CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Wageningen University & Research Publications

Industry-academia collaborations in software testing: experience and success stories from Canada and Turkey: Special Issue Industry Academia Collaborations in Software Testing

Garousi, V., Eskandar, M. M., & Herkiloğlu, K.

This is a "Post-Print" accepted manuscript, which has been published in "Software Quality Journal"

This version is distributed under a non-commercial no derivatives Creative Commons (CC-BY-NC-ND) user license, which permits use, distribution, and reproduction in any medium, provided the original work is properly cited and not used for commercial purposes. Further, the restriction applies that if you remix, transform, or build upon the material, you may not distribute the modified material.

Please cite this publication as follows:

Garousi, V., Eskandar, M. M., & Herkiloğlu, K. (2017). Industry-academia collaborations in software testing: experience and success stories from Canada and Turkey: Special Issue Industry Academia Collaborations in Software Testing. Software Quality Journal, 25(4), 1087-1089. DOI: 10.1007/s11219-016-9319-5

You can download the published version at:

https://doi.org/10.1007/s11219-016-9319-5

Industry-academia collaborations in Software Testing: Experience and success stories from Canada and Turkey

Vahid Garousi
Department of Computer Engineering, Hacettepe University, Ankara, Turkey
Information Technology Group, Wageningen University, Wageningen, Netherlands
vahid.garousi@wur.nl

Matt M. Eskandar
MR Control Systems International Inc.
Calgary, Canada
matt.eskandar@mrcsi.com

Kadir Herkiloğlu
Quality, Test and Process Management Directorate
HAVELSAN A.Ş., Ankara, Turkey
kherkiloglu@havelsan.com.tr

Post-print

Abstract. Collaboration between industry and academia supports improvement and innovation in industry and helps to ensure industrial relevance in academic research. However, many researchers and practitioners believe that the level of joint industry-academia collaborations (IAC) in software engineering (SE) is still relatively very low, compared to the amount of activity in each of the two communities. The goal of the empirical study reported in this paper is to characterize a set of collaborative industry-academia R&D projects in the area of software testing conducted by the authors (based in Canada and Turkey) with respect to a set of challenges, patterns and anti-patterns identified by a recent Systematic Literature Review (SLR) study, with the aim of contributing to the body of evidence in the area of IAC, for the benefit of SE researchers and practitioners in conducting successful IAC projects in software testing and in software engineering in general. To address the above goal, a pool of 10 IAC projects (6 completed, 2 failed and 2 ongoing) all in the area of software testing, which the authors have led or have had an active roles in, were selected as objects of study and were analyzed (both quantitatively and qualitatively) w.r.t. the set of selected challenges, patterns and anti-patterns. As outputs, the study presents a set of empirical findings and evidence-based recommendations, e.g.: it has been observed that even if an IAC project may seem perfect from many aspects, but one single major challenge (e.g., disagreement in confidentiality agreements) can lead to its failure. Thus, we recommend that both parties (academics and practitioners) consider all the challenges early on and proactively work together to eliminate the risk of challenges in IAC projects. We furthermore report correlation and inter-relationship of challenges, patterns and anti-patterns with project success measures. This study hopes to encourage and benefit other SE researchers and practitioners in conducting successful IAC projects in software testing and in software engineering in general in the future.

Keywords. Industry-academia collaborations; software engineering; software testing; empirical study; experience report; evidence; challenges; success factors (patterns); anti-patterns

Table of Contents

1 Introduction 3
2 Background and related work 4
2.1 Industry-academia collaborations in SE 4
2.2 A SLR of challenges, patterns and anti-patterns of IAC in SE 5
3 Goal and research methodology 8
3.1 Goal and research questions 8
3.2 Research methodology and synthesis methods 9
3.3 Objects under study: pool of the industry-academia collaborative projects 11
4 Results 14
4.1 A simplified process for industry-academia collaborations 14
4.2 RQ 1: Role of challenges, success factors (patterns) and anti-patterns in the projects 15
4.2.1 RQ 1.1: Challenges facing the collaborations 15
4.2.2 RQ 1.2: Success factors (patterns) utilized/applied in the projects 18
4.2.3 RQ 1.3: Anti-patterns that occurred in the projects 21
4.2.4 RQ 1.4: Additional challenges, patterns and anti-patterns (not presented in the SLR) 23
4.2.5 RQ 1.5: Correlations and inter-relationship of challenges, patterns, anti-patterns and success measures 24
4.3 RQ 2: Country-specific issues and context-specific issues 26
4.3.1 RQ 2.1: Country-specific issues and differences (Canada versus Turkey) 26
4.3.2 RQ 2.2: Issues specific to software testing 26
5 Summary and discussions 27
5.1 Summary, implications and recommendations 27

5.2 Threats to validity 29
5.2.1 Internal validity 29
5.2.2 Construct validity 29
5.2.3 Conclusion validity 29
5.2.4 External validity 30
6 Conclusions and future works 30
Acknowledgements 30
References 30
Appendix-Excerpts from the SLR study 35
Challenges impeding success 35
Best practices (success patterns): what to do to ensure success 36
Anti-patterns: what not to do to ensure success 39

1 INTRODUCTION

The global software industry and the Software Engineering (SE) academia are two large communities. However, unfortunately, the level of joint industry-academia collaborations (IAC) in SE is still very low, compared to the amount of activities in each of the two communities.

It seems that a researcher or a practitioner prefers to stay in her/his 'camp' and to talk to fellows from his/her group only. There are a large number of events and conferences organized by each of the two communities, but we usually see only handful numbers of participants from the "other" community in such events. Since the inception of SE around 1969, there has been very little effort by the two sides to collaborate with each other. Various reasons have been discussed by researchers and practitioners for such a lack of motivation for collaborations, such as each side having different objectives, industrial problems lacking scientific novelty or challenges, and the low applicability and scalability problems of the solutions developed in the academia [1]. For the SE research community to have a meaningful future, there is a critical need to better connect industry and academia.

The issue of IAC in SE has been an important topic since the early years of SE. In an applied field such as SE, industrial impact is of outmost importance. For example, there are projects such as the ACM SIGSOFT Impact project (www.sigsoft.org/impact) which have measured and analyzed the impact of SE research on practice. To stress on the importance of IAC, to discuss success stories and how to "*bridge the gap*", various workshops and panels are regularly organized in international research conferences, such as a panel called "*What Industry wants from research*" in the International Conference on Software Engineering (ICSE) 2011 in which interesting ideas from companies such as Toshiba, Google and IBM were presented. More recently an international workshop on the topic of long-term industrial collaborations on software engineering (called *WISE*) was organized in September 2014 in Sweden which hosted several interesting talks.

In his classic book "*Software Creativity 2.0*" [2], Robert Glass dedicated two chapters to "*theory versus practice*" and "*industry versus academe*" and presented several examples (which he believes are "disturbing") on the mismatch of theory and practice. The authors of this article believe that the classic book of Robert Glass is a must read for anyone who intends to make a difference on the issue of IAC. In a keynote talk entitled "*Useful software engineering research: leading a double-agent life*" in the IEEE International Conference on Software Maintenance (ICSM) in 2011, Lionel Briand, an active researcher who is passionate about and active in conducting high-impact research, mentioned that: "*Though in essence an engineering discipline, software engineering research has always been struggling to demonstrate impact. This is reflected in part by the funding challenges that the discipline faces in many countries, the difficulties we have to attract industrial participants to our conferences, and the scarcity of papers reporting industrial case studies*". An interesting blog called "*It will never work in theory*" (www.neverworkintheory.org) nicely summarizes the status-quo on the issue of the IAC as follows: "*Sadly, most people in industry still don't know what researchers have found out, or even what kinds of questions they could answer. One reason is their belief that software engineering research is so divorced from real-world problems that it has nothing of value to offer them*". The blog further states that: "*Instead of just inventing new tools or processes, describing their application to toy problems in academic journals, and then wondering why practitioners ignored them, a growing number of software development researchers have been looking to real life for both questions and answers*".

The first author and two of his colleagues have recently conducted a Systematic Literature Review (SLR) study [3] on IAC with the goal of systematically deriving and synthesizing the list challenges, patterns (best practices, i.e., what to do to ensure success) and anti-patterns (what not to do) in IAC. The SLR systematically selected and synthesized 33 papers in this area [1, 4-35]. Taking the results of the SLR study [3] as an input, the goal of the current study is to characterize a set of

IAC R&D projects in the area of software testing conducted in Canada and Turkey in which the authors have been involved in recent years, with respect to the challenges, patterns and anti-patterns identified by the SLR study [3], with the aim of contributing to the body of evidence in the area of IAC, for the benefit of SE researchers and practitioners in conducting successful projects in software testing and in software engineering. To address the goal, a pool of 10 IAC projects all in the area of software testing were selected and analyzed as objects of study. By achieving the above goal, this paper makes the following contributions:

- The most comprehensive (as of this writing) IAC-focused empirical study in the area of software testing based on evidence and quantitative assessments of challenges, patterns and anti-patterns as identified by the SLR study [3]
- Quantitative ranking of the challenges, patterns and anti-patterns in a set of 10 representative IAC projects across two continents and two countries, a method which could also be reused in follow-up studies as well
- Correlation and inter-relationship analysis of challenges, patterns and anti-patterns with project success measures
- Evidence-based recommendations to ensure success in IAC projects and prevent problems

The remainder of this article is structured as follows. A review of the related work is presented in Section 2. We describe the study goal, research questions and research methodology in Section 3. Section 4 presents the results. Section 5 summarizes the findings and discusses the lessons learned. Finally, in Section 6, we draw conclusions, and suggest areas for further research.

2 BACKGROUND AND RELATED WORK

2.1 INDUSTRY-ACADEMIA COLLABORATIONS IN SE

The issue of IAC is an important topic in almost every discipline. Searching for the phrase "industry academia collaboration" in the Scopus academic search engine returned 111 results, as of April 2015, e.g., [36-44], from various domains such as medicine, social sciences, biochemistry, biology, management and of course, software engineering. Focusing on the field of SE, the literature on IAC has been quite active, especially in the last few years. A recent SLR study [3], conducted by the first author and two of his colleagues, has identified 33 studies in this area [1, 4-35], whose titles are listed in Table 1 (sorted by paper titles).

We briefly discuss a few of the 33 papers, and the reader is referred to papers cited in [1, 4-35] or a recent SLR study in this area [3] for details. Gorschek et al. presented a useful model for technology transfer in practice [4]. Sandberg et al. proposed [6] the "Agile collaborative research" to be used as effective principles for IAC. Wohlin et al. reported [32] a list of success factors powering IAC.

Table 1- Titles of papers on IAC in SE [1, 4-35] (sorted by year of publication)

Year	Paper title	References
1995	Requirements engineering and industrial uptake	[23]
1999	Strategies for industrial relevance in software engineering education	[29]
	Understanding and improving technology transfer in software engineering	[34]
2002	Requirements engineering and technology transfer: obstacles, incentives and improvement agenda	[24]
2006	A model for technology transfer in practice	[4]
	Software conflict 2.0: the art and science of software engineering	[26]
2007	Research collaboration between academia and industry	[25]
2008	Determining the impact of software engineering research on practice	[10]
	Impact of research on practice in the field of inspections, reviews and walkthroughs: learning from successful industrial uses	[18]
2009	Bridging the research-practice gap in requirements engineering through effective teaching and peer learning	[8]
	Industry academia collaboration: an experience report at a small university	[19]
2011	Agile collaborative research: action principles for industry academia collaboration	[6]
	Useful software engineering research: leading a double-agent life	[35]
2012	Bridging the gap among academics and practitioners in non-functional requirements management some reflections and proposals for the future	[7]
	Embracing the engineering side of software engineering	[1]
	Models for technology research collaboration between industry and academia in South Africa	[21]
	The success factors powering industry academia collaboration	[32]
	It takes two to tango: an experience report on industry-academia collaboration	[20]
2013	Conducting empirical studies in industry: balancing rigor and relevance	[9]
	Empirical software engineering research with industry: top 10 challenges	[12]

Post-Print

	Software engineering research under the lamppost	[27]
	Some researcher considerations when conducting empirical studies in industry	[28]
	Success factors for empirical studies in industry-academia collaboration	[30]
	Empirical studies for innovation dissemination: ten years of experience	[13]
2014	Action research as a model for industry-academia collaboration in the software engineering context	[5]
	Embedding research in the industrial field: a case of a transition to a software product line	[11]
	Enablers and impediments for collaborative research in software testing: an empirical exploration	[14]
	Finding relevant research solutions for practical problems: the SERP taxonomy architecture	[15]
	Foundations for long-term collaborative research	[16]
	Get the cogs in synch- time horizon aspects of industry: academia collaboration	[17]
	Practical experiences in designing and conducting empirical studies in industry-academia collaboration	[22]
	The 4+1 view model of industry–academia collaboration	[31]
	Topic selection in industry experiments	[33]

Figure 1 shows the annual trend of papers on IAC in SE. As we can see, a vast majority of the papers have appeared in the window of the last three years (2012-2014), denoting that the SE industry and academia are recently starting to take this issue more seriously. 10 of the 33 papers were published in 2014. Breakdown of the papers by type of authorship are also shown in Figure 1, denoting the increased attention to this very important issue from both practitioners and researchers.

