LESSONS LEARNED OVER 25 YEARS OF TESTING TECHNIQUE EXPERIMENTS. ARE WE ON THE RIGHT ROAD TOWARDS BUILDING A BODY OF KNOWLEDGE?

Abstract

Testing technique-related empirical studies have been performed for 25 years. We have managed to accumulate a fair number of experiments in this time, which might lead us to think that we now have a sizeable empirically backed body of knowledge (BoK) on testing techniques. However, the experiments in this field have some flaws, and, consequently, the empirical body of knowledge is far from solid. In this paper, we use the results of a survey that we did on empirical testing techniques studies to identify and discuss solutions that could lead to the formation a solid empirical BoK. The solutions we found are related to two fundamental experimental issues: (1) the rigorousness of the experimental design and analysis, and (2) the need for a series of community-wide agreements to coordinate empirical research and assure that studies ratify and complement each other.

1. Introduction

We thought that the best way to pay tribute to Vic would be to use facts to "empirically" underscore the role he has played in an Empirical Software Engineering (ESE) area or topic. The selected topic has been widely addressed in ESE: experiments on testing techniques. The reason behind this choice is that my PhD dissertation, co-supervised by Vic, was concerned with testing techniques and, specifically, the identification of knowledge on which to base tuned selections of testing techniques. My PhD research took me to the University of Maryland three times, where I spent 12 months all in all.

Part of the state of the art of my dissertation focused on reviewing experiments on testing techniques to identify what knowledge the results of these studies supplied that could help to select testing techniques. I was particularly concerned with what situations (fault type, personnel type, software type, project type) each technique best was for. From this review, we discovered that, even though there were a lot of empirical studies on testing techniques, the results of these studies were quite poor.

Additionally, one of the findings of my PhD dissertation was that there is a need for some, as yet unavailable, solid empirical knowledge on testing techniques to be able to gain a deeper understanding of testing techniques behaviour and make selections with a better technique/project match. Therefore, the results of the experiments on testing techniques are vitally important.

For these two reasons, we have subsequently extended the survey conducted for the state of the art of my thesis, trying to specifically identify what of all that we know about testing techniques has been empirically demonstrated. This work is described in (ref. JESE) The conclusion that we drew from this survey was that the results of the experiments conducted to date on testing techniques are not mature enough to provide a solid empirical BoK on testing techniques [. JESE].

This article aims to go a step further in this direction by analysing in detail what the reasons or problems underlying the empirical immaturity of the current BoK on testing techniques are. We also propose a series of problem-solving approaches and position the figure of Vic in this context.

For this purpose, the article has been organised as follows. Section 2 summarises the reviewed studies on testing techniques. Section 3 describes what problems we have detected that stand in the way of the formation of an empirical BoK on testing techniques. Section 4 lists a series of guidelines that would be of assistance for solving the detected problems. Finally, section 5 presents the conclusions.

2. Reviewed Empirical Studies on Testing Techniques

For readers who would like to refresh some notions of testing techniques, appendix A gives an overview of the techniques covered by the empirical studies that we have analysed. Table 1 shows the individual techniques of which the families covered by the reviewed studies are composed.

Table 1. Techniques by family

FAMILY	Random	Functional	Control flow	Data flow	Mutation
TECHNIQUE	- Pure random - Guided by the number of cases - Error guessing	- Equivalence partitioning - Boundary value analysis	- Sentence coverage - Decision coverage (branch testing) - Condition coverage - Decision/condition coverage - Path coverage	- All-definitions - All-c-uses/ - some-p-uses - All-p-uses/ - some-c-uses - All-c-uses - All-p-uses - All-uses - All-du-paths - All-dus	- Strong (standard) mutation - Selective (constrained) mutation - Weak mutation

Our literature search located 21 papers that describe 13 experiments on testing techniques.

It can be said that the interest in conducting experiments on testing techniques became widespread after Basili published his seminal paper in 1987 (ref). Until then almost all the studies in the testing techniques area had been theoretical. Only one other experiment on testing techniques had been published (by Myers), when Basili ran his experiment. We have not considered Myers' experiment in our survey, because it did not indicate exactly what techniques the subjects used. El de Hetzel a pesar de haberlo buscado en una gran variedad de

Fuentes (x, y, z) no has logrado dar con él [Sira es esto cierto? No tenías dónde lo había publicado? Cuéntamelo para ver qué pones]. Therefore, Basili's seminal experiment can be considered to be the first experiment published XXX? – with the experimental rigour required to ….

