J. Appl. Environ. Biol. Sci., 6(8)88-97, 2016

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Trends in Automated Software Engineering: A Systematic Mapping Study

Umar Faiz, R. B. Faiz, Muhammad Saud

Faculty of Computing, Riphah International University, Islamabad, Pakistan

Received: April 15, 2016
Accepted: June 7, 2016

ABSTRACT

A secondary study provides review of primary studies to address a specific research question based upon integrated evidence. Systematic review and systematic mapping study are two approaches that vary in terms of goals but adopt a well-defined protocol to investigate a research question. A systematic map interprets the research area under study through a general map (using diagrams, charts and statistics). The purpose of a systematic mapping study (scoping study), a variant of Systematic Literature Review (SLR), is to structure a field of interest by building a classification [1, 2]. This paper systematically maps automated tools on to existing areas of SWEBoK 2004 [3].The aim of this research is to show trend of automated tools in areas of SWEBoK from 2007 till 2013 in 12 journals. Such trend analyses will identify gaps for conducting primary studies and clusters where SLR's can be conducted.
KEYWORDS: SWEBok, software engineering; systematic mapping study; automated tools.

1.INTRODUCTION

"An empirical study is really just a test that compares what we believe to what we observe" [4]. Due to increased awareness and availability of several useful guidelines including [5-8]; the number of empirical studies in software engineering research has always been on the rise [9].

Related studies (not necessarily systematic mapping studies) have been carried out in the past to determine the status of Empirical Software Engineering (ESE). Unfortunately, all these studies were either focused on a particular journal or a particular research method, thereby undermining their credibility and validity.

Sjøberg [10] presented a survey result of 5,453 articles published in 9 leading software engineering journals and 3 conferences covering the period 1993-2002. The survey was designed to study quantitatively the practice of conducting controlled experiments on different topics and their subjects in software engineering. The authors assessed the extent to which information collected from the controlled experiments is reported. The survey revealed that 103 articles reported controlled experiments in which one or more software engineering tasks were performed. As the authors have put it in their paper, "a controlled experiment is a randomized experiment or a quasi-experiment in which individuals or teams (the experimental units) conduct one or more software engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages, or tools (the treatments)" [10]. Though the survey pivoted around the controlled experiments, the effort unconsciously approached SWEBok sub-category of Software Engineering Process Tools. The scope of this survey was narrow as it dealt only experiments.

Segal [11] investigated the nature of the evidence in the Journal of Empirical Software Engineering published between 1997 and 2003 (119 articles). The investigation was conducted to determine whether the evidence gathered from case or field studies of actual software engineering practice helps understand and

***Corresponding Author:** UMAR FAIZ, Faculty of Computing, Riphah International University, Islamabad, Pakistan.

inform that practice. The authors argued that such an understanding can play a pivotal role in helping practitioners use the evidence collected by the researchers as useful information. Since this study only covers one journal, the findings cannot be easily generalized. However, this study hinted at use of measurement/metrics, maintenance, review and inspection accounting in the software engineering process.

Another study by Höfer [12], covering longer period January 1996 to June 2006, provided an assessment of dominant empirical studies that mainly consisted of experiments and case studies focusing on measurement/metrics and tools/methods/frameworks. The authors established that case studies conducted by professionals (80%) outnumber those by students, while, students dominated studies with controlled experiments. However, the study suffers from the same credibility issue as it covers only the articles published in the Journal of Empirical Software Engineering and exhibits the use of metrics/measurements for case and field studies.

To overcome above limitations, our mapping study categorizes automated tools published in 12 journals from 2007-2013 into existing areas of SWEBoK. The remainder of this work has been organized as follows: Section 2 presents the systematic mapping process. In Section 3 we present results and analysis and threats to validity of this study. Finally, we share our conclusions in Section 4.

2. Systematic Mapping Process

2.1 Protocol development

A systematic mapping study protocol has a detail plan for each step in the mapping study. A protocol specifies the methods used to conduct a mapping study and is important to reduce the possibility of researcher bias [17]. The rationale for the mapping study has already been discussed in the background section. Our mapping study protocol is based upon guidelines proposed by Kitchenham [13-16] which includes: protocol development, time-table, definition of research questions, conducting search, screening search results to finding relevant papers, classification of relevant papers and building systematic map. The section 2.3 discusses the research questions. The sub-sections 2.4.1, 2.4.2 and 2.4.3 discuss the strategy to develop the search string, identification of sources for candidate studies, identification of primary studies and setting excluding/including criteria and the strategy to resolve differences between the evaluators. Data extraction techniques from the primary studies have been discussed in section 2.5.

