A Survey on Software Requirement Models for Empirical Methods

Zafar Ali
Department of Computer Science
Federal Government University, VU Pakistan

Abstract*—*As technology advances existing theories are getting more and more complex to develop a model. Empirical methods are the only option in this situation. These methods are used for a variety of purposes. The software itself may need empirical methods. Empirical method might be needed for software engineering process analysis and improvement. In all cases, requirement models should be augmented to incorporate these empirical methods.

Keywords*—*software engineering; process layers; requirement model; empirical methods; evidence.

I. INTRODUCTION

Software requirement modeling is considered as the most important process step in a software development process [1]. A beginner for this process might think that, coding is the main part of software development. In reality, coding with the related documentation is considered to be only 30 percent or less of the whole process.

There are many models to standardize a software process. Out of them waterfall process model is most widely used. Following figure gives the process and product layers and links following this model [2].

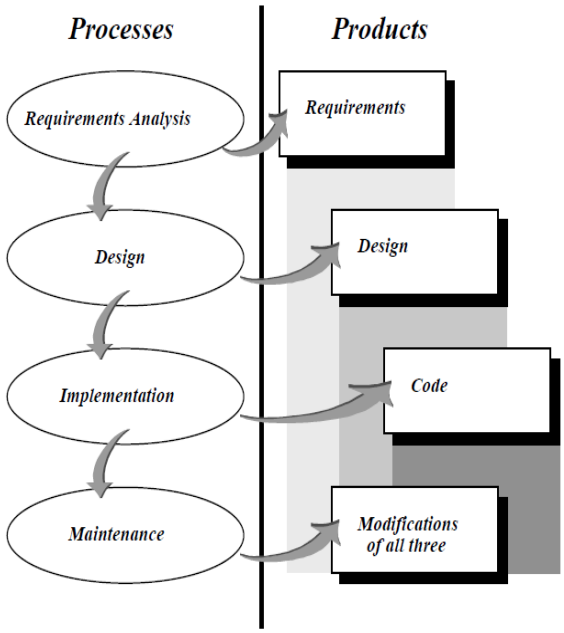

Figure 1: Process and product layers and links [2].

From the figure it can be seen that, 'Requirement Analysis' is the topmost layer in the process. As the name suggests, this is the analysis of requirement. This analyses the requirement or expectation or output from the software product. This is further detailed in the subsequent layers as the developers get more experienced.

Feasibility should be checked at this layer only. If the development is found infeasible at a later stage it will incur a great loss both in time, money and reputation. Requirement documents should be carefully prepared and clearly understood by all the team members. Any mistake or misunderstandings found at a later stage may jeopardize the product.

To avoid such losses the document should have as much details as possible. Earlier concept was to document what is expected from the software. Presently, how to fulfill that expectation is also documented here. Specifications are listed in this document.

During or after development, client's expectation might change. This necessitates a revisit to this layer. So in this waterfall, water moves down and some water moves up as well.

Empirical methods are the methods that can analyze experimental data or real-life observation. Empirical methods are needed for many reasons. One of the reasons is to estimate parameters based on data collected in previous projects. Following figure gives an example of use of empirical method to complete requirement model.

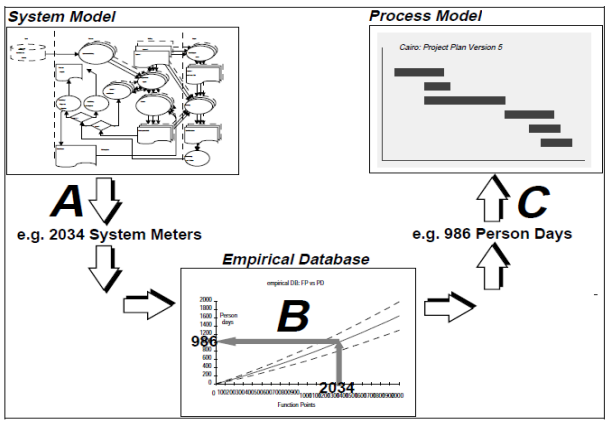

Figure 2: The generic ABC requirement model development strategy [2].