As our aim is to review the empirical studies designed to compare testing techniques and identify the main failings that stand in the way of the formation of a solid BoK, we have grouped the reviewed empirical studies into several subsets, taking into account which techniques they compare. Specifically, we have five groups of studies that can be divided into two classes:

- *Intra-family studies*, which compare techniques belonging to the same family to find out the best criterion, that is, which technique of all the family members should be used. We have identified:
    - Studies on the data flow testing techniques family. This family includes papers by Weyuker ([23] and [24]) and Bieman & Schultz studies ([4]).
    - Studies on the mutation testing techniques family. This family includes papers by Offut & Lee ([20] and [21]), Offut *et al.* ([18] and [19]) and Wong & Mathur ([25]).
- *Inter-family studies*, which compare techniques belonging to different families to find out which family is better, that is, which type of techniques should be used. We have identified:
    - Comparative studies between the control flow and data flow testing techniques families. This family includes papers by Frankl &Weiss, Weiss ([9], [10] and [11]), Hutchins et al. ([6]) and Frankl & Iakounenko ([13]).
    - Comparative studies between the mutation and data flow testing techniques families. This family includes papers by Frankl et al. ([7] and [8]) and Wong ¨& Mathur ([25]).
    - Comparative studies between the functional and control flow testing techniques families. This family includes papers by Myers ([16]), Basili & Selby ([1], [2] and [22]), Kamsties & Lott ([14]) and Wood et al. ([26]).

Table 2 lists the techniques examined by the above-mentioned experiments. Likewise, Table 3 indicates what aspects the experiments investigated.

Table 2. Techniques examined by reviewed experiments

		DATA FLOW TESTING		MUTATION TESTING			CONTROL FLOW VS. DATA FLOW			MUTATION VS. DATA FLOW		FUNCTIONAL VS. CONTROL FLOW		
		Weyuker [24]	Bieman & Schultz [4]	Offut & Lee [21]	Offut *et al.* [19]	Wong & Mathur [25]	Frankl & Weiss [11]	Hutchins *et al.* [13]	Frankl & Iakounenko [6]	Frankl *et al.* [8]	Wong & Mathur [25]	Basili & Selby [2]	Kamsties & Lott [14]	Wood *et al.* [26]
TESTING TECHNIQUE	All-c-uses	O												
	All-p-uses	O												
	All-uses	O					O		O	O	O			
	All-du-paths	O	O											
	Mutation (strong/standard)			O	O	O				O	O			
	MD EX-WEAK			O										
	MD ST-WEAK			O										
	MD BB-WEAK/1			O										
	MD BB-WEAK/n			O										
	2-selective mutation				O									
	4-selective mutation				O									
	6-selective mutation				O									
	Random selected 10% mutation					O					O			
	Constrained (abs/ror) mutation					O					O			
	Branch testing (all-edges)						O	O	O					O
	All-dus (modified all-uses)							O						
	Random (null)						O	O	O					
	White box													
	Black box													
	Boundary value analysis											O	O	O
	Sentence coverage											O		
	Condition coverage												O	

Table 3. Aspects analysed in reviewed experiments

		DATA FLOW TESTING		MUTATION TESTING			CONTROL FLOW VS. DATA FLOW			MUTATION VS. DATA FLOW		FUNCTIONAL VS. CONTROL FLOW		
		Weyuker [24]	Bieman & Schultz [4]	Offut & Lee [21]	Offut *et al.* [19]	Wong & Mathur [25]	Frankl & Weiss [11]	Hutchins *et al.* [13]	Frankl & Iakounenko [6]	Frankl *et al.* [8]	Wong & Mathur [25]	Basili & Selby [2]	Kamsties & Lott [14]	Wood *et al.* [26]
ASPECT STUDIED	Criterion compliance	X												
	Number of test cases generated	X	X	X	X		X	X						
	% mutants killed by each technique			X	X					X				
	No. of generated mutants				X	X								
	% generated sets that detect at least 1 fault					X	X	X	X	X	X			
	No. faults detected											X		X
	Time to detect faults											X	X	
	Time to detect faults/fault type													
	No. faults detected combining techniques													X
	Time combining techniques/fault type													
	No. faults found/ time											X	X	X
	No. faults isolated/hour												X	
	% faults detected/type											X	X	
	% faults isolated/type												X	
	Time to isolate faults												X	
	Total time to detect and isolate												X	
	% faults detected											X	X	X
	% faults isolated												X	