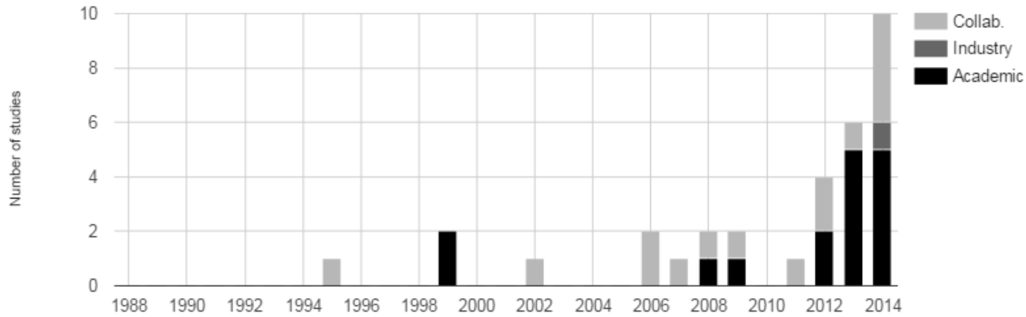


Figure 1- Annual trend of papers on IAC in SE [1, 4-35]. Breakdown of papers by type of authorship are also shown.

2.2 A SLR OF CHALLENGES, PATTERNS AND ANTI-PATTERNS OF IAC IN SE

The first author and two of his colleagues have recently conducted a SLR study [3] on IAC. Among a set of other goals, the main goal of the SLR was to derive and synthesize the list challenges, patterns (best practices, i.e., what to do to ensure success) and anti-patterns (what not to do to ensure success) in IAC. For details of how the SLR was conducted (e.g., the search strings and method, inclusion/exclusion criteria, and assessing quality in the selection of primary studies), the reader is referred to that study [3] and, for the sake of brevity, we do not intend to repeat those aspects in this paper, but only discuss some excerpts.

The SLR systematically selected and reviewed the 33 related papers as listed in Table 1. Through a systematic qualitative analysis process using open "coding" and grounded theory, the SLR derived a list of 64 challenges, 128 patterns and 36 anti-patterns. The SLR [3] categorized those challenges, patterns and anti-patterns into 11, 16 and 7 categorizes, respectively, as shown in Table 2. To provide examples on the identification process of challenges, patterns and anti-patterns and how qualitative analysis was done to derive the list of challenges, patterns and anti-patterns, Figure 2 shows an example of color-coding of the phrases in the primary studies to ensure explicit traceability. The example in the left hand side of Figure 2 is from the primary study [4].

Table 3 shows an example of open "coding" for the 'challenges' aspect. The study derived, synthesized and presented the "axial" codes as its output. In this example, two phrases, related to the 'challenges' aspect in the primary studies, were mapped to two "open" codes which in turn were then mapped to one single "axial" code (i.e., "Lack of training, experience, and skills" in this case). This particular challenge can be seen in Table 1 as the item #3 under the "Challenges" categories.

Table 2 - Categories of challenges, patterns and anti-patterns as synthesized in the SLR study [3]

Challenges	Best practices (success patterns)	Anti-patterns
1. Lack of research relevance 2. Research method related	1. Knowledge management (communication, terminology, transfer, training and skills) 2. Ensure engagement and manage commitment	1. Self-centric approach

3. Lack of training, experience, and skills	3. Consider and understand industry's needs, challenges, goals and problems	2. Stakeholder commitment and benefits presentation related anti-patterns
4. Lack or drop of interest / commitment	4. Ensure giving explicit industry benefits and solve the right problem	3. Research design related anti-patterns
5. Mismatch between industry and academia	5. Have mutual respect, understanding and appreciation	4. Unstructured decision structures
6. Communication-related issues	6. Be Agile	5. Poor change management
7. Human and organizational factors	7. Work in (as) a team and involving the "right" practitioners	6. Ignoring project organizational product characteristics
8. Management-related issues	8. Consider and manage risks and limitations	7. Ineffective communication
9. Resource-related issues	9. Researcher's on-site presence and access	
10. Contractual, and privacy concerns	10. Follow a proper research/data collection method	
11. Anti-patterns: what not to do to ensure success	11. Manage funding/recruiting/partnerships and contracting privacy	
	12. Understand the context, constraints and language	
	13. Efficient research project management	
	14. Conduct measurement/ assessment	
	15. Test pilot solutions before using them in industry	
	16. Provide tool support for solutions	

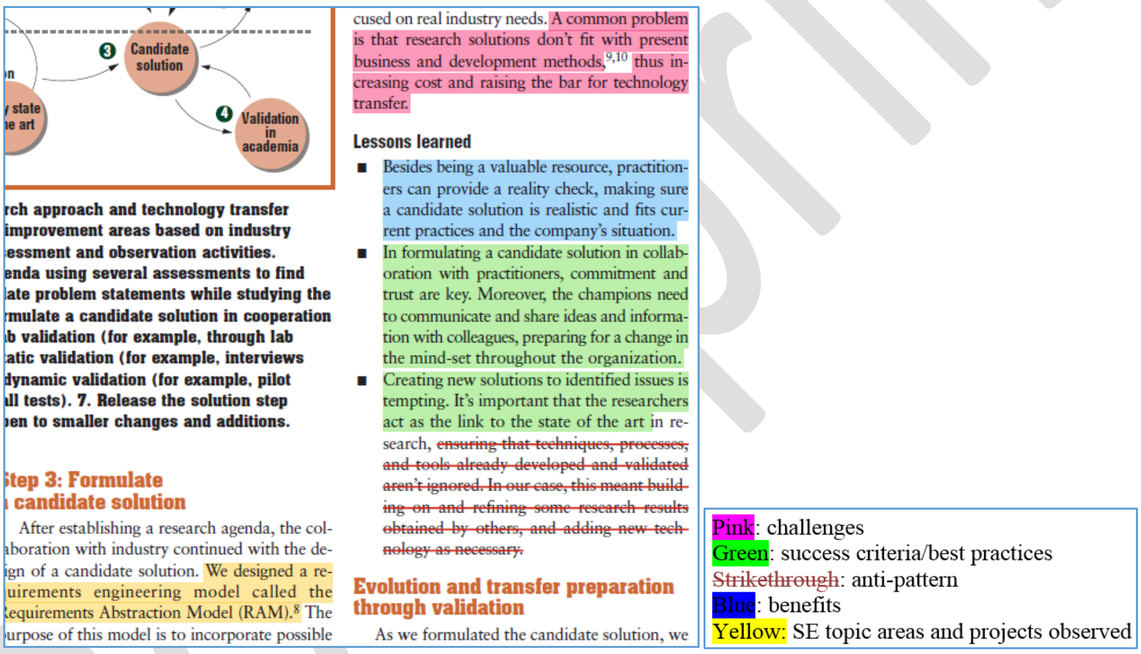


Figure 2- Color-coding of the phrases in the primary studies according to our RQs to ensure explicit traceability. The example in the left hand side is from the primary study [4]

Table 3-An example coding process for the 'challenges' aspect

Examples of phrases in primary studies	Open codes	Axial codes
"*there is a lack of well-trained requirements engineers*" [8]	Lack of well-trained software engineers	Lack of training, experience, and skills
"*Lack of RE (requirements engineering) Education*" from [24]	Lack of SE education	

Each category shown in Table 2 consisted of several items. For example, the challenges in the category of 'lack of research relevance' are shown in Table 4. For example, two important challenges in IAC are: (1) results produced through research are not relevant for practice; and (2) researchers do not properly understand the relevant problems from an industry point of view.

Best practices (success patterns) in the category of 'knowledge management' are shown in Table 5. For example, two useful success patterns in that category are: (1) both sides (academics and industry practitioners) should provide examples of

Preprint

challenges and solutions, and (2) there is a need for continuous learning and training on both sides. Anti-patterns in the category of 'following a self-centric approach' are shown in Table 6. For example, two anti-patterns to avoid are: (1) not to build solutions alone, and (2) having vested interest from either side. For easier discussions throughout this paper, all the items have been coded, e.g., C01-C64 for challenges. To ensure that the current paper is self-contained, all of challenges, patterns and anti-patterns, as synthesized in the SLR study [3], have been adopted and are listed in full in the Appendix of this paper. While almost all items in Table 4, Table 5, Table 6 and the Appendix are self-explanatory, the reader can refer to the full-text of the SLR study [3] for full explanations.

Table 4 - Challenges in the category of 'lack of research relevance'

ID	Challenge description
C01	Results produced through research are not relevant for practice
C02	Researchers do not understand the relevant problems from an industry point of view
C03	Results produced by research are not measurable and exploitable (mechanisms for exploiting them are missing)
C04	University education not focused on industrial relevance
C05	Research topic selection not driven by relevance

Table 5- Best practices (patterns) in the category of 'knowledge management (communication, terminology, transfer, training and skills)'

ID	Pattern description
BP1	Provide examples of challenges and solutions
BP2	Need for continuous learning and for training on both sides
BP3	Improvements to university and research communities
BP4	Researchers should tune their social skills
BP5	Establish common and simple terminology (vocabulary)
BP6	Researchers should better open up knowledge to practitioners
BP7	Run workshops and seminars
BP8	Use existing works (than just inventing yet other approaches)
BP9	Need for prior expertise
BP10	Effective communication
BP11	Create user documentation
BP12	Establish a steering group
BP13	Effective proprietary data management
BP14	Promote the solution and its ease of use using evidence

Table 6- Anti-patterns in the category of 'following a self-centric approach'

ID	Anti-pattern description
AP1	Building solutions alone
AP2	Vested Interest
AP3	Using scientific publications / evidence / taxonomies as the basis for communication
AP4	Assuming one-way knowledge transfer
AP5	Skewing the scientific results
AP6	Too much emphasis on industrial-need
AP7	Assuming that research projects can only be initiated by the academic side
AP8	Assuming that research is free

Note that each of the 33 related papers (listed in Table 1) are 'primary studies', i.e., papers reporting lessons learned from or experience in one or more IAC project(s) by a team of authors. Other than our recent SLR [3] in this topic, we have not seen any other secondary (or meta-) study in IACs in SE.

Looking outside the field of SE, we found two secondary studies [45, 46] about IAC. The study reported in [45] is a review of the literature on university–industry relations with respect to academic engagement and commercialization, which has been authored by a team of 13 researchers from across Europe. The study presents a SLR of research on academic scientists' involvement in collaborative research, contract research, consulting and informal relationships for university–industry knowledge transfer, which the authors refer to as 'academic engagement'. The authors identify the individual, organizational and institutional antecedents and consequences of academic engagement, and then compare these findings with the antecedents and consequences of commercialization. The study states that, apart from being more widely practiced, the academic engagement is distinct from commercialization in that it is closely aligned with traditional academic research activities, and pursued by academics to access resources supporting their research agendas.

Post-print

The study reported in [46] is another more recent (published in 2015) SLR on IACs in general (neutral of a specific discipline). The review resulted in identifying the following five key aspects, which underpin the theory of IAC: necessity, reciprocity, efficiency, stability and legitimacy.

No secondary (or meta-) study has synthesized the challenges, patterns and anti-patterns in IACs as we did in our recent SLR [3]. Our current work uses those items from the SLR study and thus, as a result, we are not able to compare the qualitative or quantitative findings of this study to other secondary studies w.r.t. challenges, patterns and anti-patterns. Almost all the previous studies (the 'primary studies' listed in Table 1) have talked about challenges, patterns and anti-patterns without measuring their intensities (either qualitatively or quantitatively). Also, each of those studies has only discussed only a few challenges, patterns and anti-patterns, while our current work assesses a list of projects w.r.t. the entire set of 64 challenges, 128 patterns and 36 anti-patterns, as produced in the recent SLR [3], and thus this work is the most comprehensive study from that perspective. Comparisons of observed challenges, patterns and anti-patterns in this work with other primary studies can be done in a case-by-case basis, but due to space limits, we do not conduct that in this work, and defer to future work.

3 GOAL AND RESEARCH METHODOLOGY

In the following, an overview of our research method, and then the goal and research questions of our study are presented.

3.1 GOAL AND RESEARCH QUESTIONS

Formulated using the Goal, Question, Metric (GQM) approach [47], the goal of this study is to characterize a set of collaborative industry-academia R&D projects in the area of software testing conducted in Canada and Turkey in which the authors have been involved in recent years, with respect to the challenges, patterns and anti-patterns identified by the SLR study [3], with the aim of contributing to the body of evidence in the area of IAC, for the benefit of SE researchers and practitioners in conducting successful projects in software testing and in software engineering in general in the future. To address the goal, a pool of 10 IAC projects all in the area of software testing, which the authors have led or have had an active role, were selected and analyzed as objects of study (details in Section 3.3).

Based on the above goal, we raise the following research questions (RQs), which have been further divided into sub-RQs:

- **RQ 1:** To what extent did each of the challenges, patterns and anti-patterns synthesized in the SLR study [3] play a role in the projects under study?
    - **RQ 1.1:** To what extent did challenges play in the projects? Which challenges were more observed than the others?
    - **RQ 1.2:** To what extent did patterns (best practices) play a role? Which patterns were more applied than the others?
    - **RQ 1.3:** To what extent did anti-patterns play a role? Which anti-patterns occurred more than the others? Similar to various types of 'retrospective' analyses conducted in conventional project management (e.g., [48-50]), our motivation behind this RQ is to conduct retrospective analysis of our IAC projects to find what could be done better and what could be prevented to prevent obstacles ın projects.
    - **RQ 1.4:** In addition to the challenges, success factors and anti-patterns synthesized in the SLR study [3], which other additional challenges, success factors and anti-patterns were observed/applied in the projects?
    - **RQ 1.5:** What are the correlations and inter-relationship of challenges, patterns and anti-patterns with each other and with success measures in the projects?
- **RQ 2:** Country-specific issues and context-specific issues:
    - **RQ 2.1:** Are there any country-specific issues and differences concerning each of the above items (challenges, success factors, or anti-patterns)? Note that since we only have data from Canada and Turkey, we will only compare the situation in these two countries.
    - **RQ 2.2:** Were there any issues specific to software testing? For example, was a gıven IAC project executed in a certain way since it was about testing, but would have been executed differently if the project was about other areas of software engineering, e.g., requirements engineering?