2.2 Time table

Table 1. Time-table for Systematic Mapping Protocol

Time	Systematic Mapping Process Steps	Outcome
10 Weeks	Conduct Search	All Papers
20 Weeks	Screening Papers	Relevant Papers
12 Weeks	Data Extraction & Classification	Classification Scheme
08 Weeks	Mapping Process	Systematic Map

2.3 Research question

The following question motivated us to conduct this study:

RQ: What is the trend of automated tools reported in areas and sub-areas of SWEBoK?

Automated tools selected from literature (12 journals) over a period of five years (2007-13) were mapped over areas and sub-areas of SWEBoK 2004.

2.4 Search strategy

There are two levels of the possible search strategy to be applied as part of the review:

2.4.1 Level 1: Search engines

The main challenge was to come up with such quality search string which filter papers reporting tools. The formulation of the search string included derivation of the major terms used in the review questions, keywords mentioned in the article, using the synonyms and alternative words, Boolean OR to incorporate alternative synonyms and using the Boolean AND to link major terms.

The complete search term initially used:

Software automated tools OR software engineering automated tools

A very limited number of results retrieved when using the initial string, thus a much complex string were derived. Then refined search string was:

(Software automated tools OR software automation OR software automation using tools AND (approach) OR (technique) OR (method)

The query yielded a large number of instances in the citation databases with the results not even related to area of software engineering.

2.4.2 Level 2: Specific proceedings and journals

Based on valuable feedback from colleagues and anonymous reviewers, we decided to conduct manual search that included all published papers (2007-2013) in twelve software engineering journals. Automated Software Engineering (ASE) journal from Springer is the only journal by scope dedicated for automated software engineering since 1994. Thus, ASE should give good indication of types of automated software engineering tools recently reported in literature.

Our manual search of twelve journals between 2007 and 2013 resulted in 4565 papers out of which 496 relevant papers were found. Data extracted from these papers was systematic mapped upon various areas and sub areas of SWEBoK. We created a separate EndNote [18] database for each journal. This allowed us to maintain complete bibliographic record of all relevant and irrelevant papers. Such systematic mapping present automated tools in various areas and sub areas of SWEBoK [3].

It is important to mention here that the classification of the candidate studies was carried out according to SWEBoK version 2004 areas of software engineering. Close to the submission of this work, an official release of SWEBoK 3.0 was released with adding new knowledge areas (KAs) and revising others.

2.4.3 Selection criteria

Another important step in systematic mapping study is to include and exclude research papers based on selection criteria.

Inclusion Criteria: Research papers which completely automate some SWEBoK area in the form of a tool.

Exclusion Criteria: Papers that report partial automation i.e. incomplete automation or it requires manual processing or the papers that lie outside the domain of SWEBoK.

Papers were selected by two researchers, which were then randomly verified by another senior researcher. The two researchers discussed papers in which they had difference of opinion and unanimously agreed on inclusion of relevant papers into existing software engineering tools categories defined by SWEBoK 2004 [3].

Table 3 shows that Journals with automation as a part of their intended scope, i.e., Automated Software Engineering, have the second highest percentage of automated tools reported in literature. Similarly journal with automation as not a part of their intended scope, i.e., Empirical software engineering and IEEE transactions on software (even though include software in their scope) has the lowest percentage of automated tools reported in literature. This reconciles our hypothesis that the journals whose scope does not include automation should exhibit low relevancy of tools. However, it was interesting to note that the journal with automation not part of its intended scope, i.e., Innovations in Systems Software has the highest percentage of reported tools.

2.5 Data extraction and classification

Our systematic mapping study was restricted to reviewing titles, abstracts and keywords. All information extracted from the candidate studies was then classified in to relevant categories through following steps:

1) Relevant papers were identified.

2) Data related to the tools was extracted using data extraction template to classify, extract and record important information about the tools from each paper.

3)This information was synthesized and tools were mapped to various areas and sub areas of SWEBoK.

4)Since we used EndNote database for local storage of citations & abstracts; papers placed inside "relevant" group worked for us as extracted data. We used "Notes" field in our EndNote database to save classification of each paper.

3 Results and analysis

Below mentioned set of graphs from fig.1-fig.28 shows the number of automated tools per year on y-axis against the period of study (2007-2013) on x-axis in, 12 journals, in each area of SWEBoK.