3. Detected Problems that Stand in the Way of the Formation of an Empirical BoK on Testing Techniques

For each experiment, we need to know whether it can generate a piece of knowledge. If it can, we will be able to form a body of knowledge incrementally by grouping empirically solid pieces of knowledge. The characteristics that we have analysed to decide whether an experiment contributes to the construction of a BoK can be grouped around three aspects. The three identified aspects, as well as the characteristics associated with each one are:

- Rigour when running an experiment. This aspect includes:
    - *Design rigour*. We have analysed how the experiment was designed. Specifically, we have taken into account how exhaustive and detailed a description has been made of the conditions under which the experiment took place.
    - *Data analysis rigour*. We have examined the data analysis techniques used to interpret the data collected during the experiment. We have paid special attention to how reliable the results yielded by these techniques are.
    - *Findings beyond mere data analysis*. We examined whether the experiments were confined merely to the results yielded by the data analysis techniques or, alternatively, higher-level findings related to the goals of the study were made.
- Level of correspondence between the experiment and the real world. This aspect would include the following characteristics:
    - *Use of meaningful programs and faults.* We have observed whether the analysed studies use programs and faults that are representative of the ones that occur in the real world. For this purpose, we have taken into account both the size and features of the programs, and the number and type of the faults.
    - *Response variable interest*. In this respect, we have taken into account the usefulness of the metrics used to collect the experiment data. Utility has been established depending on how useful the metric would be for practitioners.
    - *Realistic uses of the technique.* We also wanted to analyse how representative the running of the experiment is of reality. More specifically, during the testing process, the subjects apply the testing techniques (on their own or assisted by some sort of tool, if available) to later execute the test cases and be able to find the program defects. Rather than faithfully reproducing the process followed in the real world, however, many of the experiments studied here simulate reality without subjects.

- Establishment of a community-wide testing techniques experimentation strategy designed to ease the combination of experiments. This aspect includes the characteristics:
    - *Experiments are chained so that some take up and further investigate the findings of others.* We have examined how coherent each group of experiments is, that is, how much of a relationship there is between the different empirical studies of which each group is composed, and how they fit in with or complement each other.
    - *Methodological advancement in the experimentation sequence.* Another aspect of interest was to examine how logical the sequence of experimentation in a set of the same family of empirical studies was. For this purpose, we analysed both how many replications and what type of experiments were done and in what order.

Table 4 shows how far the analysed experiments comply with the above-mentioned characteristics. The light grey shading indicates that the studies do not take this characteristic into consideration very much at all. The darker grey shading signifies that the studies do bear in mind the characteristic to some extent. Finally, the black shading denotes that the studies very much take into account the characteristic. For a better understanding of the issues of each study listed in Table 4, readers are referred to (ref. Handbook Sira: el del Handbook es más largo? Entonces ese) . Sira: ¿No habías hablado de cambiar el gris claro por el blanco para más diferencia?

The whole table would have to be black for us to be able to consider that we have a solid empirical BoK on testing techniques. The paler the colours are, the more deficiencies we encounter. It is here, therefore, that there can be said to be a gap in the empirical knowledge on testing techniques.

Table 4. Study maturity by families

FEATURE	DATA FLOW TESTING		MUTATION TESTING			CONTROL FLOW VS. DATA FLOW			MUTATION VS. DATA FLOW		FUNCTIONAL VS. CONTROL FLOW		
	Weyuker [24]	Bieman & Schultz [4]	Offut & Lee [21]	Offut *et al.* [19]	Wong & Mathur [25]	Frankl & Weiss [11]	Hutchins *et al.* [13]	Frankl & Iakounenko [6]	Frankl *et al.* [8]	Wong & Mathur [25]	Basili & Selby [2]	Kamsties & Lott [14]	Wood *et al.* [26]
Experimental design rigour													
Data analysis rigour													
Findings beyond mere analysis													
Use of programs/faults representative of reality	N/A	N/A											
Response variables of interest to practitioners													
Real technique application environment is taken into account													
There are no topics remaining to be looked at or confirmed													
Existence of strict replications													
Experiment chaining													
Methodological advancement in experimentation sequence													

4. Guidelines for Maturing the Testing Technique Empirical BoK

Based on the issues identified in the last section, we have developed a series of guidelines that should help to form a solid BoK on testing techniques. These guidelines are described in detail in the following.

