposal is on all these areas primarily on testing.

- To find the best available Testing technique through a judicious and scientific approach.
- Software cost estimation/effort process is also benefitted if conducted through EBSE, thus we propose to go through case studies through which appropriate cost can be estimated.
- Assess /Validate the findings of the research.

Gathering evidences will be the primary source of data collection along with it we propose to read related literature, classify it according to the objectives based upon which the conclusions will be drawn. A mapping study provides a systematic and objective procedure for identifying the nature and extent of the empirical study data that is available to answer a particular research question.

For evidence based approach the most feasible approach is systematic literature review, mapping studies and evidence gathering.

The process of **Systematic Literature Review** is widely used to aggregate the results of primary studies. Conducting such a review involves an exhaustive search of the relevant literature, selection and data extraction. As shown in Fig 4 [7] all of this is conducted in accordance with a **review protocol**, which is composed ahead of performing the study and specifies in detail how the study is to be performed and the reasoning behind any choices.

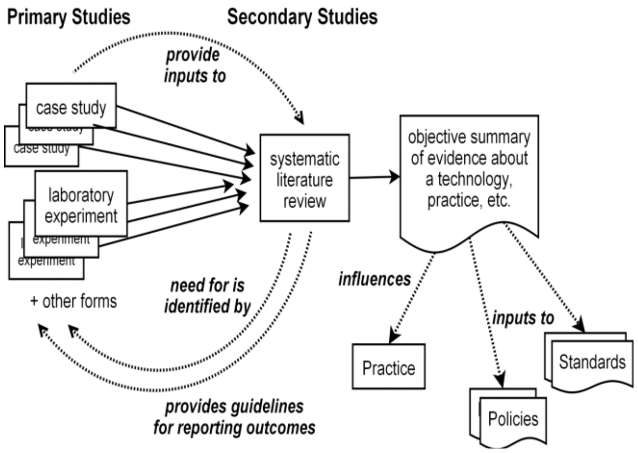

Fig 4. Working of SLR

In analyzing the available evidence the notion of '**best available evidence**' is used. This means that appropriate high quality experimental results are given more weight than those studies judged to be of a low quality. The analysis of evidence can be done using meta-analysis techniques and narrative synthesis techniques as may be appropriate to the forms of evidence available [13].

This research paper suggests that review protocol, consists of several steps. In order to examine the evidence of testing, cost estimation and effort estimation properties using metaheuristic search techniques, we suggest the following research questions:

Q1. In which testing, cost estimation and effort estimation areas have metaheuristic search techniques been applied?

After having identified these areas, we have three additional research questions applicable in each area:

Q1.1. What are the different metaheuristic search techniques used for testing each property?

Q1.2. What are the different fitness functions used for testing each property?

Q1.3. What are the current challenges or limitations in the application of metaheuristic search techniques for testing, cost estimation and effort estimation property?

Once the research questions have been identified the next step is generation of search strategy followed by Study selection criteria and procedures for including and excluding primary studies. Quality data can be used to devise a detailed inclusion/exclusion criterion and/or to assist data analysis and synthesis.

The result of a review is an objective summary of evidence about the technology or practice that was studied. The results of Systematic Literature Reviews can be used to inform, practice, policy, standards and to identify where new primary research is needed. Using evidence to make decisions about the adoption of technology and techniques ensures that those decisions are justified and can be shown to be based on sound research findings. Fig 5 [14] represents the methodology which is to be followed.