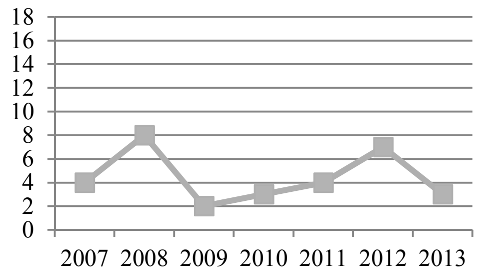

Fig.1. Requirement Modeling

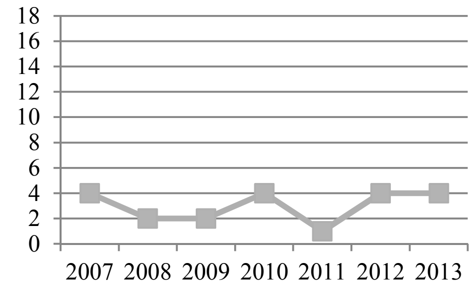

Fig.2. Requiremment Tracebility

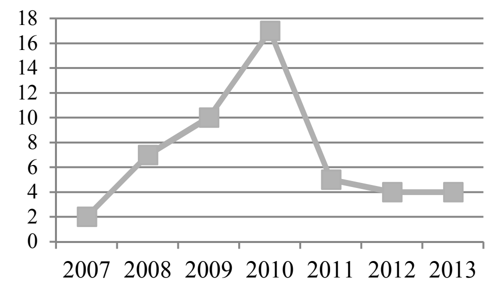

Fig. 3. Software Design

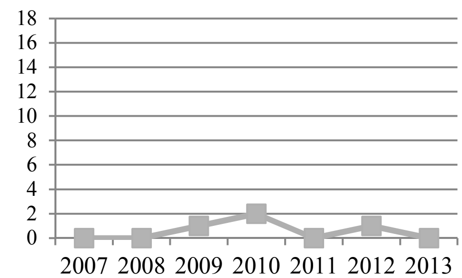

Fig.4. Program Editors

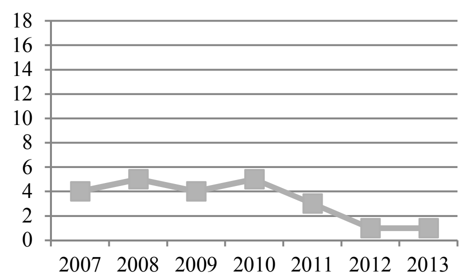

Fig.5. Compilers and Code Generators

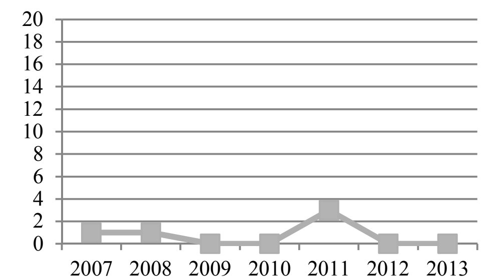

Fig. 6. Interpreters

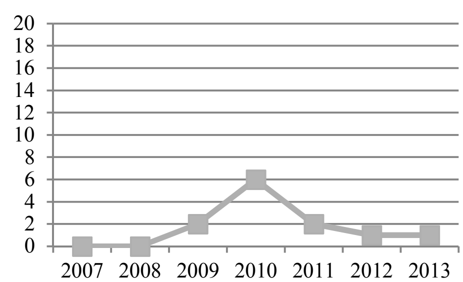

Fig.7. Debuggers

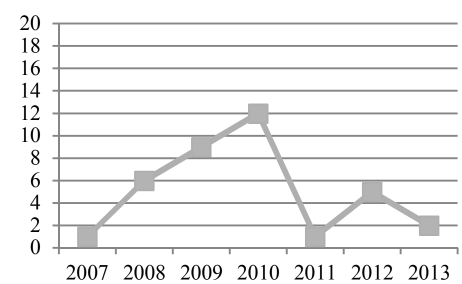

Fig.8. Test Genertors

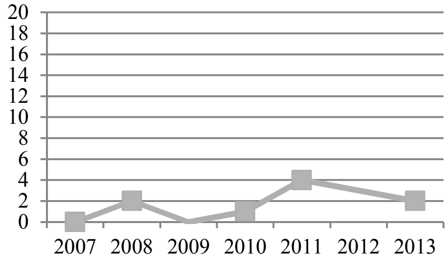

Fig.9. Test Execution Frameworks

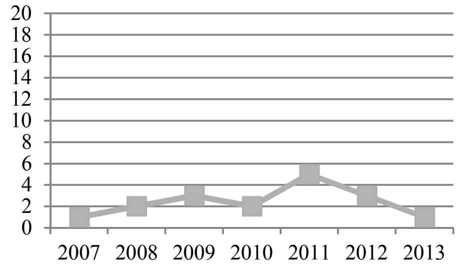

Fig.10 .Test Evaluation

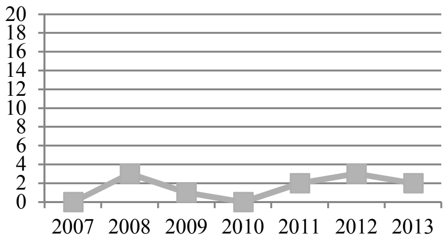

Fig.11. Test Management

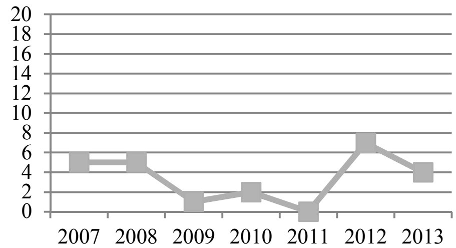

Fig.12. Performance Analysis

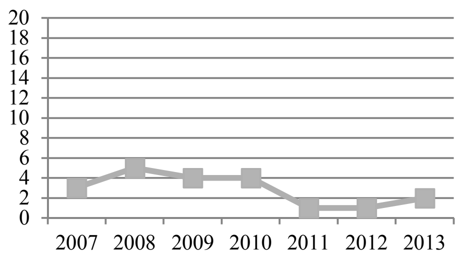

Fig.13. Comprehension

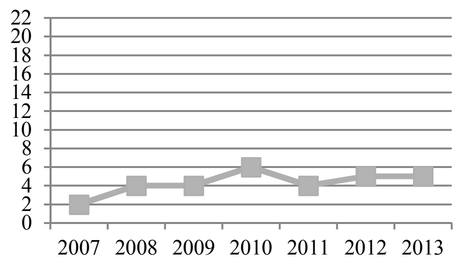