4.1. Rigorous Design (Sira: una cosa es el planteamiento y otra la explicación del mismo)

One essential feature of an experiment is that it should be able to be repeated under exactly the same circumstances. Exact replication is useful for establishing whether or not the results of an experiment are conclusive. For example, if the results of replicating an experiment differ from the earlier results, it will mean that these results were not irrefutable. In this case, the piece of knowledge in question should be reformulated, since, as it is set out, it is apparently difficult to demonstrate. Exact replications of an experiment are very important, as they corroborate or refute the results of the experiment. The fact that an experiment yields results should not be taken as evidence enough for these results to be considered a universal truth. It is the repetition of this at different places under if not equal then similar conditions that leads us to have more and more confidence in the results.

For an experiment to be replicated, the conditions under which the experiment was run need to be defined in full detail. This is the only way of assuring that the replication will be able to performed under the same conditions.

If an experiment is not defined with the necessary rigour, it will be difficult (and sometimes impossible) to replicate. This means that the study cannot be included in the empirical knowledge maturity chain, unfortunately making the study useless. By trying to replicate a study that is not very rigorously defined, we run the risk of getting different results from the earlier experiment the second time around and not being able to find out why. The problem could be either that the replication was not exact, that is, one of the conditions under which the experiment was run was changed, or that, indeed, the results of the earlier experiment were not conclusive. Unfortunately, we will unable to find out which one applies because of the poor description of the experimental design.

As Basili mentions in (ref. paper 5) and in (ref. paper 7), a fully defined empirical study should reflect its objectives, hypotheses, response variables, factors and their levels, parameters (or, in Basili's words, *high level hypotheses, detailed hypotheses, dependent variables, independent variables and context variables*, respectively), as well as experimental design and operation, together with sufficient documentation for replication.

From my personal experience with Vic, I can attest to just how important experimental design rigour is to him, as he was very much involved in the design of the experiment for my PhD research and insisted that I should do a thorough job.

Table 4 shows that the lack of rigour during experiment design is a fairly widespread problem[1]. Although Basili (ref.) describes the experiment conducted in full detail in his seminal paper, this rigor has only been emulated by the experiments run by Kamsties & Lott and Wood *et al.* The other experiments do not describe the experiments exhaustively enough for replication. The most common problems in the analysed empirical studies are omission of experimental designs and parameter descriptions. Few details are usually given about the factors and their levels, and only the response variables are explained thoroughly.

4.2. Rigorous Data Analysis

Another essential feature of an experiment is that the experimental data should be rigorously analysed. The way to achieve a reliable analysis is using data analysis techniques. Data analysis techniques can ascertain whether the results of the experiment are due either to the desired variations made to factors or to unrelated environmental factors.

If no such techniques are used to analyse the results of an experiment, we run the risk of misinterpreting the results, leading to mistaken findings. By just looking at the results of the experiment, the causes of the variation could be attributed to the factors when there could be other reasons behind this difference.

Although, in this case, we have not found a reference to anywhere where Vic explicitly said that data analysis techniques should be used to analyse the data collected in an experiment, we can easily extrapolate that it is an important question for him by observing his experiments (ref. este, Shull, mío, Guillherme), as he always uses formal analysis techniques.

From Table 4, we can see that, generally, around half of the studies do not back up their findings with statistical analyses, and the authors merely conduct a rough analysis based on the subjective interpretation of data.

4.3. Establishing and Linking High-Level Findings Beyond Mere Data Analysis to Objectives

To achieve empirical knowledge through experimentation, research goals need to be established that are succeeded by empirical studies (which range from controlled experiments

[1] In this respect, note that we have relaxed the constraint concerning what documentation an experiment should provide to permit replication, as we understand that a journal or conference paper is not the best place for this.

to case studies) that generate new knowledge (ref. QIP). This knowledge then generates new goals or research objectives to start a new cycle. In other words, empirical knowledge is gained by means of a series of successive goals/experimentation/results cycles (ref. paper 5). This means that an experiment should not stop at the mere analysis of the data output, but should also establish high-level findings from these data that can be traced back to the hypotheses and objectives of the experiment. In this manner, we could generate new hypotheses to refine the knowledge gained.