To address the above questions, a set of quantitative and qualitative metrics are adopted and discussed in the next sub-section.

post-print

3.2 RESEARCH METHODOLOGY AND SYNTHESIS METHODS

To achieve the study's goal and answer the RQs (discussed above), our research methodology is shown as a UML activity diagram in Figure 3. Round edges denote activities and sharp edges represent data/findings. As objects of study, we wanted to select a pool of several IAC projects in Turkey and Canada, the two countries where the authors have been involved in IAC projects.

To ensure covering a variety of challenges, patterns and anti-patterns, we wanted to have both small and large-scale projects and also different industry partner sizes. Also we wanted to have a mix of successful, 'challenged', and failed projects. Also, we wanted to have in the pool both completed and ongoing (running) projects. Since we had a list of 22 software testing IAC projects in which we have been involved and could sample from, given the criteria above, by reviewing the sampling methodologies [51], we chose and used the 'stratified' sampling [51], which is a type of probabilistic (representative) sampling. In this sampling method, the population is first divided into various characteristics of importance for the research, which in our case were, by country, industry partner size, project scale, and project success level. Ideally, stratified sampling based on the above four criteria would have resulted in 2^4=16 combinations (objects of study), but since partner size and project scale were quite related in our pool (i.e., we usually had larger project scales for large partner companies), we sampled 10 projects. We also for example did not have some of the combinations in our pool, e.g., small project for a large partner company. Then the population was randomly sampled within each category or 'stratum'. To keep our effort manageable, a pool of 10 IAC projects all in the area of software testing were sampled whose details are discussed in Section 3.3. 6 of the 10 projects were completed successful, 2 failed, and 2 are ongoing. Almost each project was 'challenged' to a certain extent (to be discussed further in Section 4.2).

As Figure 3 shows, in our research methodology, we also gathered inputs from our personal experience and qualitative evidence in the projects, the papers which have resulted from those projects (e.g., [52-59]), additional email and personal communication with the project staff at the time of working on this paper, and finally from the recent SLR study [3] about the set of challenges, patterns and anti-patterns that we chose as baselines.

At the center of our analysis is the ranking of each IAC project w.r.t. the challenges, patterns and anti-patterns. We used an online spreadsheet to store and discuss the ranks (a partial snapshot is shown in Figure 4). We used a 5-point Likert scale for ranking challenges, patterns and anti-patterns, as shown in in Table 7, e.g., '1' meant if a challenge was 'somewhat' observed, or a pattern was 'somewhat' applied. A challenge or a pattern could be 'not applicable' (N/A) for a given project, e.g., challenge C01 (results produced through research are not relevant for practice) was not applicable for two of the projects since they were cancelled after initial planning (details in Section 3.3).

Each ranking was entered by one author and was then peer reviewed by another author/collaborator, both of whom were actually involved in the given project. To keep the co-authors' team size manageable or based on preference of several industry partners, industry contact points of several projects were not involved directly as authors of this paper, but their inputs were directly fed into the data pool. We ensured that for each assessment, consensus was reached between the first assessor and the peer reviewer. Commenting feature of the online Google Document spreadsheet tool (Figure 4) and Skype tele-conferencing were used to discuss disagreements and reach consensus and to ensure high quality of our quantitative assessments. To minimize subjectivity of judgements for each item by the reviewers, the assessment of each and every single item was thoroughly discussed in person (face-to-face meeting) or by phone/Skype to ensure objectivity of assessment of rankings. Essentially, through meetings, we ensured that each project was ranked jointly by both researchers and industrialists. To increase transparency and replication potential, we would have liked to make the raw data available for other researchers to review and replicate, however unfortunately, some of the rankings are considered sensitive and thus cannot be disclosed.

For the synthesis of data and analyzing trends, we combined the thematic synthesis [64] and the 'case survey' approach [65]. We used the case survey method to collect data about facts of each case study. The data were then analyzed theme by theme using thematic synthesis.

Post-print

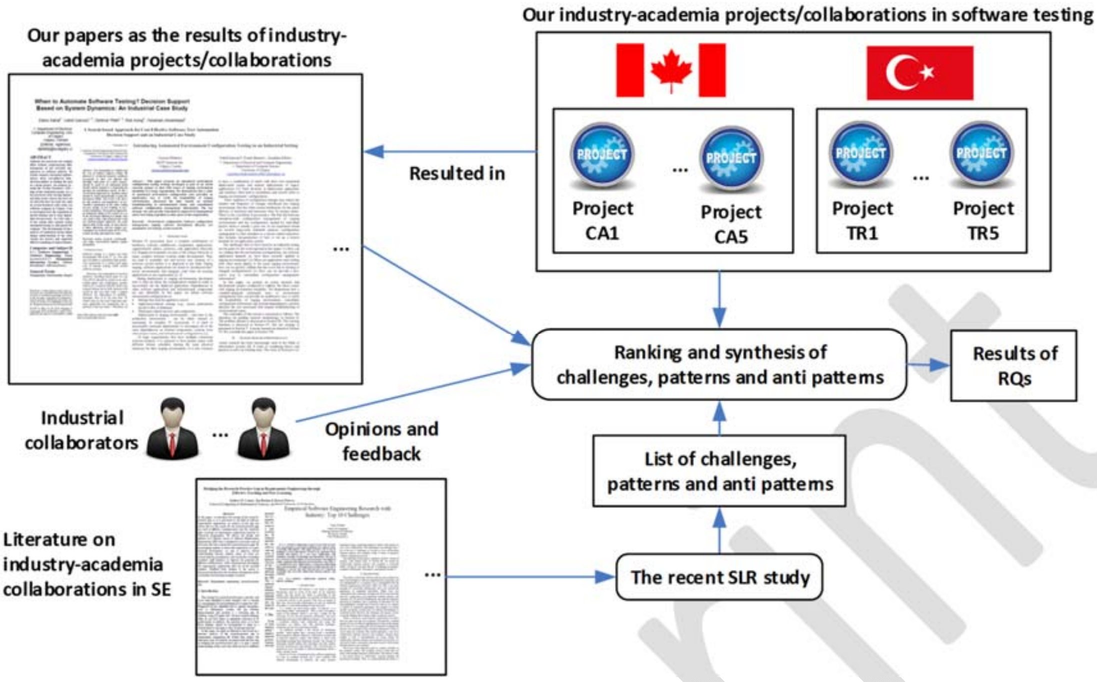


Figure 3-An overview of our research methodology

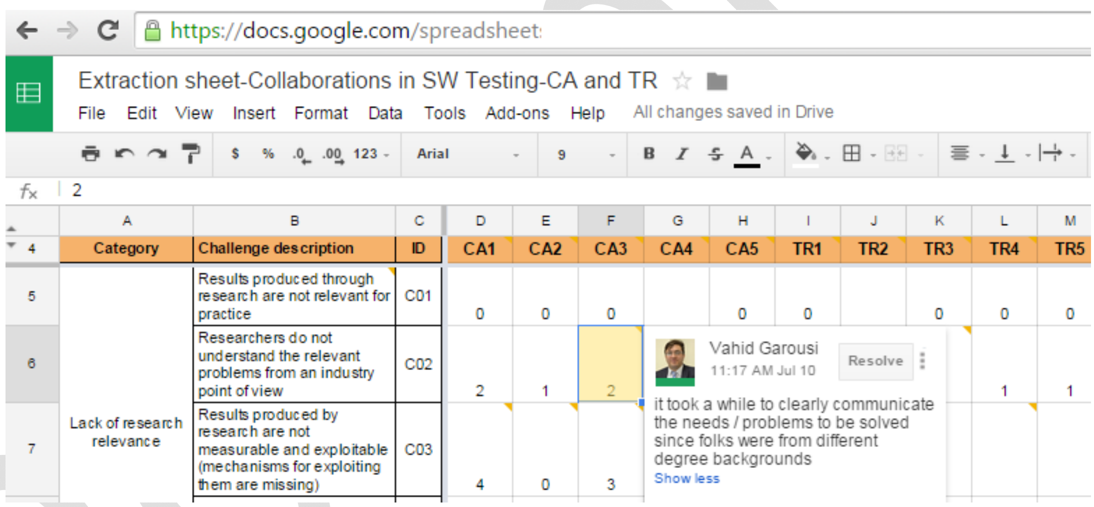


Figure 4-An overview of the spreadsheet used to rank each of the IAC project w.r.t. challenges, patterns and anti-patterns

Table 7- Five-point Likert scale used in ranking challenges, patterns and anti-patterns

Empty	0	1	2	3	4
N/A	Not observed/ applied	Somewhat observed/ applied	Observed/ applied to a medium level	Highly observed/ applied	Very highly observed/ applied

The analysis methods that we intend to apply to answer each of the RQs are as follows. To answer RQ 1 (role of challenges, patterns and anti-patterns), after ranking each project w.r.t. those each challenge, pattern and anti-pattern, we calculated mean values and standard deviations of ordinal Likert scale data, and used these statistics to rank the items and identify the most highly observed challenges, the most applied patterns and the most occurred anti-patterns. We also discussed among the team members to derive qualitative reasons and justifications for the selected challenges, patterns and anti-patterns.

To answer RQ 2 (country-specific issues and context-specific issues), we partitioned the data for project by country first and then derived the country-specific issues, challenges, patterns and anti-patterns. To identify context-specific issues (those specific to projects on software testing), we separated projects on software testing from the rest of the projects in the pool of objects under study and compared the trends of challenges, patterns and anti-patterns in the two partitions.

3.3 OBJECTS UNDER STUDY: POOL OF THE INDUSTRY-ACADEMIA COLLABORATIVE PROJECTS

As discussed in Section 3.2, as objects of study, we selected a pool of 10 IAC projects, as listed in Table 8. These projects are from two different countries and covering a total time-span of about 6 years (2009-2015). For each project, the need (the challenge to be solved) which initiated each project in the first place has been stated. Using proven ccollaboration models such as action research [66], the needs and success metrics were clearly identified early in the projects to ensure success. The solution developed by the research team and released to the industrial context, along with the positive impacts to the partners, and further readings (resulting papers) are also listed in Table 1. We have also added industry partners' domain and size (in terms of number of employees) to Table 8. In each project, only one industry partner was involved. For the company size column in this table, small (S) corresponds to 1-100 employees. Medium (M) corresponds to 101-500 employees, and large (L) corresponds to 500+ employees. As we can see, there is a mix of different partner domains and company sizes in the list. Also, the funding source of each project is shown, in which 'GRA' denotes a Governmental Research Agency and 'C' denotes Company. For confidentiality reasons, further identifying information cannot be disclosed.

Another important project dimension that we would like to note is the collaboration mode (style), i.e., how the research or IAC was performed. One of the best models in this context that we would like to mention is the one proposed by Wohlin [27], which we have adopted and show in Figure 5. There are five levels in this model [27]: (1): not in touch, (2): hearsay, (3): sales pitch, (4): offline, and (5): one team. Except for the case of project CA1, in which the collaboration mode was level 2 ('hearsay'), all the other projects were in level 5 ('one team') since the problems they were trying to address was fully rooted in the 'real' industry problems.

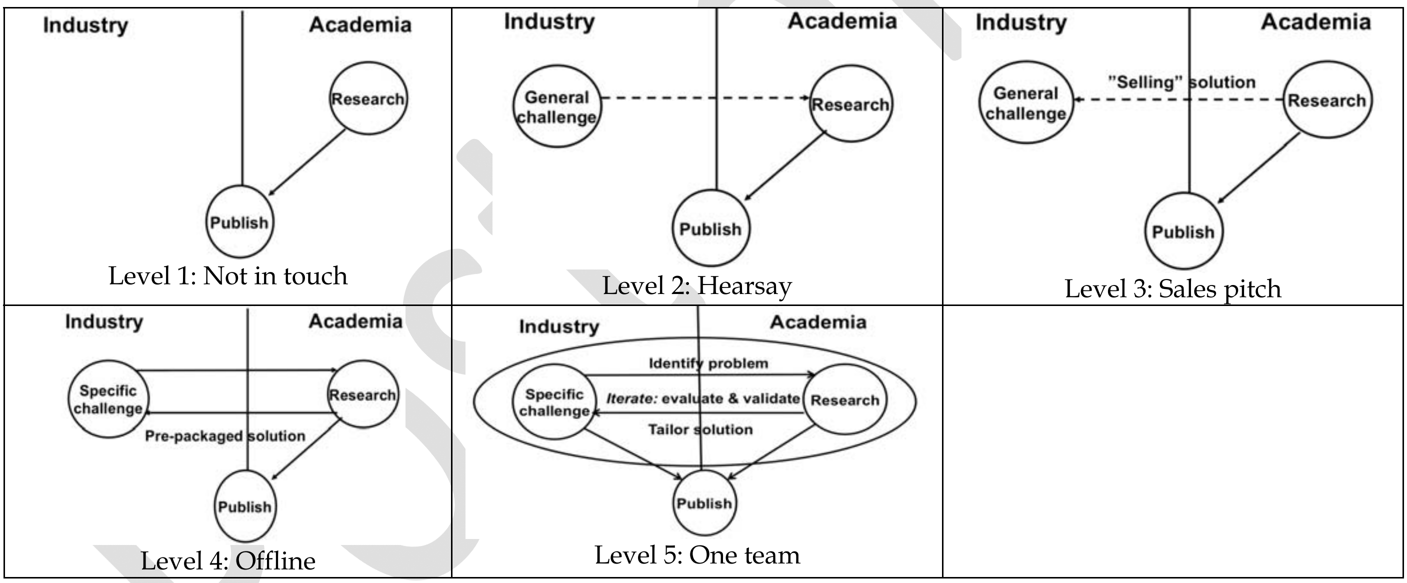


Figure 5-The five levels of collaboration model (closeness) between academia and industry, proposed by Wohlin [27]

Of special importance are projects labelled as CA4 and TR2 which did not get a chance to be executed really. For the case of project CA4, the project was cancelled in early stages after planning, even after getting research funding from a government funding agency in Canada (i.e., NSERC). The main cancelation reason was purely non-technical and it was due to inability to get security clearance (a process internal to the company) for two graduate students who were assigned to be involved in the project. For the case of project TR2, there were a number of initial fruitful meetings toward selecting the concrete research topic to work on [33] and the planning stage was proceeding reasonably. However, in the midst of planning, the project suddenly lost its support from the company management and thus the project was cut short 'stillborn' (details to be discussed in Section 4).