Fig.14. Reengineering

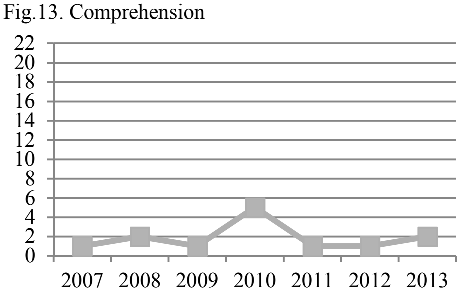

Fig. 15 Defect, Enhancement, Issue and Problem Tracking

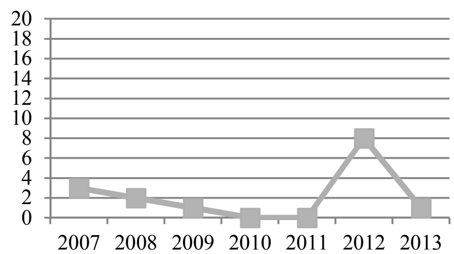

Fig.16 Version Management

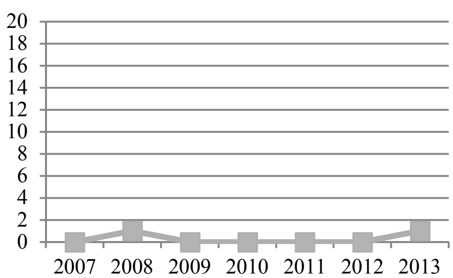

Fig.17. Release and Build

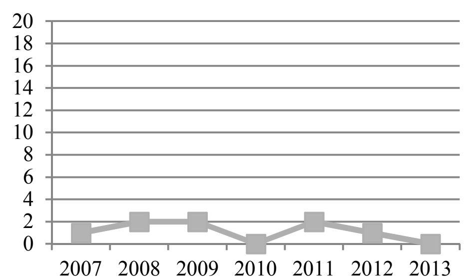

Fig.18. Project Planning and Tracking

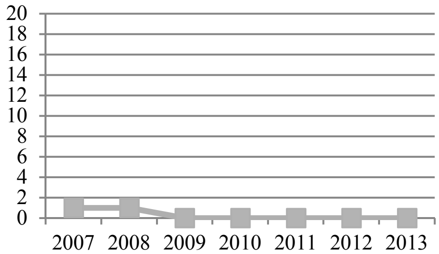

Fig.19. Risk Management

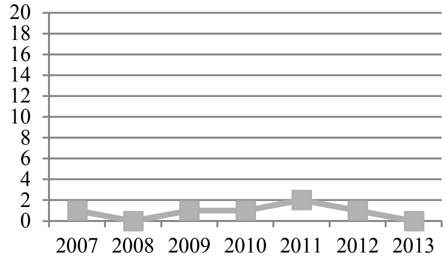

Fig.20. Measurement

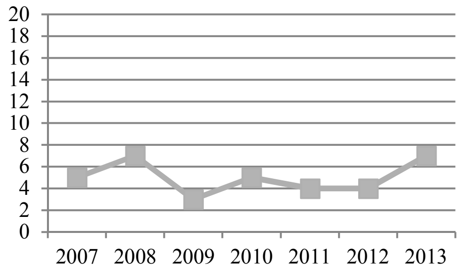

Fig.21. Process Modeling

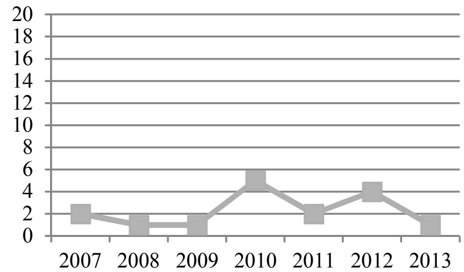

Fig.22. Process Management

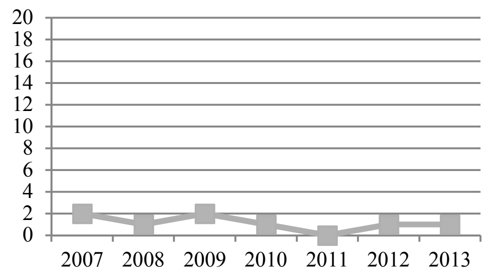

Fig.23. Integrated CASE Environment

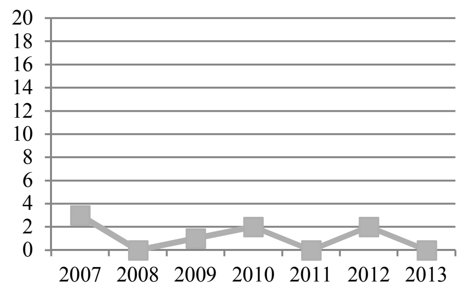

Fig.24. Process Centered SW Engg.

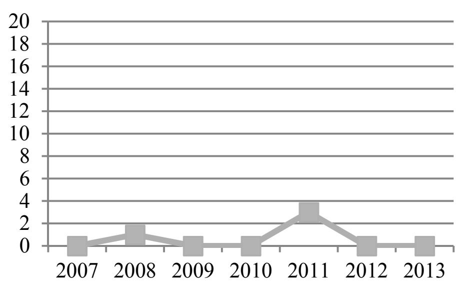

Fig.25. Review and Audit

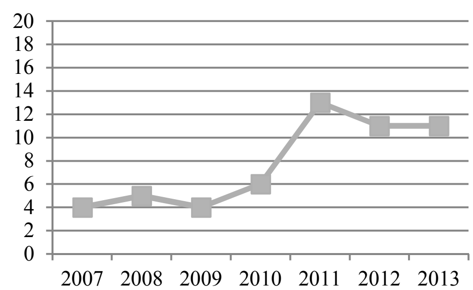

Fig.26. Static Analysis