An experimenter who stops at analysing the data collected breaks this cycle, thereby preventing either the researcher or other groups from further investigating and empirically maturing this piece of knowledge. A single experiment will never be enough for a piece of knowledge to be well enough understood to become part of a solid BoK. Several trials will certainly be needed until it is well enough known. The failure to (ref. paper 7) "record findings and make recommendations" as Basili puts it, that is, the shortage of qualitative analysis, breaks the links in the chain of studies required to confirm a piece of knowledge.

Table 4 shows that this is an important shortcoming of the empirical studies. Only the studies by Weyuker and Basili & Selby establish high-level findings. These are followed by the studies by Kamsties & Lott and Wood *et al.* that are only partially beset by the problem. The other studies do not draw high-level conclusions that can be used to make recommendations and taken up in other experiments.

4.4. Using Programs and Faults that are Representative of Reality

Another essential feature of experiments is that they should be run using objects that are representative of reality. The high level focus of an experiment should be motivated by reality (ref. paper 7). From this we can conclude that the objects used during the experiment should reflect reality as closely as possible. Otherwise, the results of the experiment will not be useful. In the particular case of testing techniques, the programs used during the experiment, and the faults that they contain should be representative enough of reality for the results of the experiment to be meaningful.

It is very true that, because of their features, controlled laboratory experiments, which are the most abundant in Table 4, are smaller in size than other types of empirical studies or, as Basili said (ref. paper 7), are "microcosms of reality". This means that they are going to work on small programs. Even so, programs taken from programming books rather than real projects are often used. These programs contain few faults (often one or two) that have in many cases been entered artificially, that is, they are not faults that have really occurred in the program, but have been inserted afterwards. However, the results of the experiments would be more meaningful if both the programs used in the experiments and the faults they contain were closer to the real world.

In Table 4, we find that none of the studies deals with this problem satisfactorily. This is because most of the experiments examine very small programs, whose faults were entered artificially (never occurred in the real world) and with sometimes very few faults (at most one or two).

4.5. Using Response Variables of Interest

As already mentioned, the purpose of experiments in software engineering is to get a solid body of knowledge that can be used by practitioners. It is therefore important for the response variables of the empirical study to be of interest. When identifying the parameters, factors and response variables of an experiment, Basili suggests using the GQM (no hay que definirlo Goal Question Metric method?) (ref. GQM). The GQM is "a mechanism for defining and interpreting operational measure goals of experiments (no parece correcta la frase, comprobar" (ref. paper 7).

Looking at Table 4 we find that, unfortunately, not all the empirical studies focus on response variables that are of interest for practitioners. A clear example of this is the response variable *percentage of test cases that detect at least one fault*, examined in the two studies by Wong & Mathur and Frankl & Weiss, Hutchins *et al*., Frankl & Iakounenko and Frankl *et al*. In these cases, practitioners and researchers do not view technique effectiveness in the same way. Practitioners usually prefer alternative metrics for measuring effectiveness, such as the *number of defects* a technique detects or *ratio of detected faults to total faults*.

Generally, it is clear from Table 4 that there are studies, for instance, by Frankl & Iakounenko, Frankl et al. and Wong & Mathur, that examine no response variables that are of interest to practitioners, although most of the studies do examine some response variable that is of interest to practitioners. Finally, only the studies by Weyuker, Bieman & Schultz, Basili & Selby, Kamsties & Lott and Wood et al. focus exclusively on response variables of interest to practitioners.

4.6. Using an Application Environment in the Experiment that Resembles Reality as Closely as Possible

Returning to the idea of an experiment as a microcosm of reality, suggested by Basili (ref. paper 7), apart from using objects, such as, testing techniques, programs and faults, that are representative of reality in experiments, we should also consider that the actual running of the experiment should be representative of the real world. More specifically, the testing techniques should be applied or used as they are in the real world.

Again, as deduced from the process of formation of a BoK proposed by Basili, the experiments are motivated by reality (re. Paper 7). It is, therefore, essential for the experiment to represent reality as best it can.

As mentioned in the last section, the role of testing techniques during the testing process is for the subjects to apply them (on their own or with the aid of some sort of tool if available) to get a set of test cases that are then executed to find the defects of the program. The danger of running an experiment that does not resemble the real world closely enough is that we could get findings that cannot be extrapolated to this reality.