Industry partner sizes and project scales (and duration) were also quite in different ranges, from small to large, from six month to multi-year projects. Thus, to ensure that our study covered a wide spectrum of projects, we see that the projects in the pool have a good level of ‘diversity’ in terms of location (country), industry partner size, project scale, and project success levels.

Post-print

Project ID	Duration	Initiator	Industry partner's...		Funding source	Need (challenge)	Solution	Impact	Resulting papers
			Domain	Size					
CA1	2009-2011	Mostly by the academia (but both side had past collaborations)	Embedded software	L	GRA+C	Need for more effective tool-support traceability analysis in embedded software	A traceability analysis tool-set was developed and released to the industrial context.	Based on results from improving case studies, the traceability analysis tool-set was found useful.	[52]
CA2	2009-2010	Jointly by both sides	Governmental enterprise system	M	C	Manual troubleshooting of environmental configuration issues and staging environment instability was tedious and error prone. For example, there were over 50 hours of service downtime in 3 months due to those issues.	An automated environment configuration testing was developed and released to the industrial context.	The staging environment instability issues were automated detected by the tool and corrected in minutes. The service downtime reduced to 0 - 10 minutes per week.	[53]
CA3	2009-2011	Jointly by both sides	Control systems	S	GRA+C	Cost of manual testing was too high and too many regression faults were observed.	A tool named AutoBBUT for automated test code generation for black-box unit testing was developed and released to the industrial context.	Based on results from improving case studies, with the help of the tool, about 46 hours of testers time was saved in each unit testing iteration.	[54, 55]
CA4	2012-2013	Jointly by both sides	Control systems	L	GRA	Automated software testing of communication frameworks	None since the project was cut stopped in early stages	None. Project got cancelled after planning.	No output papers
CA5	2012-2013	Jointly by both sides	Control systems	L	GRA	Lack of a systematic approach to decide what test cases to automate in software testing	A systematic approach, based on optimization and system dynamics, was developed and released to the industrial context	Based on quantitative measurements, the approach improved the cost effectiveness of software testing activities.	[56-58]
TR1	2013-2014	Industry	Governmental enterprise system	L	C	A major web application had performance issues with high user loads	Systematic software performance testing practices were applied to improve the system	The performance of the web application improved and there was no problems with high user loads	A paper is in preparation
TR2	2014	Jointly by both sides	Defense	M	N/A	The manual GUI testing of a family of safety-critical GIS software systems has proved to be very costly in the last several years	None since the project was cut short in the planning stage	None	No output papers
TR3	2013-2014	Jointly by both sides	Defense	L	C	Testing and maintenance of safety-critical middleware communication protocols had various challenges, i.e., unsynchronized interface artifacts across various SDLC phases, inefficient maintenance and documentation, and change management and integration processes	A model-driven engineering (MDE)-based approached was developed	Many of the challenges were addressed and improving case studies are now underway to quantitatively measure the benefits and highlight the areas for further improvements	[59]
TR4	2015-running	Jointly by both sides	Defense	L	None	Need for test process maturity assessment and improvement (using TMMI and TPI)	Solutions are being developed	Impacts will be measured as solutions are developed and applied	None yet. But several are being planned
TR5	2015-running	Jointly by both sides	Defense	L	None	Need for improvement of test automation practices	Solutions are being developed	Impacts will be measured as solutions are developed and applied	None yet. But several are being planned

Table 8- Pool of the industry-academia collaborative projects analyzed in this study

Post-Print

4 RESULTS

Before presenting the ranking results of challenges, patterns and anti-patterns to answer the study RQs, we first present in Section 4.1 a simplified process model for IAC to better understand the dynamics of these aspects in IAC. Then we move on to answer the study RQs in Sections 4.2 and 4.3.

4.1 A SIMPLIFIED PROCESS FOR INDUSTRY-ACADEMIA COLLABORATIONS

Figure 6 shows a simplified process model for IAC which we have synthesized based on our experience and also the SLR study [3]. Note that this is not a collaboration model (such as those discussed in [4-6]), but only a 'process' for IAC projects. The process can be broken into four phases:

- Inception (formation) phase: approaching and topic selection
- Planning: defining the goal, scope, etc.
- Operational phase: running, controlling and monitoring
- Transition phase: technology/knowledge transfer and impact

Academics and practitioners are involved in each phase. In the inception (formation) phase, the project could be initiated either by academics, practitioners or both (jointly). Joint efforts of both parties are needed to successfully plan and operate the projects and then transition the outcomes to both the industry context and to the research literature. Measures of project success and satisfaction of both parties are determined at the end and the parties may choose to conduct other projects as well afterwards. A set of motivations, challenges, patterns and anti-patterns are 'cross-cutting' to all the phases and would influence/impact the end-to-end lifecycle of the project.

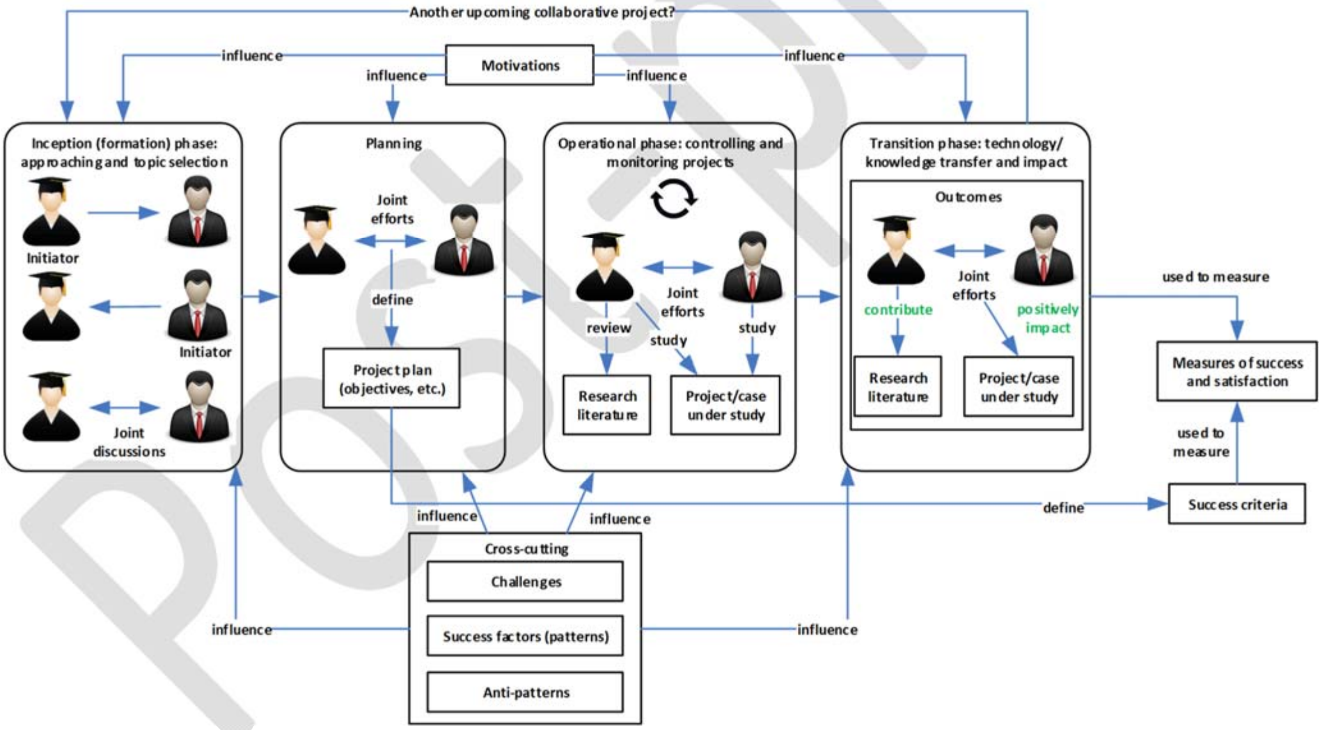


Figure 6-A typical simplified process for industry-academia collaborations

The simplified process model (Figure 6) relates to our projects and study as follows. We showed in Table 8 the initiators and also the impacts of each project. Also, we see in the process model that challenges, patterns and anti-patterns impact and relates to every phase of the process. When reading the rest of this paper and our discussions about challenges, patterns and anti-patterns in IACs, the reader is recommended to refer back to Figure 6, to put things in perspective (e.g., how a given challenge would impact the project inception phase).

Postprint

4.2 RQ 1: ROLE OF CHALLENGES, SUCCESS FACTORS (PATTERNS) AND ANTI-PATTERNS IN THE PROJECTS

Results for RQ 1.1-1.5 are presented next.

4.2.1 RQ 1.1: Challenges facing the collaborations

Figure 7 shows the histograms of the observed level of the set of 64 challenges (see Table 12 in the appendix) in each project. Recall from Section 3.2 that we used a 5-point Likert scale for ranking challenges, patterns and anti-patterns (Table 7), e.g., '1' meant if a challenge was 'somewhat' observed, 4: if a challenge was very highly observed. For example, for project CA4, 13 challenges were N/A, e.g., challenge C01 (results produced through research are not relevant for practice) since the project was cancelled after initial planning and no solution (result) was actually developed. For project CA4, 5 challenges were very highly observed, e.g., issues in intellectual property rights and privacy limited access to data (challenge C61), intangible human factors with organization-wide impact (politics in this case) (C44), and different requirements on novelty were seen by the two sides (C35). The two stopped projects are highlighted with a red box. Throughout the rest of the paper, for analyzing challenges, success factors and anti-patterns, we calculate mean values and standard deviations of ordinal Likert scale data, i.e., use parametric statistics on ordinal data. Such an approach is quite common when interpreting Likert scale data, e.g., [67], to be able to quantitatively assess and compare the trends.

We discuss next the important observations in these histograms. There is a quite a wide variation in the levels of challenges across the projects, the standard deviation (STD) value is 0.42, from a minimum average value of 0.72 (corresponding to the minimum 'somewhat' challenge level in TR4, according to the Likert scale) to a maximum value of 2.17 (corresponding to 'medium' challenge level in TR2), denoting that we experienced a variation of challenges in different contexts. Thus, we can suggest that researchers and practitioners should be prepared to face and deal with any of the 64 challenges listed in the SLR study [3, 68].

TR4 (an in-progress project) has had the lowest average challenge level and 10 N/A challenges, all of which are so since the project has just entered the operational phase as of this writing and several specific challenges have not had the chance to show themselves, e.g., C01 (results produced through research are not relevant for practice). Also, as mentioned in Table 8, both TR4 and TR5 started in 2015, and it should be acknowledged that the IAC 'maturity' of the first author (acting as the principal investigator in all of the projects) has increased during these years after so many joint projects. Thus, even if there is potential for challenges, he is able to partially predict and address them, to best of his ability, before their manifestation. To more systematically assess the challenge levels of the projects versus years and whether there is any influence by an increase in 'maturity' of conducting IAC by the principal investigator on those levels, we depict in Figure 8 the average challenge level of each project versus its year of execution (the midpoint for multi-year projects). For example, for project CA2 which ran during 2009-2010, we assigned 2009.5. To see the trends, a quadratic fit curve has also been added to this chart. We have also conducted regression analysis and shown in Figure 8 the regression p-values and R-squared (R^2) values. As per their statistical meanings, R^2 is the percentage of response variable variation that is explained by its relationship with one or more predictor variables. R^2 is between 0 and 100%, and the higher the R^2, the better the model fits your data. R-squared is also known as the coefficient of determination or multiple determination (in multiple linear regression). For the quadratic fit in Figure 8, R^2 is only 2.9% denoting the very low fit of data on the fit curve. Note that, as it can be seen visually, based on the very few data points, the parameters of the lines are sensitive to the placements of the data points. Thus, we by no means claim that the lines are fully representative of the situation. We can only interpret them as 'rough' trend lines.Figure 8 shows that the increased 'maturity' of the principal investigator in IAC has decreased the challenge levels. For example, we have seen noticeable improvement in the following challenges and many others: communication gaps between researchers and practitioners (C36), difficulty of managing multiple research partners (C37), difficulty to elicit information from developers and software engineers (C38). Project TR2 could be somewhat considered an 'outlier' here since the project suddenly lost its support from the company management due mostly to political reasons (inside the company) and differences of personal styles. By removing the TR2 point in Figure 8, the fit curve will be even more downward, thus further supporting the above observation.