Above figure explains the way to decide the human resource required for this project. For all previous projects the actual workload and corresponding resource requirement can be recorded. With this information an empirical plot can be made. For the present project the workload is 2034 unit. This corresponds to 986 person days in the plot. From this information duration and cost of the project can be estimated.

Empirical methods are also essential when the software product deals with experimental data. For example, automated software for medical diagnosis might use the data of past cases. Statistics are applied to frame the rules for diagnosis. Often the developers have inadequate experience on statistical analysis. This leads to error. Extensive research is going on to develop standard requirement models to avoid those mistakes.

II. RELATED WORK

Conradi and Fuggetta [3] analyzed software improvement process in general. When such structure is applied for empirical methods it does not give good results.

Kitchenham et al. [4] framed some guidelines for correct use of empirical methods. It was observed that, statistical analysis fails because of inadequate knowledge of statistics. For decision making, it is always better to have a theoretical model. The point was illustrated with an interesting example. Dr. Jenner developed the small pox vaccines from the antigens of cow pox. He succeeded as he could develop the theory from observation that, women who get cow pox do not get small pox. Such cause effect relations should be framed for empirical methods.

Lowgren [5] compared two aspects of design namely, engineering and creative. Following table lists the differences. Software involving empirical methods often falls in the category of creative design.

Table 1: Comparison of two aspects of design [5]

	Engineering Design	**Creative Design**
Process as a whole	Entirely convergent	Has divergent aspects
Key question	How: determining how to solve problem	Why: is this the right problem to solve
Stages in the solution	Sequential: one candidate solution is refined	Parallel: many alternatives are explored
Nature of process	Analytical: can be described as structured	Creative: inherently unpredictable
Purpose	One satisficing solution	Explore possibilities before committing
Ownership	Impersonal: designer is an objective instrument	Personal: the designer is "Present" in the design; assumes responsibility for social action

Van Domburg [6] discussed about their project 'QUADREAD' in details. The project goal is to develop a tool that takes care of many existing problems in software development. The first phase of the project focuses on requirement models. One important point is traceability. It is defined as the degree to which a relationship can be established between two or more products of the development process.

In the context of requirement engineering there are two types of traceability: pre-requirement and post-requirement. Also there are intra-level and inter-level traceability. Following figure illustrates different types of traceability.

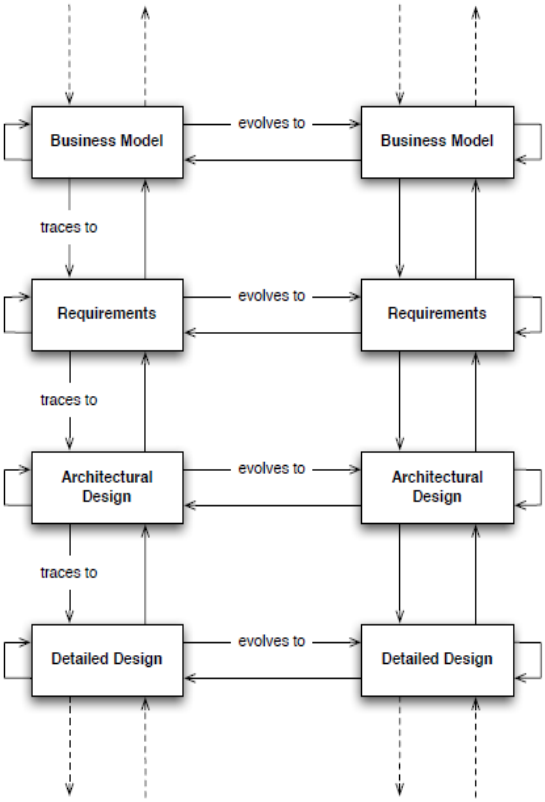

Figure 3: Traceability in software development [6].

Various standards for requirement is listed in this work. The standard followed for the project is 'IEEE Std 830-1998'.

Table of Contents

I. Introduction
   1) Purpose
   2) Scope
   3) Definitions, acronyms and abbreviations
   4) References
   5) Overview
II. Overall description
   1) Product Perspective
   2) Product functions
   3) User characteristics
   4) Constraints
   5) Assumptions and dependencies
III. Specific Requirements

Appendixes

Index

Figure 4: Prototype software requirements specification outline [6],

Author did a study on existing tools for software requirement documentation. Following figure gives a comparison of these tools.