Looking at Table 4, we find that only three experiments, one by Basili, one by Kamsties & Lott and the other by Wood *et al.*, include real applications of the techniques under examination. This means that the empirical study includes a series of subjects who are responsible for applying the techniques in question. The other studies all automatically generate (there are no subjects involved) test cases, primarily at random, until they get a set that meets the constraints of the technique. In these cases, the empirical studies have overlooked a crucial concern, namely, the human factors. People apply techniques, and not all people are likely to generate the same test cases. Moreover, they are more likely to generate the cases following some sort of heuristic than at random.

4.7. Chaining Experiments

Basili mentions (ref. paper 7) that "combining experiments is necessary to build a BoK that is useful to the discipline". Indeed, the concept of families of experiments introduced by Basili (ref paper 2) is designed to assure that experiments do not occur in isolation but are related to each other. This will allow the empirical knowledge to grow and be consolidated.

The way to establish families of experiments is described in (ref. handbook) and in (ref. paper 2). According to Basili, the families of experiments are created when experiments are replicated. In this respect, Basili describes three types of replications:

1. Replications that do not vary any research hypothesis. Here we make a distinction between: strict replications and replications that vary the manner in which the experiment is run.
2. Replications that vary the research hypotheses. Here we make a distinction between: replications that vary independent variables, dependent variables or context variables.
3. Replications that extend the theory.

We have analysed the questions that remain to be looked at or confirmed in the reviewed empirical studies. It is interesting to note that all the results of the reviewed experiments need be confirmed. This can occur for two reasons: either because there are replications of an experiment that lead to contradictory results (as is the case in the group of experiments by Weyuker and Bieman & Schultz or in the group by Basili & Selby, Kamsties & Lott and

Wood et al.) or, otherwise, because there are no two groups of experiments with the same research hypotheses.

On the other hand, we find that, in all cases, the studies have questions awaiting confirmation, mainly because they are isolated studies, in which there are neither exact replications nor type-2 or type-3 replications.

Therefore, it is essential for the ESE community to reach agreement on and establish a more or less organised working framework that sets out the families of empirical studies that are needed for testing techniques so that the BoK grows in an ordered and organised manner, ruling out the now uncontrolled growth.

Additionally, to prevent the problem of replications, replication packages or what Vic calls "laboratory packages" should be created, a subject that Basili discusses in depth in (ref. paper 4) , (ref paper 2) or (ref. paper 1).

Sira: El tema de las replicaciones exactas lo tocas en dos sitios y no se entiende bien por qué. Como aquí hablas de encadenar, no de replicar deja este tema fuera de aquí y llévalo a 4.8 o al 4.1.

4.8. A Methodological Advancement in the Experimentation Sequence

Exact replication is not enough to consolidate a piece of knowledge. The control in the empirical studies needs to be gradually relaxed until the hypothesis is tested in the real world, outside the laboratory.

Basili describes the different levels of studies in (ref. paper 5), which he classes as experiments and observational studies. In experiments, "at least one treatment or controlled variable" exists and they are either controlled experiments (if they are run in vitro, that is, in the laboratory) or quasi-controlled experiments (if they are run in vivo, that is, outside the laboratory). On the other hand, "no treatment or controlled variables exist" in observational studies, and they are either case studies or field studies.

By establishing these levels of studies, it would appear that the advancement in the pursuit of knowledge should follow some sort of order. It makes no sense to conduct a case study without having run several controlled experiments, as cases studies are more costly and it is only reasonable that there should be some confidence in the truthfulness of the original hypotheses. Additionally, several replications of one and the same experiment will need to be run to get an acceptable level of confidence in a hypothesis.

Table 4 shows that the sequence in the advancement of the maturity of empirical knowledge as regards the study types conducted is fairly chaotic. Often, observational studies are conducted without having first run any more than one experiment (as in group X) or even straight off without having run any experiment at all beforehand (as in group Y).

In this respect, it would also be important for the ESE community to reach agreements to better combine the studies to assure that all the experimental efforts are useful and play a role in the construction of an empirical testing techniques BoK.

5. Conclusions

In this article, we have presented a survey of the problems concerning the experiments now existing on testing techniques. These problems prevent the 13 examined studies (divided into five families) from being analysed as a whole even informally, let alone processed using meta-analysis techniques.