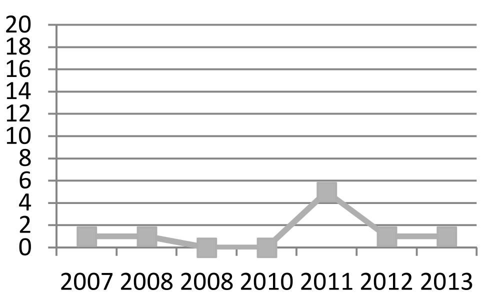

Fig. 27. Tool Integration

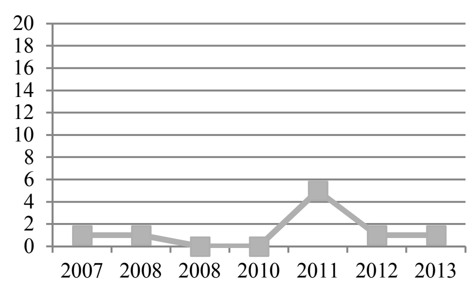

Fig.28. Meta Tools

The Figs. 1, 2, 6, 7, 9, 10, 11, 12, 13, 14, 15 20, 21, 22, 25 belonging to the SWEBoK areas Requirement Modeling, Requirement Traceability, Interpreters, Debuggers, Test Execution Frameworks, Test Evaluation, Test Management, Performance Analysis, Comprehension, Reengineering, Defect Enhancement Issue and Problem Tracking, Measurement, Process Modeling, Process Management, Process-centered Software Engineering and Review and Audit, respectively, show no distinguishable trend. Peaks are observed in Figs. 3, 8, 15 in the year 2010 that correspond to Software Design, Test Generators, Defect, Enhancement, Issue and Problem Tracking respectively. For these sub-areas of SWEBoK, there is an upward trend in the reported tools for the period 2007-10 and downward trend thereafter.