Among the set of completed projects, the project with the lowest challenges level is TR3 (with the value of 0.84, corresponding to the minimum challenge level), denoting that the project outset of TR3 did not provide its participants with lots of challenges. Recall from Table 8 that, in TR3, a model-driven engineering (MDE)-based approached was developed for testing and maintenance of safety-critical middleware communication protocols. In this project, a PhD student working full-time in the company was the efficient 'bridge' between industry and academia, and mainly thanks for his efforts, most of the challenges were addressed as soon as they were faced.

Post-print

The case of CA4 is interesting. Although its average challenge level was only 0.98 (quite low in the range), that project stopped after planning and even getting the funding from a government agency in Canada. As discussed in Section 3.3, the cancelation reason of that project was one single purely non-technical issue, i.e., inability to get security clearance for two graduate students planned to be involved in the project. Thus, we observe that, even if an IAC project does not possess challenges from many aspects, one single major challenge is enough to lead to its halt/failure.

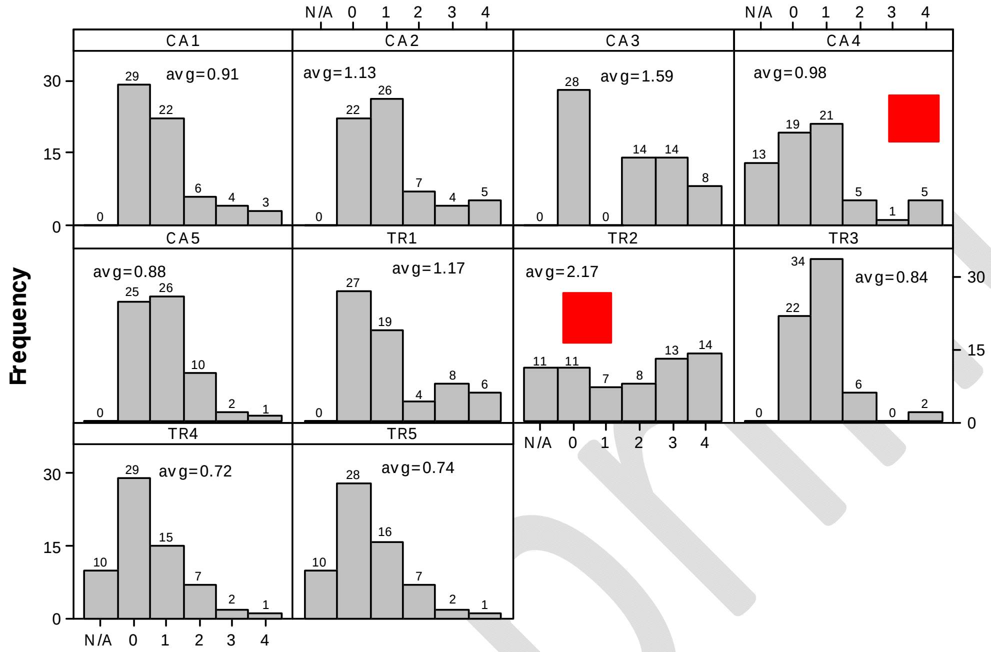


Figure 7- Histograms of the level of challenges in the projects (0: not observed, to 4: very highly observed)

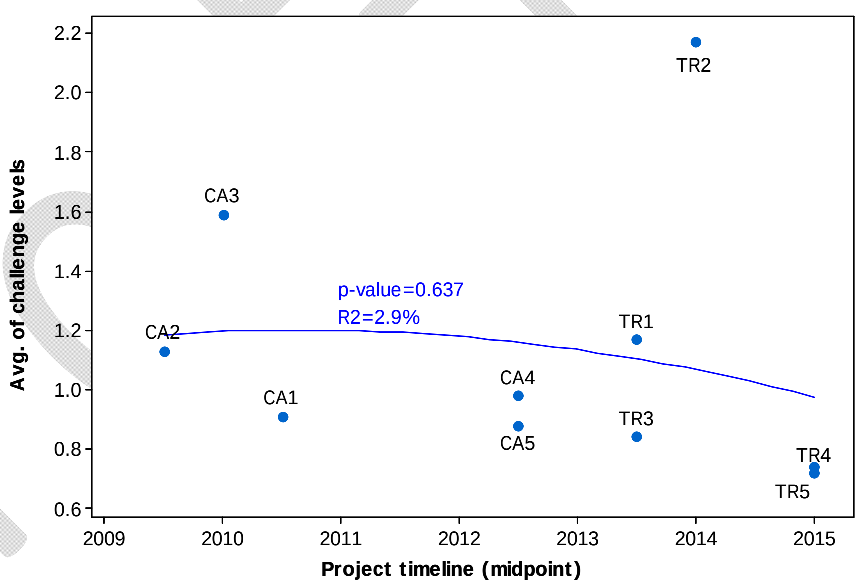


Figure 8- Average challenge level of each project versus its year of execution (the midpoint for multi-year projects)

As the next step in our analysis, Figure 9 shows the average observed level of each individual challenge (C01-C64) across the 10 projects. The shown challenge codes are sorted in descending order by their average value. The top five observed challenges are discussed in the following.

- C04: University education not focused on industrial relevance. The average value is 4, denoting that it was very highly observed in all projects. We observed that, in both Canada and Turkey, university education does not focus on industrial relevance in general. As a result, there is a challenge in IAC projects due to: (1) a misperception of seeing universities as only places for education and not a place for conducting research useful for the industry, and (2) students not taking industry research seriously (at least in the beginning). Still, in informal discussions with companies, we hear practitioners saying things such as: "*I thought university professor only teach!*". This is an issue that the first author and few academics around the world are aware of and have been trying to improve (e.g., [69-73]).

- C35: Different requirements on novelty: Scientists' definition of novelty is mostly about finding new algorithms and new models. However, most practitioners' definition of novelty is about building a new product and improvements in the SE issues and practices used in building those products are not seen as novel. We have seen this different view on novelty especially in dealing with the R&D units of the large partners with which we have been discussing to initiate new joint projects. For some of them, a typical R&D project is usually about building a new software or software-intensive system not improvements in the SE process used for building it. Additionally, we have observed in many occasions that most practitioners do not consider empirical studies worthy of investigation and novel contributions. For example, in many of our projects, we repeatedly observe that researchers would like to develop a novel approach in test-case design phase or to apply existing novel approaches in this area (e.g., those in the area of search-based software testing, e.g., [74-76]). However, we observe most of the time that test-case design is not really the 'problem' practitioners want to be solved. Most practitioners we have seen seem to be OK in terms of the existing test-case design practices that they have in place and they seem to be using white- and black-box test approaches to some extent. Thus, 'pushing' a solution for which there is no need really does not work and creates a challenge of some sort, reminding us of the "Small nail, big hammer" metaphor. What we have seen, again and again is that, what most practitioners look for in terms of topic of IAC projects in testing are better ways to manage testing and execute test cases (usually by automation).
- C30: Different types of knowledge available (industry vs. academia): Researchers are good at research methodologies, SE methods, concepts, algorithms and models. However, practitioners are good at seeing the 'big picture' and what really matters in a large project. Unfortunately, either side is not good at what the other side is good at. This is a challenge and an opportunity (to collaborate) at the same time.
- C28: Different expectations on quality of evidence in research: Researchers usually aim at maximum rigor and evidence in empirical research, while for practitioners, 'good enough' rigor is enough.
- C29: Different focus on scale of solutions: Researchers usually experiment the approaches developed in their research activities on 'toy' examples [65] or small real systems. However, for practitioners, most systems and contexts are usually large-scale.

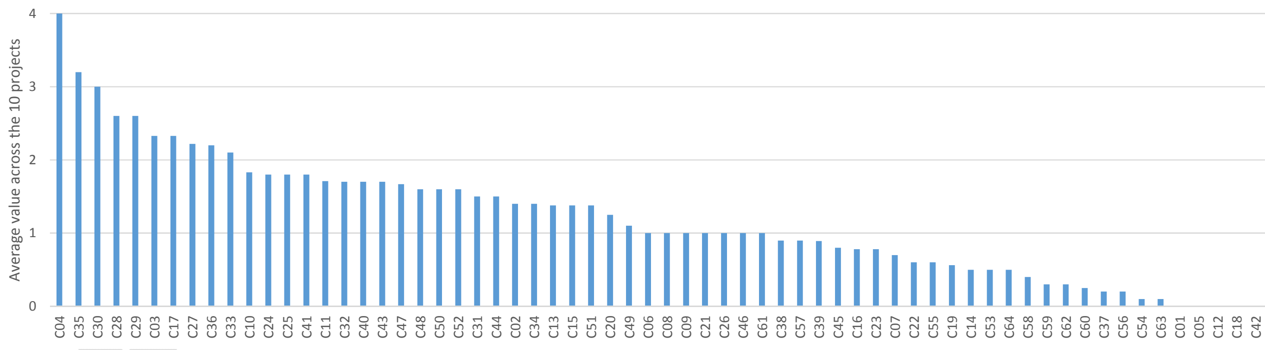


Figure 9- Average observed level of each challenge (C01-C64) across the projects (Y-axis: 0=not observed, to 4=very highly observed)

As a lesson learnt, the authors have found, by experience, that researchers and practitioners should strive for aligning the viewpoints and 'meeting in the middle', e.g., for the following challenges: different requirements on novelty (C35), different types of knowledge available (C30), different focus on scale of solutions (C29).

It would be interesting to compare the observed challenges, patterns and anti-patterns in this work with other primary studies (the 33 papers listed in Table 1). This can only be done in a case-by-case basis since each of those studies has discussed only a few challenges, patterns and anti-patterns, while our current work assesses the projects w.r.t. the *entire* set of 64 challenges, 128 patterns and 36 anti-patterns, as produced in the SLR [3]. For example, Table 9 shows three example challenges and the studies which have mentioned them. Also, as discussed in Section 2.2, almost all these primary studies have not discussed the intensities (either qualitatively or quantitatively) of challenges, patterns and anti-patterns. Thus, only a high-level case-by-case comparison of our findings with the discussions in these studies can be conducted, but due to space shortage, we defer that to future work.

Table 9 – The studies which have mentioned three example challenges (data from the SLR study [3])

ID	Challenge description	# of times mentioned in the existing literature as synthesized by the SLR [3]	Mentioned in the existing literature
C01	Results produced through research are not relevant for practice	8	[4-8, 19, 23, 34]
C02	Researchers do not understand the relevant problems from an industry point of view	4	[1, 8, 28, 35]
C03	Results produced by research are not measurable and exploitable (mechanisms for exploiting them are missing)	1	[23]

4.2.2 RQ 1.2: Success factors (patterns) utilized/applied in the projects

Figure 10 shows the histograms of the usage (application) level of the set of 128 patterns (see Table 13 in the appendix) in each project. We discuss next the important observations in these histograms.

It is interesting to see that all of the average values are quite similar (between 1.95 and 2.99) and the standard deviation for this values (STD=0.29) is not as wide as the case of challenges (STD= 0.42) (Figure 7).

For all the completed projects (CA1, CA2, CA3, CA5, TR1 and TR3), we can see that most of the usage (application) level of patterns are in level 3 (highly applied), while for the two failed projects, this value is much lower, i.e., 2.39 for CA4 and 1.95 for TR2.

For each of the two ongoing projects (TR4 and TR5), as of this writing, about 50 patterns (50/128=39% of all the patterns) are being utilized in level 4 (very highly applied). This is partly due to the fact that, similar to the issue of challenges, the principal investigator (the first author) has gained experience and 'lessons learned' from his past IAC projects. Thus, he is able to apply more patterns in the right time and properly. Similar to what was done above in challenges, to more systematically assess the application level of patterns in the projects versus their timeline and whether there is any influence by increase of IAC 'maturity' of the principal investigator on those levels, we depict in Figure 11 the average application levels of patterns in each project versus its year of execution (the midpoint for multi-year projects). To see the trends, a quadratic fit curve has also been added to this chart. For the quadratic fit in Figure 11, R^2 is 11.4% (slightly better than R^2=2.9% in Figure 8), denoting a slightly better fit of data on the fit curve in this case compared to Figure 8.

As another factor in this context, fortunately, the context of the two projects, including industry partner firm and its staff members involved in the collaboration are welcoming the IAC and are mature in terms of research collaborations. All of these factors have led to major reduction of challenges and also easy applicability of patterns.

Similar to the discussion of challenges, the case of CA4 is also interesting in terms of patterns. As discussed above, although the average challenge level of that project was quite low (only 0.98), that project stopped after planning and even after getting research funding from a government agency in Canada. As discussed in Section 3.3, the cancelation reason of that project was one single purely non-technical issue, i.e., inability to get security clearance for two graduate students to be involved in the project. In terms of patterns, as Figure 10 shows, a lot of patterns were applied to ensure success of this project (during planning), however, one single major challenge (issue) led to its halt/failure. Thus, we observe that, in CA4, albeit investing a lot of efforts and time into applying patterns (e.g., finding an interesting topic and also getting R&D funding from government), one single major challenge (failure in getting security clearance) halted the project and wasted the impact (fruits) of all the patterns applied. This reminds us of the Turkish proverb: "*Damp wood, too, will burn alongside the dry*".

The highest number of N/A patterns are for the cases of the two stopped projects CA4 and TR2 (51 and 53 patterns, respectively). Since these projects did not go beyond the planning phase and did not enter the operational phase, a large number of patterns could not possibly be applied, e.g., those under the category "Follow a systematic research/data collection method" (see Table 13 in the appendix), e.g., BP76 (Use established guidelines and data collection methods), BP77 (Collect different kinds of data collection: quantitative, qualitative, triangulation), and BP78 (Personally interact with the practitioners during data collection).