First, we described the experiments taking into account the parameters that they investigate and the testing techniques they work on. Then, we presented the criteria for examining each experiment. Subsequently, we identified the issues affecting each experiment, taking into account these criteria. Also we established how frequently an issue crops up (very often, fairly often, not very often). Finally, we presented a series of guidelines that could fix the flaws in the experiments on testing techniques to date.

The identified guidelines refer to two important aspects in ESE: on the one hand, rigorousness as regards the experiment (this would include experimental design and analysis) and, on the other, the need to reach a series of community-wide agreements to coordinate empirical research and assure that the results yielded by the studies are ratified and complemented, enabling progress to be made in the formation of a solid and rigorous empirical BoK for testing techniques.

Basili has already pointed out many of these guidelines, and, indeed, most of the problems we have dealt with here are absent from this seminal experiment. This leads us to think that if the experiments existing in ESE were to have followed the guidelines set out by the experiments run by Vic, we would have much more mature and reliable knowledge on testing techniques today.

References

[1] V.R. Basili and R.W. Selby. Comparing the Effectiveness of Software Testing Strategies. Department of Computer Science. University of Maryland. Technical Report TR-1501. College Park. May 1985,

[2] V.R. Basili and R.W. Selby. Comparing the Effectiveness of Software Testing Strategies. *IEEE transactions on software engineering.* Pages 1278-1296. SE-13 (12), 1987.

[3] B. Beizer. *Software Testing Techniques*. International Thomson Computer Press, second edition, 1990.

[4] J.M. Bieman and J.L. Schultz. An Empirical Evaluation (and specification) of the All-du-paths Testing Criterion. *Software Engineering Journal*. Pages 43-51, January 1992.

[5] A. Davis. *Software Requirements: Objects, Functions and States*. PTR Prentice Hall. May, 1993.

[6] P. Frankl and O. Iakounenko. Further Empirical Studies of Test Effectiveness. In *Proceedings of the ACM SIGSOFT International Symposium on Foundations on Software Engineering,* pages 153-162, Lake Buena Vista, Florida, USA, November 1998.

[7] P.G. Frankl, S.N. Weiss and C. Hu. All-Uses versus Mutation: An Experimental Comparison of Effectiveness. Polytechnic University, Computer Science Department. Technical Report. PUCS-94-100. February 1994.

[8] P.G. Frankl, S.N. Weiss and C. Hu. All-Uses vs Mutation Testing: An Experimental Comparison of Effectiveness. *Journal of Systems and Software*. Volume 38. Pages 235-253. September 1997.

[9] P.G. Frankl and S.N. Weiss. An Experimental Comparison of the Effectiveness of the All-uses and All-edges Adequacy Criteria. *Proceedings of the Symposium on Testing, Analysis and Verification*. Pages 154-164. Victoria, BC, Canada. October 1991.

[10] P.G. Frankl and S.N. Weiss. Comparison of All-uses and All-edges: Design, Data, and Analysis. Hunter College, Computer Science Department. Technical Report. CS-91-03. March, 1991.

[11] P.G. Frankl and S.N. Weiss. An Experimental Comparison of the Effectiveness of Branch Testing and Data Flow Testing. *IEEE Transactions on Software Engineering*. Volume 19 (8). Pages 774-787. August 1993.

[12] R. Hamlet. Theoretical Comparison of Testing Methods. In *Proceedings of the ACM SIGSOFT '89 Third Symposium on Testing, Analysis and Verification.* Pages 28-37, Key West, Florida, ACM. December 1989.

[13] M. Hutchins, H. Foster, T. Goradia and T. Ostrand. Experiments on the Effectiveness of Dataflow- and Controlflow-Based Test Adequacy Criteria. *Proceedings of the 16th International Conference on Software Engineering*. Pages 191-200. Sorrento, Italy. IEEE. May 1994.

[14] E. Kamsties and C.M. Lott. An Empirical Evaluation of Three Defect-Detection Techniques. *Proceedings of the Fifth European Software Engineering Conference*. Sitges, Spain. September 1995.

[15] B. Latour, Woolgor D. *Laboratory Life. The Construction of Science Facts*. Princeton, USA: Princeton University Press, 1986.

[16] G.J. Myers. A Controlled Experiment in Program Testing and Code Walkthroughs/Inspections. *Communications of the ACM*. Vol. 21 (9). Pages 760—768. September 1978.