Table 2. Frequency of tools in various journals (2007-2013)

Journal		2013	2012	2011	2010	2009	2008	2007	Total	%age Relevancy
Springer Innovations in Systems Software Engineering	Papers	**16**	**18**	**28**	**31**	**20**	**31**	**25**	**169**	
	Irrelevant	14	7	22	20	13	17	22	115	
	Relevant	2	11	6	11	7	14	3	54	32%
SpringerAutomated Software Engineering	Papers	**16**	**20**	**10**	**14**	**14**	**15**	**13**	**102**	
	Irrelevant	7	17	6	9	10	13	9	71	
	Relevant	9	3	4	5	4	2	4	31	30%
ACM Transactions on Software Engineering and Methodology	Papers	**35**	**18**	**18**	**13**	**13**	**20**	**25**	**142**	
	Irrelevant	32	6	15	05	11	16	21	106	
	Relevant	3	12	3	8	2	4	4	36	25%
Springer Software Tools for Technology Transfer	Papers	**35**	**37**	**34**	**33**	**32**	**32**	**34**	**237**	
	Irrelevant	29	30	28	27	28	23	28	193	
	Relevant	6	7	6	6	4	9	6	44	19%
Springer Software Quality Journal	Papers	**22**	**20**	**31**	**18**	**18**	**22**	**20**	**151**	
	Irrelevant	19	16	23	16	16	20	17	127	
	Relevant	3	4	8	2	2	2	3	24	16%
Springer Requirements Engineering	Papers	**17**	**16**	**19**	**20**	**14**	**15**	**17**	**118**	
	Irrelevant	12	15	15	18	13	13	17	103	
	Relevant	5	1	4	2	1	2	0	15	13%
Elsevier Information and Software Technology	Papers	**121**	**90**	**94**	**90**	**127**	**84**	**75**	**681**	
	Irrelevant	110	77	83	74	106	71	71	592	
	Relevant	11	13	11	16	21	13	4	89	13%
IEEE Transactions on Software Engineering	Papers	**80**	**54**	**47**	**75**	**52**	**53**	**54**	**525**	
	Irrelevant	71	51	31	53	45	44	47	464	
	Relevant	9	3	4	22	7	9	7	61	12%
Springer **Software and Systems Modeling**	**Papers**	39	37	28	23	27	30	25	209	
	Irrelevant	36	35	22	21	25	25	23	187	
	Relevant	3	2	6	2	2	5	2	22	11%
Elsevier **Journal of Systems and Software**	**Papers**	**218**	**193**	**180**	**205**	**167**	**163**	**150**	**1276**	
	Irrelevant	**212**	**165**	**168**	**190**	**159**	**153**	**133**	**1180**	
	Relevant	6	28	12	15	8	10	17	96	8%
Springer Empirical Software Engineering	Papers	**32**	**23**	**26**	**23**	**24**	**27**	**26**	**181**	
	Irrelevant	29	21	26	23	24	27	26	176	
	Relevant	3	2	0	0	0	0	0	5	3%
IEEE Transactions on Software	Papers	**83**	**94**	**107**	**121**	**141**	**120**	**108**	**774**	
	Irrelevant	82	93	104	119	137	115	105	755	
	Relevant	1	1	3	2	4	5	3	19	2%