Post-print

The issue of success patterns in this context somewhat relates to success factors, critical success factors (CSF), and the related body of knowledge in the conventional software project management, e.g. [77-79]. While attempts have been made to offer limited lists of success factors for IAC projects, e.g. in [30, 32], more comprehensive studies in this context are needed. As we realize, IAC projects are quite different in nature compared to conventional software projects and thus the related success factors would be different.

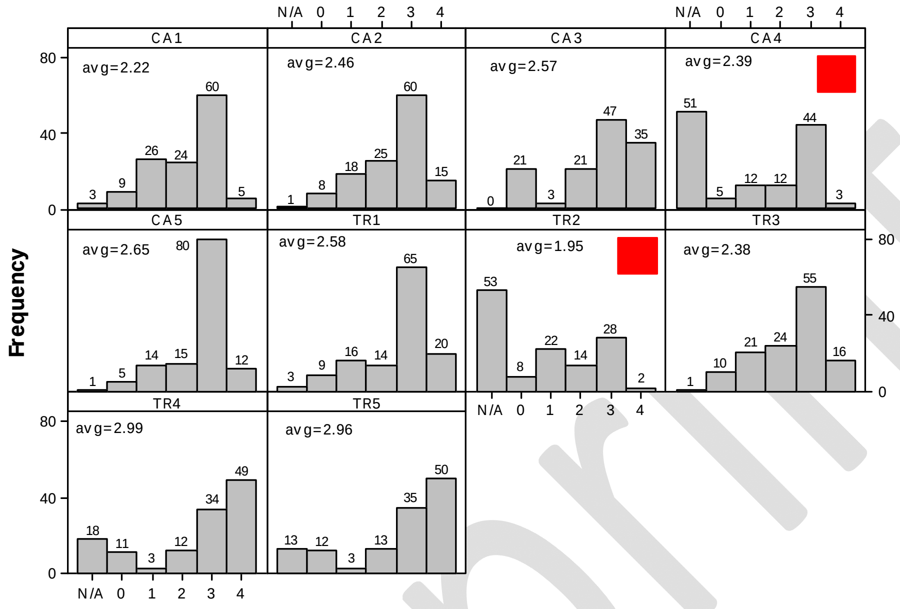


Figure 10- Histograms of the level of patterns that were utilized in the projects (Y-axis: 0=not applied, to 4=very highly applied)

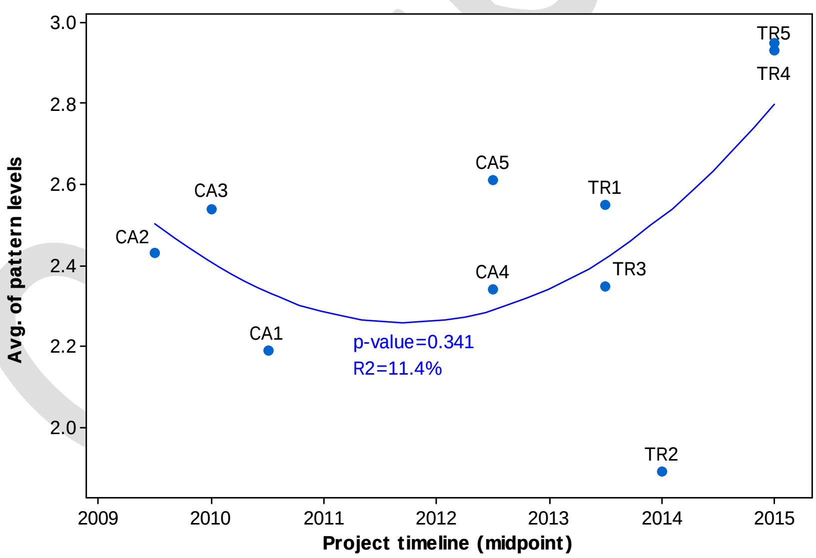


Figure 11- Average application levels of patterns in each project versus its year of execution (the midpoint for multi-year projects)

As the next step in our analysis, Figure 12 shows the average observed level of each individual best practice (patterns) (BP01-BP128) across the 10 projects. The shown pattern codes are sorted in descending order by their average values. To ensure readability of the labels, the chart has been broken into two pieces. The top five patterns applied the most are discussed next:

- BP6: Researchers should better open up knowledge to practitioners: We found out in numerous occasions that the research team (principal investigator and his team of graduate students) should better communicate their knowledge, experience and expertise to practitioners and, more importantly, in 'simple' terminology. According to the results of the SLR study [3, 68], this pattern has also been applied and discussed in several related work too [7, 15, 22, 25].

Preprint

- BP80: Agree on confidentiality before collecting data: To respect sensitive data of the industrial partners, it has been very important to agree on confidentiality term before starting to collect data and that has to be done in the early project inception phase. This pattern has also been applied and discussed in a related work too [12].
- BP1: Provide examples of challenges and solutions: We realized that, especially in early project inception phase, the discussions should start from high-level abstract concepts and soon have to involve concrete concepts, such as software testing challenges in the company, what and where improvements are needed in test processes. In addition, potential high-level solution ideas are better to be discussed to start the buy-in from the industry side (especially the high-level management) which is crucial for the continuity of the project. According to the results of the SLR study [3, 68], this pattern has also been applied in several related work as well [4, 15, 34].
- BP112: Establish a measurement program and define measurable objectives: To assess the 'before' and 'after' the project situations, we strived to establish a measurement program and define measurable objectives in almost all projects. For example, this pattern was applied in project CA2 (with results reported in [53]). The project goal was to introduce automated 'environment configuration testing' for an industrial partner which was a governmental agency who developed contemporary systems to enable timely responses to requests from the oil & gas industry. To objectively assess the approach, which was called Build Verification Testing (BVT), and to provide quantitative insights to its usefulness, we measured a set before- and after-BVT metrics as shown Table 10.

Table 10- Quantitative measures for showing usefulness of a build verification testing (BVT) approach (adopted from [53])

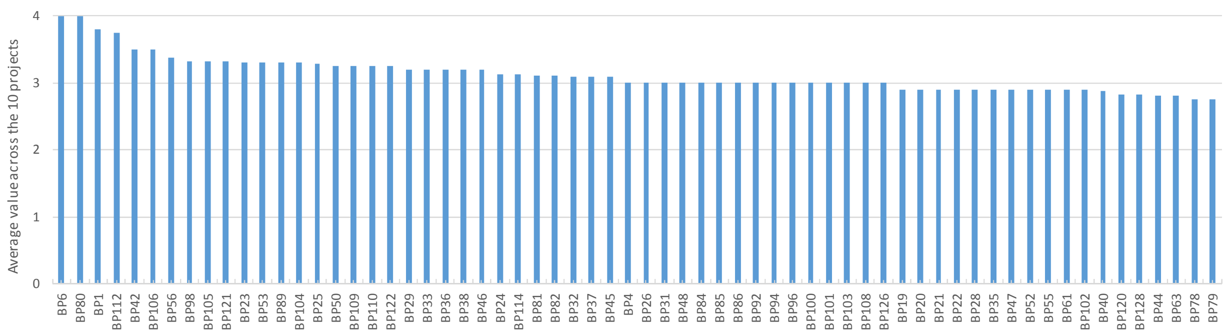
Before BVT	After BVT
• Over 50 hours of downtime in 3 months • Very low confidence • Outage required 2 to 5 people • Primary root cause was very difficult to determine.	• 0 - 10 minutes of downtime per week • 97% of all instability problems were associated with environment configuration • Focused (1 - 2 people) and corrected in minutes • Problem root cause identification is immediately available.

Project CA3 is another example in which this pattern was applied (with results reported in [54, 55]). One of the research questions (RQ) of the study in [55], as a part of that project, was to estimate the cost saving of using the test tool (named AutoBBUT) that we developed in the project. We conducted precise time logging, and according to the measurements, we calculated the time saving of using the test tool as follows:

87 (initial development)+87*2 (test code maintenance)-
120(AutoBBUT's development time)-3 (test code inspect and completion)
=138 hours

The pattern BP112 has also been applied and discussed in several related work as well [4, 16].

- BP42: Use industrial data in research: it is imperative and quite obvious that industrial data, input and parameters should be used in IAC research projects. To our surprise, we have seen IAC projects conducted by our colleagues in which over-simplifications are made and hypothetic data are used instead of industrial data in many parts of the solutions. Such over-simplifications seriously reduce industrial applicability and usefulness of such efforts. This pattern has also been recommended by several related work [8].



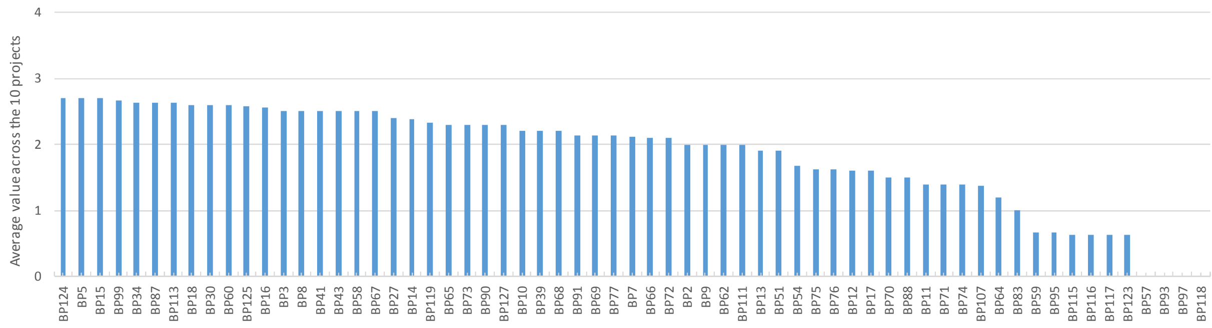


Figure 12- Average utilization level of each pattern (BP01-BP128) across all the projects (Y-axis: 0=not applied, to 4= very highly applied)

4.2.3 RQ 1.3: Anti-patterns that occurred in the projects

Figure 13 shows the histograms of the usage (application) level of the set of 36 anti-patterns (see Table 14 in the appendix) in each project. Next, we discuss the important observations in these histograms.

The highest peak in each case is the level 0 (not conducted). This denotes that all the team members (researchers and practitioners) were 'careful' in not committing any anti-pattern, or reducing their magnitude (scale). However, partly due to lack of experience, out-of-control circumstances, and other reasons, a number of anti-patterns were committed. For example, in project CA1, the research team had to somewhat build the solution semi-alone (level 2 of anti-patterns AP1) since the industry partner was not easily reachable after the project started (due to staff turnovers, etc.). As another example, in project CA3, the industry partner assumed initially that academic research is free (level 3 of anti-pattern AP8) since it had no prior experience in working with university researchers. After a series of discussions, this issue was clarified.

The case of the still-born stopped project TR2 is interesting as it has the highest number of anti-patterns with levels values (chance) larger than one ('observed to some extent'). 11 anti-patterns were 'observed to some extent' in this still-born stopped project, and six anti-patterns occurred to a medium level in it. A few examples are discussed below:

- Insufficient benefits presentation (AP9), level 2 (somewhat observed): After the project was halted, the principal investigator (the first author) and his colleagues conducted a 'retrospective' analysis (e.g., [48-50]) and concluded that researchers did not really sufficiently presented the benefits of the research solutions to be developed in project. Thus the industry partner somewhat doubted whether the project is really worth it to be involved in. This has been a very important 'lesson learnt'.
- Missing management support (AP11), level 2: The top management of the industry partner in project TR2 was quite aggressive in terms of targeting highest value (benefit) without putting too much in (cost). In fact, the other anti-pattern AP8 'academic research seen as free by industry' was the case here too. Thus, when the top management felt that even for getting government R&D funds, some effort and budget has to be spent, it cut its support and the project halted, even if the technical engineers were interested to pursue it. Once again, it clearly resonated to us that 'business priorities' (by top management) drives all projects and that includes IAC projects too.
- Forcing change (AP29), level 2: Although researchers were diligent in not offering a solution without corresponding justifications, it became apparent that the management somewhat perceived the researchers willingness to collaborate as trying to 'force' changes/improvements in the testing practices in the company. Due to the slightly transparent and honest communication style of the principal investigator, 'offering' improvement ideas by the researcher in the initial meetings was perceived by the practitioners as 'forcing change'. This had unfortunately a negative influence which was hard to change once the initial perception was made. Also, we realized that the issues of 'first impression' and 'chemistry' between researchers and practitioners, and mindsets were important in this context. This issue once again highlighted the importance of 'soft' skills for researchers and practitioners in all interactions (including IAC projects). Since that time, the first author has started to read books on this topic, e.g., a book entitled "*How to Influence People*" [80].
- Too much emphasis on the industrial need (AP6), level 1, Observed to a medium level): Although the researchers were aware that not all the emphasis in the meetings should have been placed on the industrial needs and the latest scientific advances should also be discussed in the meetings, things did not always workout quite as the researchers would have liked to. Mainly due to the practitioners' mindset and style, the meetings' direction was inevitably going towards to be focused on their needs only and neglecting the latest scientific advances and their potential usefulness to the needs. In

TR2, the practitioners often were perceiving the scientific advances and methods in the scientific papers as 'too dry' and 'will not work in practice'.