[17] G.J. Myers. *The Art of Software Testing.* Wiley-interscience, 1979.

[18] A.J. Offut, G. Rothermel and Zapf. An Experimental Evaluation of Selective Mutation. *Proceedings of the 15th International Conference on Software Engineering*. Pages 100—107. Baltimore, USA. IEEE. May 1993.

[19] A.J. Offut, A. Lee, G. Rothermel, RH. Untch and Zapf. An Experimental Determination of Sufficient Mutant Operators. *ACM Transactions on Software Engineering and Methodology*. Volume 5 (2). Pages 99-118. April 1996.

[20] A.J. Offut and D. Lee. How Strong is Weak Mutation?. *Proceedings of the Symposium on Testing, Analysis, and Verification*. Pages 200—213. Victoria, BC, Canada. ACM. October, 1991.

[21] A.J. Offut and S.D. Lee. An Empirical Evaluation of Weak Mutation. *IEEE Transactions on Software Engineering*. Vol. 20(5). Pages 337—344. August 1994.

[22] R.W. Selby and V.R. Basili. Evaluating Software Engineering Testing Strategies. *Proceedings of the 9th Annual Software Engineering Workshop.* Pages 42—53. NASA/GSFC, Greenbelt, MD. November 1984.

[23] E. Weyuker. An Empirical Study of the Complexity of Data Flow Testing. *Proceedings 2nd Workshop on Software Testing, Verification and Analysis*. Pages 188—195. Banff, Canada. July 1988.

[24] E.J. Weyuker. The Cost of Data Flow Testing: An Empirical Study. *IEEE Transactions on Software Engineering*. Volume 16 (2). Pages 121—128. February 1990.

[25] E. Wong and A.P. Mathur. Fault Detection Effectiveness of Mutation and Data-flow Testing. *Software Quality Journal*. Volume 4. Pages 69—83. 1995.

[26] M. Wood, M. Roper, A. Brooks and J. Miller. Comparing and Combining Software Defect Detection Techniques: A Replicated Empirical Study. *Proceedings of the 6th European Software Engineering Conference*. Zurich, Switzerland. September 1997.

Appendix A

This is not the place to exhaustively describe the features of testing techniques or their families, as this information can be gathered from the classical literature on testing techniques, like, for example [3], [17]. However, for readers not versed in the ins and outs of each testing techniques family, we will briefly mention each family covered in this paper, and the techniques of which they are composed, the information they require and what aspect of code they examine:

- *Random Testing Techniques*. This family of techniques proposes randomly generating test cases without following any pre-established guidelines. Nevertheless, pure randomness seldom occurs in reality, and the other two variants of the family, are the most commonly used.
- *Functional Testing Techniques.* This family of techniques proposes an approach in which the program specification is used to generate test cases. The component to be tested is viewed as a black box, whose behaviour is determined by studying its inputs and associated outputs. The key for generating the test cases is to find the system inputs that have a high probability of causing anomalous system behaviour. For this purpose, the technique divides the system inputs set into subsets termed equivalence classes, where each class element behaves similarly. The techniques of which this family is composed differ from each other in terms of the rigorousness with which they cover the equivalence classes.
- *Control Flow Testing Techniques.* Control flow testing techniques require knowledge of source code. This family selects a series of paths throughout the code, thereby examining the system control model. The techniques in this family vary as to the rigour with which they cover the code.
- *Data Flow Testing Techniques*. Data flow testing techniques also require knowledge of source code. The objective of this family is to select program paths to explore sequences of events related to the data state. Again, the techniques in this family vary as to the rigour with which they cover the code variable states.
- *Mutation Testing Techniques.* Mutation testing techniques are based on modelling typical programming faults by means of what are known as mutation operators (dependent on the programming language). Each mutation operator is applied to the program, giving rise to a series of mutants (programs that are exactly the same as the original program, apart from one modified sentence, originated precisely by the mutation operator). Having generated the set of mutants, test cases are generated to examine the mutated part of the program. After generating test cases to cover all the

mutants, all the possible faults should, in theory, be accounted for (in practice, however, coverage is confined to the faults modelled by the mutation operators).

The problem with the techniques that belong to this family is scalability. A mutation operator can generate several mutants per line of code. Therefore, there will be a sizeable number of mutants for long programs. The different techniques within this family aim to improve the scalability of standard (or strong) mutation to achieve greater efficiency.