Table 3. SWEBOK Area / Journal

SWEBOK Area	ASE	TOSEM	TSE	TS	IST	JSS	ISEE	SQJ	STTT	SoSym	ESE	RE
Software Requirements Tools												
Requirements modeling	13%	14%	5%	0%	6%	3%	2%	0%	0%	0%	0%	67%
Requirement traceability	0%	6%	5%	0%	6%	2%	0%	8%	2%	5%	0%	33%
Software Design Tools												
Design	6%	14%	10%	10%	14%	19%	0%	8%	2%	5%	0%	0%
Software Construction Tools												
Program Editors	6%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Compilers and Code Generators	0%	11%	10%	5%	6%	4%	0%	0%	13%	0%	0%	0%
Interpreters	0%	0%	0%	0%	0%	0%	2%	0%	13%	5%	0%	0%
Debuggers	0%	6%	6%	5%	1%	2%	0%	4%	4%	0%	0%	0%
Software Testing Tools												
Test Generators	0%	6%	15%	5%	14%	8%	2%	4%	8%	0%	0%	0%
Test Execution Frameworks	0%	0%	3%	5%	5%	3%	4%	0%	0%	0%	0%	0%
Test Evaluation	3%	6%	5%	0%	3%	3%	6%	0%	4%	0%	20%	0%
Test Management	3%	0%	0%	0%	2%	2%	4%	13%	4%	0%	0%	0%
Performance Analysis	0%	6%	2%	0%	1%	9%	12%	4%	8%	5%	20%	0%
Software Maintenance Tools												
Comprehension	6%	3%	10%	0%	7%	3%	0%	0%	4%	0%	20%	0%
Reengineering	10%	3%	8%	0%	7%	4%	10%	4%	13%	9%	0%	0%
Software Configuration Management Tools												
Defect, Enhancement, Issue and Problem Tracking	6%	0%	3%	0%	1%	3%	0%	8%	8%	0%	20%	0%
Version Mgmt.	6%	3%	3%	0%	2%	1%	0%	13%	17%	0%	0%	0%
Release and Build	0%	0%	0%	0%	2%	0%	0%	0%	0%	0%	0%	0%
Software Engineering Management Tools												
Project Planning and Tracking	0%	0%	0%	11%	2%	2%	0%	4%	4%	0%	0%	0%
Risk Management	0%	0%	0%	5%	0%	1%	0%	0%	0%	0%	0%	0%
Measurement	0%	0%	0%	0%	2%	2%	0%	4%	0%	5%	0%	0%
Software Engineering Process Tools												
Process Modeling	6%	0%	0%	0%	6%	4%	18%	0%	21%	45%	0%	0%
Process Management	3%	0%	0%	5%	2%	1%	16%	0%	0%	14%	0%	0%
Integrated CASE Environments	3%	0%	0%	0%	2%	3%	2%	0%	4%	0%	0%	0%
Process-centered Software Engineering Environment	3%	0%	0%	16%	0%	0%	6%	4%	0%	0%	0%	0%
Software Quality Tools												
Review and Audit	0%	0%	0%	16%	0%	1%	0%	0%	0%	0%	0%	0%
Static Analysis	19%	19%	11%	16%	6%	6%	14%	0%	50%	5%	0%	0%
Miscellaneous Tools Issues												
Tool Integration Techniques	0%	0%	0%	0%	1%	1%	4%	0%	0%	0%	0%	0%
Meta Tools	3%	0%	2%	0%	0%	3%	0%	13%	0%	5%	0%	0%
Tool Evaluation	0%	0%	2%	0%	3%	7%	6%	8%	0%	0%	20%	0%

Table 2 gives a detailed account of classification of papers in terms of relevancy in terms of our research question.

- It shows that Journals with automation as a part of their intended scope, i.e., Automated Software Engineering, have the second highest percentage of automated tools reported in literature.
- Similarly, Journals with automation as not a part of their intended scope, i.e., Empirical Software Engineering and IEEE Transactions on Software (even though they include software in their scope) have the lowest percentage of automated tool reported in literature. It reconciles with our hypothesis that the journals whose scope does not include automation should exhibit low relevancy of tools. However, it was interesting to note that the Journal with automation not part of its intended scope. i.e., Innovations in Systems Software has the highest percentage of reported tools.

Table 3 shows frequency of tools reported over the period 2007-2013 in areas and subareas of SWEBoK.

- The percentage of relevancy of tools reported in literature is consistent with the scope of the journals. For example, Journal of Requirements Engineering has the highest number of reported tools in Requirements Modeling (67%) and Requirement Traceability (33%). Similarly, Journal of Software Systems Modeling is densely populated by tools in Process Modeling (45%). This endorses selection of current 12 journals from literature.
- However, Software Quality Journal does not report any tool in Review and Audit and Static Analysis. Similarly, this journal also reports a very low percentage of automated Testing Tools. One probable reason is that the focus of the said Journal is on quality attributes rather than automated tools.
- It reveals trend of automated tools through identification of gaps and clusters in areas of SWEBoK. Most investigated areas i.e. clusters include Software Quality, Software Engineering Management and Software Tools & Method in order to conduct SLR's. Similarly, least investigated areas (i.e. gaps) include Software Configuration Management, Software Requirements and Software Maintenance exist to conduct primary studies.
-

3.1 Threats to validity

1) Classification is one of the major sources of threats to the validity. Other researchers may possibly come up with different classification schemes. As mentioned earlier, classification in mapping study is done based on titles, abstracts and keywords; and since many author-claimed case studies are not really case studies. However, we tried to ensure the consistency in classification by using two independent researchers.