Table 2: Comparison of tools for requirement documentation [6].

	Microsot Excel 2003	IBM Rational Requisite Pro 7.1	TRIC 1.1.0
Intelligence	Low	Medium	High
Maturity	Mature	Mature	Prototype
Requirements management	Ad-hoc	Supported	Supported
Requirements relations management	Ad-hoc	Yes	Yes
Traceability matrix	Yes	Yes	Yes
Displaying inferred relations	No	Yes, based on transitivit y only	Yes based on reasoning
Displaying inconsistencies	No	No	Yes
Explaining reasoning results	No	No	Yes

Though TRIC is at prototype level it gives better results for dependent requirements. None of these tools can handle dependencies of empirical methods. Further studies show that the feature is missing due to lack of expertise in this area.

Batra and Bhatnagar [7] did an extensive literature review on requirement models. The review shows model lacks maturity for empirical methods. There is enough scope for improvement. Authors developed the desired features that the requirement model should have. Following figure shows those features.

Figure 5: Desired features in requirement engineering process model [7].

It can be observed that, any solution for existing challenges associated to empirical method is missing in this desired features list.

Bai *et al.* [8] made an extensive literature review in the area of empirical research in Software Process Modeling (SPM). Significant publications were selected and analyzed. Following figure gives the year wise distribution of the publications.

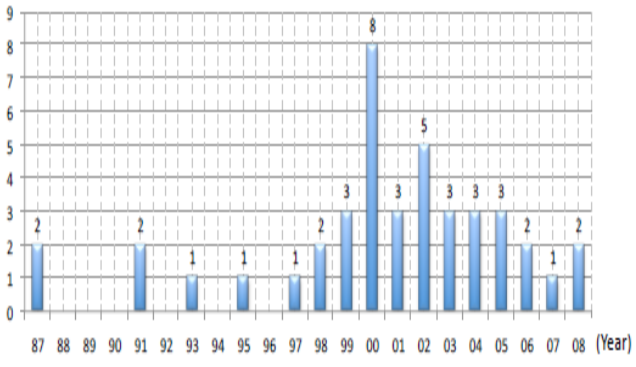

Figure 6: Distribution of publications [8].

According to the authors there is a significant continuous rise in publications since the year 1999.

They attributed it to the availability of suitable tools. Author of this paper observes that, there is a peak in the year 2000. After that there is no rise. The credit should be given to the global economic boom around the year 1999.

Following Research Questions (RQ) were checked in the papers.

Table 3: Research Questions (RQ) or objectives of the paper [8].

RQ	Attribute	Description
N/A	Basic Info	Title, Authors, Publication, Year, Venue etc.
RQ.1	Primary Objective	The primary objective of the selected study
RQ.2	Techniques	Which SPM technique did the selected study employed or evaluated?
RQ.3	Research Method	The empirical research methods/data analysis techniques used in the selected study
RQ.4	Rigor Assessment	How rigorous is the selected empirical study with regard to the empirical research guideline?

Following table gives different categories for RQ.1 or primary objectives.

Table 4: Primary objectives types [8].

Primary Objectives	No. of Studies	Mostly used research methods
Process Understanding	11 (25.6%)	Case Study (72.7%)
Process Management and Improvement	6 (14.0%)	Action Research (80.0%)
SPM Technique Evaluation	18 (41.9%)	Case Study (66.7%)
Other	9 (20.9%)	Phenomenology (77.8%)

From the table it can be observed that, process management and improvement gets least attention. According to the authors this should get maximum attention. There are several probable reasons. There could be difficulties in the deployment of processes due to organization politics. Collection of data might be difficult due to confidentiality. The activity might get the lowest priority.

Following figure gives the analysis for RQ.2 or techniques.