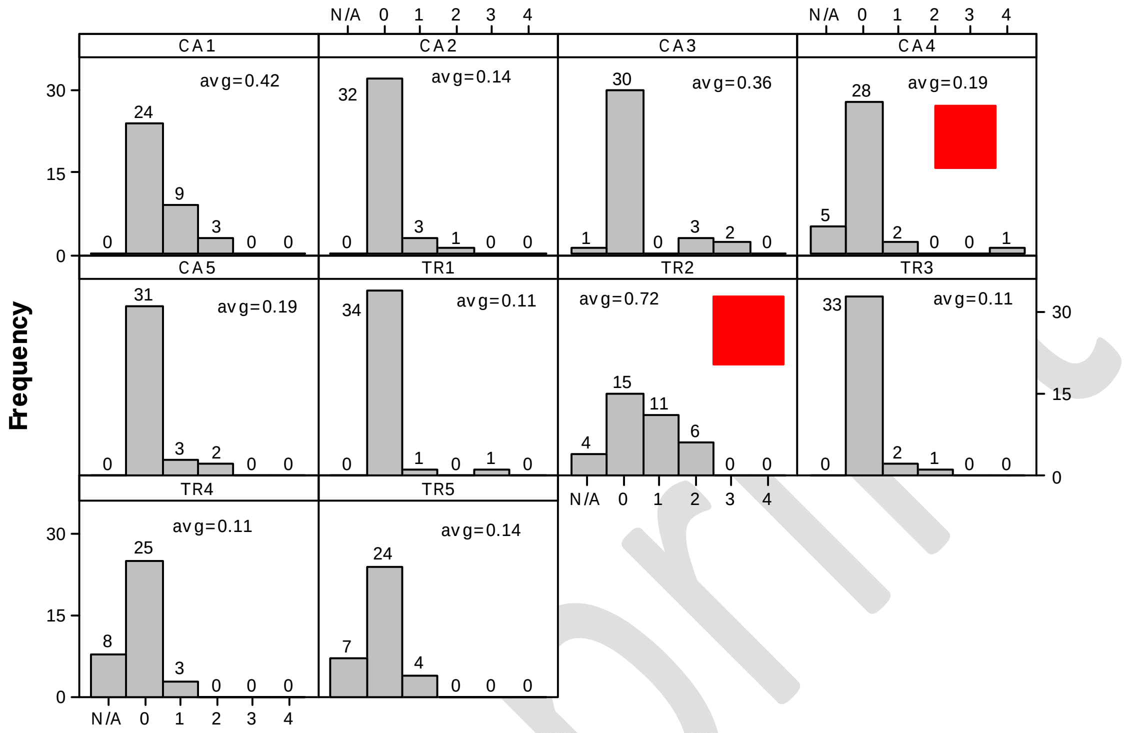


Figure 13- Histograms of the level of anti-patterns that occurred in the projects (X-axis: 0=not conducted, to 4= very highly conducted)

As the next step in our analysis, Figure 14 shows the average observed level of each individual anti-pattern (AP01-AP36) across the 10 projects. The anti-pattern codes are sorted in descending order by their average value. The five most observed anti-patterns are discussed next:

- AP6: Too much emphasis on industrial-need: This issue did occur in project TR2 and was discussed above.
- AP4: Conducting one-way knowledge transfer only: This issue occurred to some minimal extent even in successful projects. It must be realized that knowledge has to be 'exchanged' between industry and academia, and one-way knowledge 'transfer' from academia to industry is not a suitable approach and even can come across as quite demeaning in practitioners' viewpoint (i.e., that they have lack of knowledge and they need to learn from academia). In his 2013 work [12], Wohlin nicely puts it this way: *"It is about doing studies 'with industry'. This is a key issue when it comes to the mindset, which is not only a matter for the researcher, but also for the industrial collaborators who must understand and respect the difference between a researcher and a consultant. Industry must understand and accept that researchers can address areas with industrial challenges; researchers do not solve short-term problems. It must be understood that knowledge has to be exchanged"*. We are constantly trying to convey this message in all of our meetings that the goal of IAC projects is exchange of knowledge between industry and academia and to build solutions together as a team.
- AP9: Insufficient benefits presentation: This issue did occur in project TR2 and was discussed above.
- AP12: Using lab experiments for argumentation before convincing stakeholders: While we conducted lab experiments before trying out the research approaches in industry, e.g., in [52, 54, 81], this anti-pattern occurred in some cases, e.g., lab experiments were simply not possible (due to shortage of hypothetic data in lab settings, e.g., [56, 57]) or was too time consuming.
- AP17: Not distinguishing experimental environment from real-life situations: This anti-pattern slightly overlaps with the one above (AP12). We had to ensure that experimental industry environments (sometimes called 'pilot' projects) in which we were trying our candidate approaches were completely separate from real-life situations (e.g., those exposing to the clients of a given industry partner). In a few cases, such a separation becomes quite challenging to implement and it was not carefully controlled. In those cases, slight problems due to impact of continuous testing (conducted by the research team) on the candidate product releases occurred which were immediately rectified.

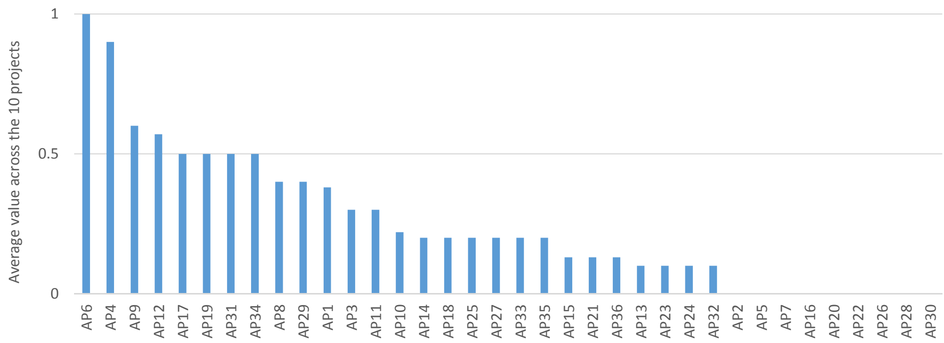


Figure 14- Average level of each individual anti-pattern (AP01-AP36) across the projects (0: not observed, to 4: very highly observed)

4.2.4 RQ 1.4: Additional challenges, patterns and anti-patterns (not presented in the SLR)

RQs 1.1-1.3 explored the extent to which each of the challenges, patterns and anti-patterns synthesized in the SLR study [3] played a role in each of the projects in the pool. We also wanted to know whether any additional challenges, patterns and anti-patterns (not presented in the SLR) were also involved (RQ 1.4). To our surprise, the list of 64, challenges, 128 patterns and 36 anti-patterns, as derived in the SLR study [3] were almost enough to capture all the possible issues in all the pool projects. However, we want to emphasize next a few issues that we observed as playing critical roles. Also, several issues specific to the nature of projects (being software testing) was identified which are discussed in Section 4.3.2.

In support of three existing patterns: BP40 (show benefits of the research solutions for the industrial partner), BP43 (solution should be cost-efficient) and BP108 (Ensure that end research results hit the right trade-offs, e.g., quality and cost), we observed the need for continuous cost-benefit analysis of joint R&D IAC projects, since businesses are very careful about costs (and that includes their staff time, "time is money"). The first author was involved in at least one major IAC project (location and context cannot be disclosed due to confidentiality), which was in another area of software engineering (not testing), in which the industry partner suspended the project in the middle purely due to the fact that they did not see any more the positive balance of cost-benefit in the project.

Related to the issue of need for continuous cost-benefit analysis of IAC projects, in the first chapter of a related book entitled "*Value-based software engineering*" [82], Barry Boehm mentions a very important point: "*Much of current software engineering practice and research is done in a value-neutral setting, in which every requirement, use case, object, test case, and defect is equally important. However, most studies of the critical success factors distinguishing successful from failed software projects find that the primary critical success factors lie in the value domain*". Similar to software projects, considerations of costs and benefits should be taken seriously in SE research and IAC research projects which highlights the need for 'value-based management' of SE research.

Both researchers and practitioners invest time, energy and often research funding (the cost side of the equation) into the collaborative projects. There should be tangible benefits and returns on investment (ROI) from the investments, or the collaboration will fail to succeed. Although researchers and practitioners have often different sets of cost and benefit drivers, it is possible (often not easy though) to somewhat align them or to ensure that the ROI is achieved for both sides. Clearly, the funding scheme of such projects should be also taken into account in the picture. In most cases, including the case of our projects, as long as there is positive impact to the industrial partner and that impact surpasses the costs involved, the industrial partner will be 'happy'. As for the researchers, as long as the researcher gets a number of high-quality publications, solves technically-challenging problems, and receives (reasonable) funding to cover research costs, s/he will be 'happy'. Ultimately, we have observed that, it is essential that researchers focus on solving the challenges and addressing the needs in the most efficient manner possible.

Similarly, we have also seen repeatedly that the GQM+ family of strategies is also invaluable in systematically setting the technical directions of IAC projects on the basis of its alignment with business strategies of a given industrial partner (relating to cost and benefit again). Ultimately, to ensure the success of IAC projects, the team shall define and continuously measure both technical success metrics and also business (cost-benefit) success metrics. In all of our projects, we carefully followed this approach.

4.2.5 RQ 1.5: Correlations and inter-relationship of challenges, patterns, anti-patterns and success measures

After exploring challenges, patterns and anti-patterns 'separately' in RQs 1.1-1.4, we would like to investigate their correlations and inter-relationship in this part.

Figure 15 shows a simplified but realistic cause-effect diagram of inter-relations of challenges, patterns and anti-patterns. Challenges and anti-patterns are expected to negatively impact success, while patterns (their application) positively impacts success. Challenges necessitate the need for application of patterns which then address challenges. Patterns and anti-patterns usually neutralize impacts of each other. Anti-patterns usually bring more challenges.

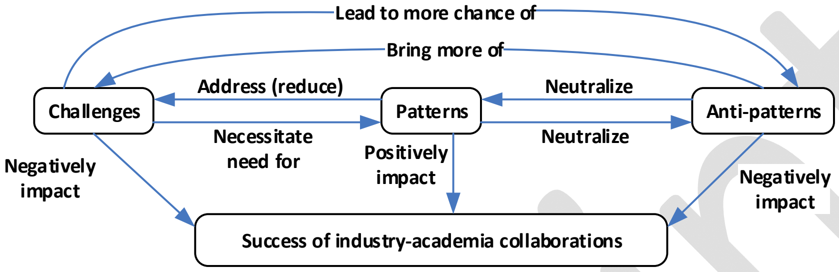


Figure 15- Inter-relation of challenges, patterns and anti-patterns in industry-academia collaborations

To provide further context to our quantitative assessment, we assessed additionally, as a team, another very important factor, the project's success measure. Using the same 5-point Likert scale rubric used for the other three factors (shown in Table 7), the success of each IAC project was measured by a value from 0 (very unsuccessful)-4 (very successful). To decrease subjectivity, opinions of industry and research partners were included and average values were calculated. To ensure following a systematic approach and minimizing subjectivity, principles of the Delphi methodology [83] were used in this assessment in which both industry and academic partners were involved. The criteria we used included the success and satisfaction *during* the project progress (e.g., the interactions) and with the project outcomes. Justifications and reasoning for success measures were discussed by email and voice communications (using Skype) until consensus was reached. The success measure values are shown in Figure 16 (see the Y-axis in scatter-plots in the top-row).

The two failed projects (CA4 and TR2) have been identified by two circles in top-right scatter-plot of Figure 16 and could be easily located using their labels in the other scatter-plots too. As expected, their success measures were reported to be the lowest by the team, 1 and 0.5 (out of 4), respectively. Success measure of 1 out of 4 was assigned to CA4 since it passed both the inception and planning phases and just got halted in start of the operational phase (as discussed in Section 3.3). A success measure of 0.5 was given to TR2 since it did not even pass the planning phase. For the two failed projects, we see how the combinations of challenge, pattern and anti-pattern measurements have led to their fate (failure).

The matrix scatter-plot in Figure 16 shows pair-wise inter-relation of challenges, patterns, anti-patterns and success measures with each another. Linear regression fits have also been added.