2) Another objection to the protocol development may be exclusion of the conference publications reporting an automated tool. It was done on purpose as it undermines the sanctity of the protocol. The classification of the candidate studies was made on the basis of the fact that all journals are ISI-indexed. However, there exist no such criteria for the conference publications. Another concern in regards to construct validity may be the scope or number of publications. To keep number of results manageable we have conducted our mapping study on a journal dedicated to empirical software engineering and eleven most well reputed software engineering journals.

4. Conclusions

This paper analysis trend of automated tools published in 12 journals over the period of 7 years. It shows an upward trend in the reported tools for Software Design, Test Generators, Defect, Enhancement, Issue and Problem Tracking(2007-10) and then downward trend from (2010-13). Besides, it identifies gaps in Software Configuration Management Tools, Software Engineering Management Tools and Miscellaneous Tool Issues for conducting primary studies and clusters in Software Requirement Tools, Software Design Tools, Software Testing Tools, Software Quality Tools, Software Maintenance Tools and Software Engineering Process Tools where SLR's can be conducted.

REFERENCES

[1] Budgen, D., Turner, M., Brereton, P. and Kitchenham, B. Using mapping studies in software engineering. In: Proceedings of PPIG, 2008, 195-204

[2] Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M. Systematic mapping studies in software engineering. In: Proceedings of 12th International Conference on Evaluation and Assessment in Software Engineering. Blekinge Institute of Technology, Italy, 2008, 71-80

[3] (http://www.computer.org/portal/web/SWEBoK/html/ch10#Ref1).

[4] Perry, D. E., Porter, A. A. and Votta, L. G. 2000. Empirical studies of software engineering: a roadmap. In: A. Finkelstein (ed.) The Future of Software Engineering 2000: Twenty-second International Conference on Software Engineering, New York: ACM Press, 345-3555

[5] Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K.,et al. Preliminary guidelines for empirical research in Software Engineering. IEEE Transactions on Software Engineering, 2002, 8(8), 721–734

[6] Janice, S. Ethical issues in empirical studies of software engineering. IEEE Transactions on Software Engineering, 2002, 28(3):1171-1180

[7] Basili, V. R., Selby, R. W. and Hutchens, D. Experimentation in software engineering. IEEE Transactions on Software Engineering, 1986, 12(7):733-743

[8] Zelkowitz, M. V. and Wallace, D. Experimental validation in software engineering. Journal of Information and Software Technology, 1997, 39(11):735-743

[9] Zannier, C., Melnik, G. and Maurer, F. On the success of empirical. In: Proceedings of the 28th ACM International Conference on Software Engineering, Shanghai, China, 2006, 341–350

[10] Sjoeberg, D. I. K., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N. K. and Rekdal, A. C. A survey of controlled experiments in software engineering. IEEE Transactions on Software Engineering, 2005, 31(9):733-753.

[11] Segal, J., Grinyer, A. and Sharp, H. The type of evidence produced by empirical software engineers. In: ACM Proceedings of the 2005 workshop on Realising evidence-based Software Engineering, St. Louis, Missouri: ACM Press, 2005

[12] Höfer A, Tichy W.F., Status of empirical research in software engineering. In: Proceedings of the 2006 International conference on Empirical Software Engineering Issues: Critical Assessment and Future Directions. Berlin, Heidelberg, Springer-Verlag, 2007

[13] Kitchenham, B. A., Dyba, T. and Jorgensen, M. Evidence-based software engineering. In: ICSE Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society Washington, DC, USA, 2004, 273-281

[14] Kitchenham, B., Budgen, D., Bereton, O. The value of mapping studies – A participant observer case study. In: Proceedings of EASE'10: BCS eWic, 2010, 1–9

[15] Kitchenham, B., Budgen, D., Brereton, O. Using mapping studies as the basis for further research – a participant–observer case study. Journal of Information and Software Technology, 2011, 53(6), 638–651

[16] Barbara Kitchenham, Pearl Brereton, and David Budgen, Mapping study completeness and reliability – a case study. Proceedings of EASE 2012 Ciudad Real, 2012

[17] Khan, Khalid, S., terRiet, Gerben., Glanville, Julia., Sowden, Amanda, J. and Kleijnen, Jo. (eds) Undertaking Systematic Review of Research on Effectiveness. CRD's Guidance for those Carrying Out or Commissioning Reviews. CRD Report Number 4 (2nd Edition), NHS Centre for Reviews and Dissemination, University of York, IBSN 1 900640 20 1, 2001

[18] http://www.endnote.com