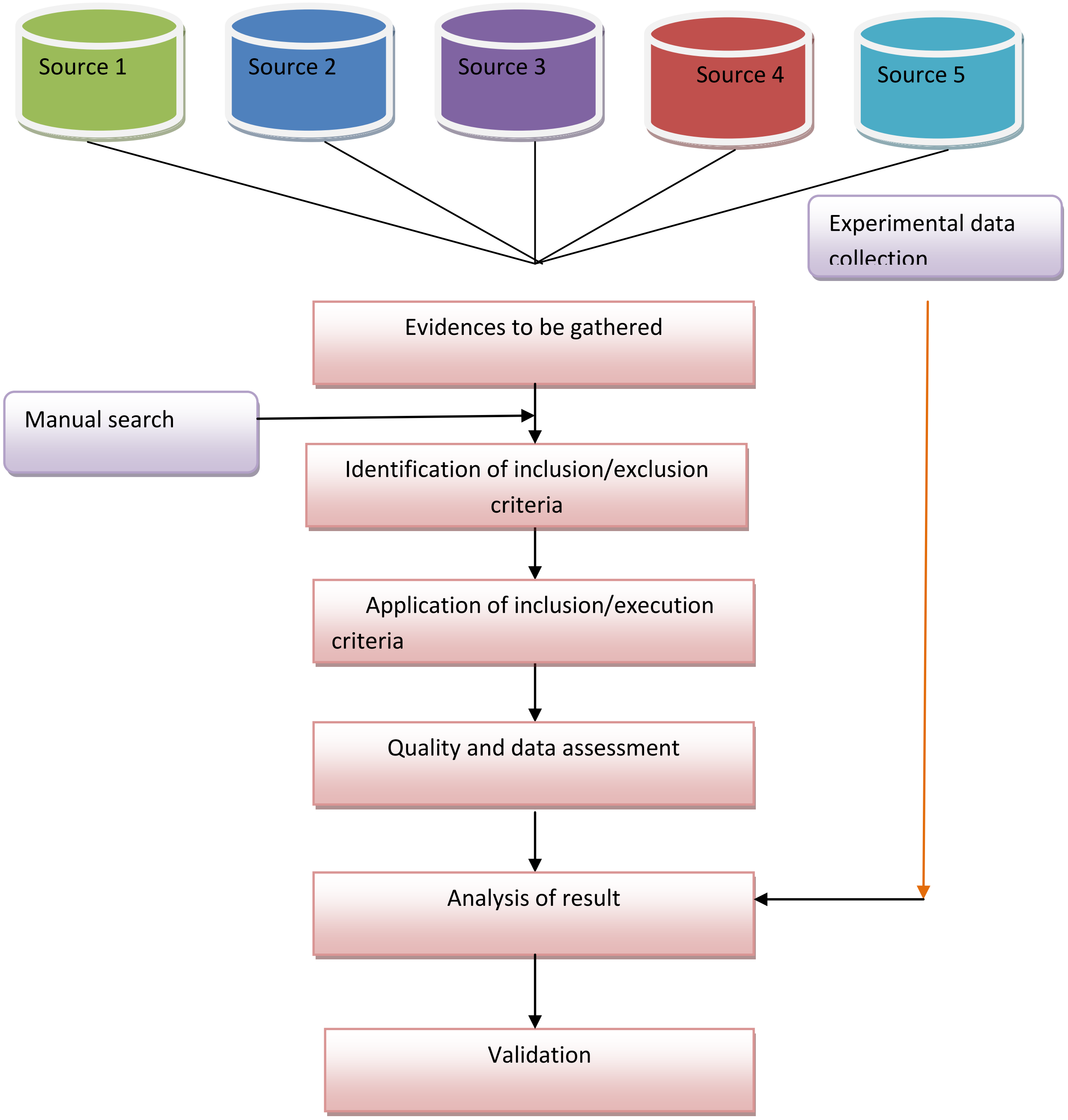
Source 1
Source 2
Source 3
Source 4
Source 5
Experimental data collection
Evidences to be gathered
Manual search
Identification of inclusion/exclusion criteria
Application of inclusion/execution criteria
Quality and data assessment
Analysis of result
Validation

Fig 5 SLR Methodology

4. REFERENCES

[1] www.Human brain.org

[2] Sackett, D.L.,Straus, S.E., Richardson,W.S.,Rosenberg,W., and Haynes,R.B. Evidence-Based Medicine: How to Practice and Teach EBM, Second Edition, Churchill Livingstone: Edinburgh, 2000.

[3] www.agile.csc.ncsu.edu

[4] B.A. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering, in: Proceedings of the 26th International Conference on Software Engineering, (ICSE'04), IEEE Computer Society, Washington DC, USA, 2004, pp. 273–281.

[5] M.V. Zelkowitz, D.R. Wallace, and D.W. Binkley, “Experimental Validation of New Software Technology,” Lecture Notes on Empirical Software Engineering, World Scientific, 2003, pp. 229–263

[6] www.dur.ac.uk/ebse

[7] B.A. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering: Proceedings of the 26th International Conference on Software Engineering, (ICSE'04), IEEE Computer Society, Washington DC, USA, 2004, pp. 273–281.

[8] Sackett, D.L., Straus, S.E., Richardson, W.S., Rosenberg, W., And Haynes, R.B. Evidence-Based Medicine: How to Practice and Teach EBM, Second Edition, Churchill Livingstone: Edinburgh, 2000.

[9] Gada Kadoda, Michelle Cartwright, Liguang Chen, and Martin Shepperd in Experiences Using Case-Based Reasoning to Predict Software Project Effort, 2000 - decgradschool.bournemouth. ac.uk

[10] Juristo, N., Moreno, A., Vegas, S. & Solari, M. (2006), 'In search of what we experimentally know about unit testing', IEEE Software 23(6), 72–80

[11] W. Afzal et al., A systematic review of search-based testing for non-functional system properties, Inform. Softw. Technol. (2009)

[12] Sarvnaz Karimi Falk Scholer SIGIR Workshop on " Supporting Complex Search Tasks"July 28, 2011, Beijing.

[13] B.A. Kitchenham, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep., EBSE-2007-001, UK (July 2007).

[14] Adapted from "A systematic review of search-based testing for non-functional system properties" Wasif Afzal , Richard Torkar, Robert Feldt