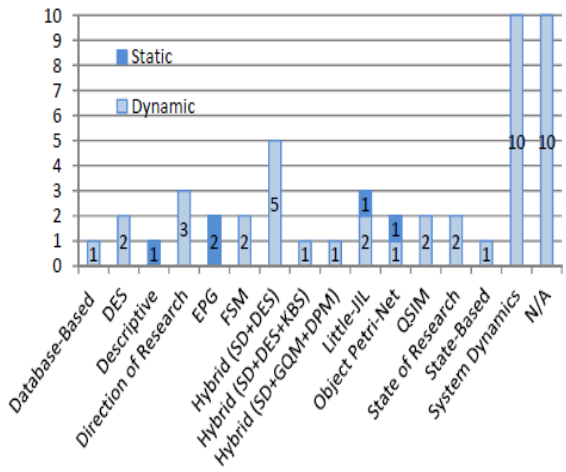

Figure 7: The number of studies reporting different techniques [8].

15 different techniques were found. Following table gives the paper distribution for RQ.3 or research method used.

Table 5: Empirical research methods types [8].

Empirical Research Method	Data Analysis		Total	%
	Qualitative	Quantitative		
Action Research	12	-	12	27.9%
Case Study	13	4	13*	30.2%
Experiment	4	1	4*	9.3%
Meta Analysis	3	1	3	7.0%
Phenomenology	9	1	9	20.9%
Survey	3	2	3	7.0%
Sub Total	43	9	43*	-
* one study claimed using both case studies and controlled experiment				

It can be observed that analysis were mainly qualitative. When RQ.4 or Rigor Assessment was analyzed it was observed that, 2 most important assessments were missing in most of the reported publications. Following table gives the percentage of papers with these problems.

Table 6: Number of papers with missing assessment [8].

Section	No. of Studies	Percentage
All Studies	43	
Research Design	6	13.95%
Execution	10	23.26%

In future work this point should get its due attention.

Kitchenham et al. [9] developed the guidelines for empirical methods with a comparison with the well established processes used in medicine. The outcome of the empirical method is called evidence. Following table gives the evidence based medicine steps that can be compared with software engineering steps.

Table 7: Steps used in medicine [9].

Step	Evidence-based Medicine
1	Converting the need for information (about prevention, diagnosis, prognosis, therapy, causation, etc) into an answerable question
2	Tracking down the best evidence with which to answer that question
3	Critically appraising that evidence for its validity (closeness to the truth), impact (size of the effect), and applicability (usefulness in our clinical practice).
4	Integrating the critical appraisal with our clinical expertise and our patient's unique biology, values and circumstances
5	Evaluating our effectiveness and efficiency in executing steps 1-4 and seeking way to improve them both for next time

Table 8: Analogous steps in software engineering [9].

Evidence-based Software Engineering
Converting the need of information (about development and maintenance methods, management, procedures etc.) into an answerable question.
Tracking down the best evidences with which to answer that question
Critically appraising that evidence for its validity (closeness to the truth), impact (size of the effect) and applicability (usefulness in software development practice)
Integration the critical appraisal with our software engineering expertise and with our stackholders' value and circumstances.
Evaluating our effectiveness and efficiency in executing steps 1-4 and seeking way to improve them both for next time

The work is useful for medical practitioners who are willing to contribute in software engineering domain. In the similar line a technical report was made [10] to find similarities between software engineering and other disciplines.

Table 9: Agreement of methodologies [10].

Discipline	Comparison with SE (1 is perfect agreement, 0 is complete disagreement)
Nursing & Midwifery	0.83
Primary Care	0.33
Organic Chemistry	0.83
Empirical Psychology	0.66
Clinical Medicine	0.17
Education	0.83

The analysis shows clinical medicine has the least agreement with software engineering.

III. ANALYSIS AND CONCLUSION

- As technology is advancing associated theory is getting more and more complex. Empirical study is a must.
- This is true for VLSI technology. For simulation of characteristics every device or component in the design should have a correct model. The accuracy of simulation is predominantly decided by the accuracy of the models. For simpler devices theoretical models were developed. But for complex devices empirical models are necessary.
- Data collection is an important part of empirical methods. Data could be collected by human beings or by an automated system. The latter option is better. The former option is expensive and error prone.
- In an automated environment care should be taken to avoid erroneous data. This can be understood from an interesting fictitious situation.
  - A survey is conducted to find out the effect of smoking on two medical problems prevalent in a village.
  - There is a strange culture in the village. All males smoke and the females do not smoke.
  - Villagers can select the sex as male / female in the automated data collection system.
  - They can answer for two medical problems, whether experienced or not with yes / no.
  - Following table gives the answers for 4 villagers.
  - Now a correlation analysis is done to find out whether smoking creates prostate enlargement.
  - The records show some smokers (males) have this problem and all non-smokers (females) do not have this problem.
  - Obvious inference will be smoking creates prostate enlargement.

- Similarly another drawn conclusion is: smoking prevents miscarriage.
- What leads to such wrong results?
- Firstly, the answers should not be in yes / no form. 'Not Applicable' option should also be there.
- Secondly, the data analysis expert should know about the local culture.

- The problem is addressed by Zave and Jackson [11] as one of the dark corner. The guideline given is: "All the terminology used in requirements engineering should be grounded in the reality of the environment for which a machine is to be built". Some of the software engineering terminologies may sound unfamiliar to workers with different background. Data collector may not be able to map culture in this scenario to environment in the guideline for the dark corner.
- Data might be confidential. It can only be processed automatically. If there is any discrepancy human intervention is not allowed.
- Confidential data could be in the encrypted form. It is a challenge to develop statistical analysis tool that can correlate encrypted data.
- As discussed in the previous section some researchers think that, there should already be a cause-effect relation before correlation of data is studied. There is a dark side of it. Already proposed relation might make the researchers biased. There could be a tendency to hide data that contradicts this proposed relationship.
- There is a general agreement that the area demands focused research.
- Metrics development is very important to bring quality. Existing metrics need refinement for empirical methods. Additional metrics are required. This is discussed for the metric 'traceability' in the previous section.

Table 10: A fictitious data record

	Villager 1	**Villager 2**	**Villager 3**	**Villager 4**
Sex	Male	Female	Female	Male
Do you smoke?	Yes	No	No	Yes
Do you have prostate enlargement?	Yes	No	No	Yes
Have you ever experienced miscarriage?	No	Yes	Yes	No

IV. REFERENCES

[1] D. Pandey, U. Suman, and A.K. Ramani, "A framework for modelling software requirements", International Journal of Computer Science Issues, vol. 8, issue 3, no. 3, pp. 164 – 171, May, 2011.

[2] S. Moser, "Measurement and Estimation of Software Processes", PhD Thesis, University of Berne, Switzerland, pp. 1 -31, November, 1996. Available: http://scg.unibe.ch/archive/phd/moser-phd.pdf viewed on 30th June, 2016. (*references*)

[3] R. Conradi and A. Fuggetta, "Improving software process improvement", IEEE Software, pp. 2 – 9, July / August 2002.

[4] B.A. Kitchenham, S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K. El-Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering", National Research Council, Canada, pp. 1 – 27, January, 2001.

[5] J. Lowgren, "Applying design methodology to software development". in *Proc. ACM Conference on Designing Interactive Processes, Practices, Methods and Techniques*, DIS 1995, pp. 87 – 95.

[6] R.S.A. Van Domburg, "Empirical evaluation of change impact predictions using a requirements management tool with formal relation types", M.S. Thesis, University of Twente, Enschede, pp. 1 – 48, November, 2009. Available: https://www.utwente.nl/ewi/trese/graduation_projects/2009/VanDomburg.pdf viewed on 15th July, 2016. *(references)*

[7] M. Batra and A. Bhatnagar, "Descriptive literature review of requirements engineering models", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 2, pp. 289 – 293, February, 2015.

[8] X. Bai, H. Zhang, and L. Huang, "Empirical research in software process modeling: a systematic literature review", in *Proc. IEEE International Symposium on Empirical Software Engineering and Measurement*, September, 2011, Banff, A.B., pp. 339 – 342. (*references*)

[9] B.A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence based software engineering", in *Proc. 26th International Conference on Software Engineering*, ICSE 2004, IEEE, pp. 1 – 9.

[10] B.A. Kitchenham, Guidelines for performing systematic literature reviews in software engineering", EBSE Technical Report, EBSE-2007-01, pp. 1 – 57, Department of Computer Science and Mathematics, Keele University, and Department of Computer Science, University of Durham, UK, 2007.

[11] P. Zave and M. Jackson, "Four dark corners of requirement engineering", ACM Software Engineering and Methodology, vol. 6, no. 1, pp. 1 – 30, January, 1997.