Post-print

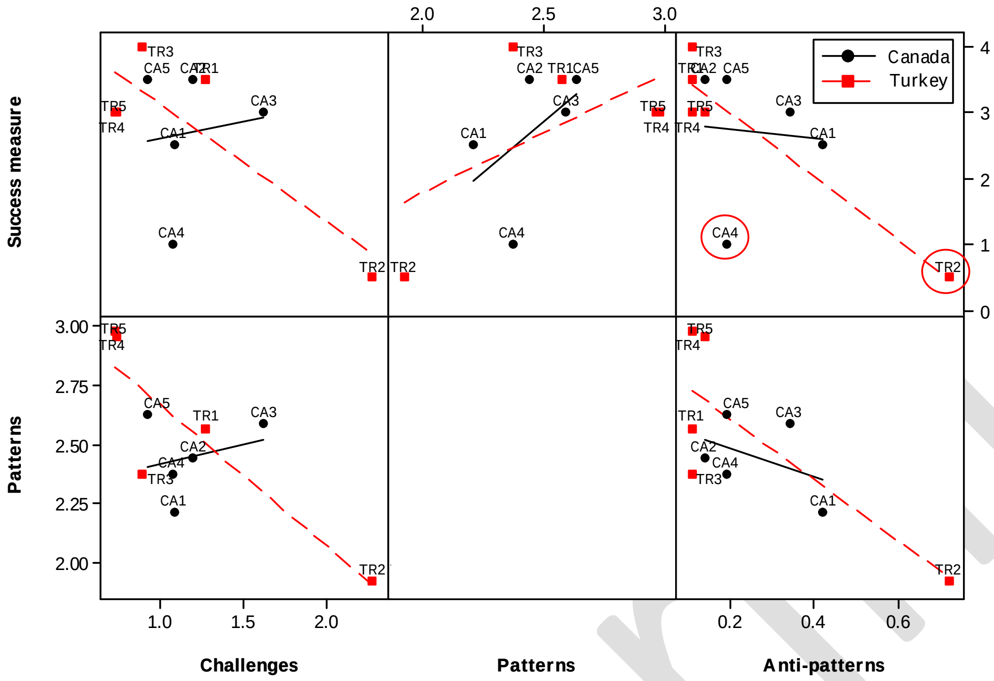
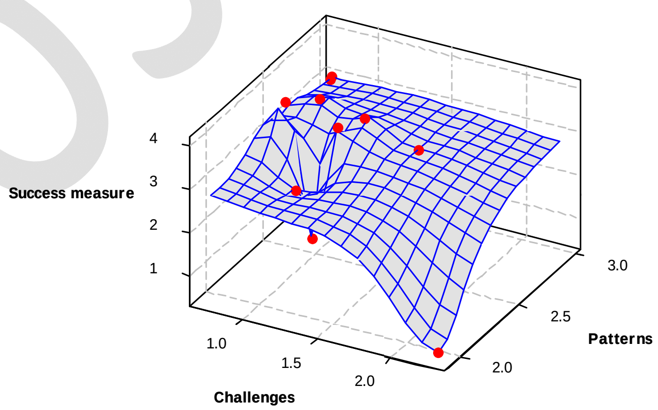


Figure 16- Matrix scatter-plot of challenges, patterns, anti-patterns and success measures

To better understand the inter-relation of tuples of the above four factors, and to augment graphs in Figure 16, Figure 17 visualizes 3D-surface plots of two groups of 3-variable combinations of four items: challenges, patterns, anti-patterns and success measures. Table 11 shows the Pearson correlation and P-values of the data-set of Figure 16. We discuss next the findings from these data and visualizations.

The highest Pearson correlation value (0.865) in Table 11 belongs to the pair of challenges and anti-patterns, denoting that the higher the amount of challenges, the higher possibility of committing anti-patterns, i.e., the higher the chances of a mistake, the higher the probability of the mistake (validating the edge labeled as 'Lead to more chance of' in Figure 15). The second strongest Pearson correlation value (-0.65) is for the pair of anti-patterns and success, meaning that the higher the rate of committing anti-patterns, the less the chance of success, as one would expect. These two empirical measurements and also the other three correlation values in Table 11 nicely confirm what one would expect and show that these four factors are indeed tightly correlated and inter-connected.



Post-Print

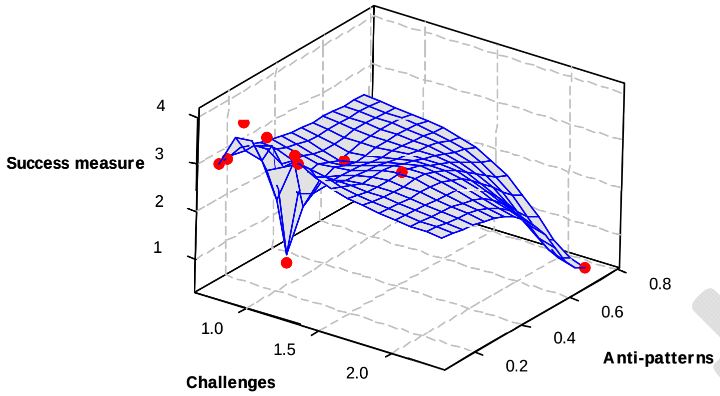


Figure 17- 3D-surface plots of challenges, patterns, anti-patterns and success measures

Table 11- Pearson correlation and P-values of the data-set of Figure 16

	Success measure	Challenges	Patterns
Challenges	Pearson correlation=-0.5 (P-value=0.141)		
Patterns	0.482 (0.158)	-0.19 (0.598)	
Anti-patterns	-0.65 (0.042)	0.865 (0.001)	-0.325 (0.36)

4.3 RQ 2: COUNTRY-SPECIFIC ISSUES AND CONTEXT-SPECIFIC ISSUES

4.3.1 RQ 2.1: Country-specific issues and differences (Canada versus Turkey)

RQ 2.1 aimed at exploring whether there are any country-specific issues and differences concerning each of the above items (challenges, patterns, anti-patterns). To facilitate answering this RQ, the points in the scatter-plots of Figure 16 have been grouped by country: circular black points denote the Canadian projects, while square red points correspond to projects in Turkey. The most noticeable country-dependent differences as seen in Figure 16 are discussed next.

For the case of patterns, the bottom-left scatter-plot shows that the five Canadian projects are somewhat 'closer' to each other, compared to the five Turkish projects. Considering that the impacts of other factors are equal in all cases, this may imply that the project teams in Canada had similar opportunities of applying patterns while, for the case of Turkey, there seems to be quite a wide variety in such opportunities. But let us note that, given the small size of the project pool, this is only an initial observation which needs investigation in large data sets.

By looking at the linear regression fits for the two countries, we can see that in three of the five scatter-plots, both fit lines have similar slopes (either upward or downward). However, for top-left and bottom-left scatter-plots, the slopes of the Canadian and Turkish fit lines differ. For the former (top-left), the Turkish fit line is as expected (more challenge leads to less success). But in the first sight, the Canadian fit line seems somewhat counter-intuitive (more challenge leads to more success!). But let us note that, as discussed above and shown in Figure 15, challenge is not the only determinant of success, e.g., it could be that there were more challenges, but the team perhaps applied much more patterns in those projects.

4.3.2 RQ 2.2: Issues specific to software testing

In the last sub-RQ, our goal was to assess whether the IAC projects, taken as objects of this study, were executed in a certain way since their scope was about software testing, rather than other areas of software engineering, e.g., software requirements engineering. In the ranking of challenges, patterns and anti-patterns, several issues were highlighted in this regard and are summarized next.

One challenge specific to software testing was that some industry partners perceived, in the project inception phase, that the university team will help them in testing their software systems. We even remember quotes such as: "*How long will it take for your team to [manually] test our software and report the bugs?*". After several meetings, the research team was able to communicate 'gradually' that their job is not to offer manual testing services, but to develop better and efficient ways to testing. Another challenge specific to software testing was about test tools and test automation. Some industry partners perceived that the researchers will help them choose a 'good' commercial test tool and help them in using the tool (e.g.,

Postprint

writing the automated test scripts). We remember quotes such as: "*Which test tool can help me test this software faster?*", or "*Have you used the test tool X and do you recommend it?*". Again, in the expense of several meetings, the research team was able to communicate that they are not the experts in choosing and using commercial test tools, but instead, they are here to develop better and efficient ways to test.

An important anti-pattern for researchers that we have observed many times in many projects is not to disclose (e.g., in public presentations or in papers) the test results or number of defects found in the systems during the projects. This can seriously damage the creditability and image of the industry partner in eyes of its customers and immediately put a negative impact on the continuity of the IAC project. We as a team were constantly careful not to make this mistake.

5 SUMMARY AND DISCUSSIONS

Summary of and implications of our findings are discussed in Section 5.1. Potential threats to the validity of our study and the steps that we have taken to minimize or mitigate them are discussed in Section 5.2.

5.1 SUMMARY, IMPLICATIONS AND RECOMMENDATIONS

Our study gathered quantitative and qualitative results from a set of 10 IAC projects conducted between researchers from the Canadian and Turkish universities and practitioners in these countries.

The survey revealed interesting insights in terms of challenges, success factors (patterns) and anti-patterns in IAC projects, with the aim of contributing to the body of evidence in the area of IAC, for the benefit of SE researchers and practitioners in conducting successful projects not only in software testing, but in the SE in general. As we observed in our projects and also discussed in other works cited in this papers, it is not hard to achieve success in IACs and we thus recommend researchers and practitioners in the area to take active steps in that direction and follow the success patterns as we reported in this work and also reported in the previous studies [1, 4-35].

The work reported in this work complements our recent SLR study [3] in that we have assessed each of the 64 challenges, 128 patterns and 36 anti-patterns, discussed in the literature, in the context of actual SE projects and have made recommendations on what to do and what *not* to do to ensure success. It is the authors' hope that studies such as this one encourage more IACs in software testing and SE.

Let us recall from Section 2.2 that our SLR study [3] reviewed a set of 33 primary studies and we used the synthesized/classified set of observed challenges, patterns and anti-patterns discussed in those 33 primary studies. Those studies had mentioned the challenges, patterns and anti-patterns in qualitative/narrative manners however our work aimed at quantifying them in the context of the IAC projects selected in this work (Section 3.3). Essentially, the set of challenges, patterns and anti-patterns that we considered in this work was a 'union' of all those items in all the existing literature in this area. Thus, our study took a much more inclusive approach for analyzing those aspects in IAC projects. Since our approach was quantitative, one cannot directly compare our numerical data to the qualitative/narrative discussion of challenges, patterns and anti-patterns in the 33 primary studies. We hope to see further quantitative studies like ours in this area in future, which will enable cross-case comparisons of findings. Also, similar to various types of 'retrospective' analyses conducted in conventional project management (e.g., [48-50]), our motivation behind this study was to conduct retrospective analysis of a set IAC projects to find out what was challenging, what actions were beneficial (patterns), and what mistakes could be prevented (anti-patterns). A highlight of the results, their implications and a set of evidence-based actionable recommendations are presented in the following.

<u>RQ 1: Role of challenges, patterns and anti-patterns</u>

- RQ 1.1: Role of challenges: We summarize our observations and the corresponding recommendations next.
    - *Observation*: We found out that there is a quite a wide variation in the levels of challenges across the projects.
        - *Recommendation*: Prepare yourself for challenges (or: nothing is easy, or: embrace challenges)-Thus, we can suggest that researchers and practitioners should be prepared to face and deal with any of the 64 challenges listed in the SLR study [3].
    - *Observation:* The five top most observed challenges were: C04: University education not focused on industrial relevance, C35: Different requirements on novelty, C30: Different types of knowledge available (industry vs. academia), C28: Different expectations on quality of evidence in research, C29: Different focus on scale of solutions. We also observed that even if an IAC project does not possess challenges from many aspects, one single major challenge can lead to its halt/failure. Furthermore, it was observed that one single major challenge could waste the fruits of applying many patterns and lots of initial time/effort investment.

        - *Recommendation:* Watch out for the one and only major challenge- Both parties (academics and practitioners) should consider all the challenges early on and proactively work together to eliminate the risk of challenges in IAC projects. A lot of your efforts could go wasted by one single issue. We recommend that both parties (academics and practitioners) do not under-estimate the impact of challenges and proactively work together to eliminate the risk of challenges in IAC projects as early as possible.
    - *Observation:* As a lesson learnt, the authors have found, by experience, that researchers and practitioners should strive for aligning the viewpoints and 'meeting in the middle' (compromise), e.g., for the following challenges: different requirements on novelty (C35), different types of knowledge available (C30), different focus on scale of solutions (C29). For example, for C30, to decrease the knowledge gap between the two sides, we have found it very beneficial to arrange researchers visit and spend full-days or even weeks in the industry partner to gain some industrial knowledge (in the 'real world').
        - *Recommendation:* Compromise on perspectives: In case of having different viewpoints and opinions on subjects, learn from each other and come to a 'middle ground'. For example, when researchers and practitioners discuss and want to define requirements on novelty (C35), researchers may want too much technical (i.e., academic) novelty, but practitioners usually want an approach that does the work (solves the problem) which may or may not be novel. In such situations, by mutual understanding of the styles and perspectives, we have seen that it is possible to define an average novelty level which both sides would be happy with.
- RQ 1.2: Role of patterns (best practices)
    - *Observation*: If the both sides apply patterns, they could improve the chance of being successful in IAC.
        - *Recommendation*: Be proactive in applying the patterns (see Table 13 in the appendix)
- RQ 1.3: Role of anti-patterns
    - *Observation*: Similar to the retrospective analysis that we conducted for the pool of our projects, it is important to do so in each IAC to ensure learning and performing 'continuous improvement' in IAC.
        - *Recommendation*: Conduct retrospective analysis: During each project, in milestones, and after each IAC project, conduct retrospective analysis (e.g., [48-50]) to find what could be done better and what could be prevented (not done) to prevent issues.
    - *Observation*: The five top most observed anti-patterns were: AP6: Too much emphasis on industrial-need, AP4: Conducting one-way knowledge transfer only, AP9: Insufficient benefits presentation, AP12: Using lab experiments for argumentation before convincing stakeholders, and AP17: Not distinguishing experimental environment from real-life situations.
        - *Recommendation*: Avoid the top hurting anti-patterns (the above) at all cost
- RQ 1.4: Other additional challenges, success factors and anti-patterns
    - *Observation*: Based on the discussions in answering RQ 1.4, we would like to put forward the following two recommendations:
        - *Recommendations*: (1) Continuously conduct cost-benefit analysis of IAC projects, (2) Use, if applicable, the GQM+ strategy to align the technical directions of IAC projects with business strategies
    - *Observation*: We furthermore noticed that, from the practitioners' viewpoint, developing a new idea, software product or techniques to develop something new is valuable and there are a lot of incentives to work with research community in such initiatives. However, it is the authors' opinion that many practitioners in the software industry consider testing a necessary "evil" that requires not so highly intellectual skillset to perform. At the same time, we have observed that practitioners working in the projects discussed in this study have realized that there is a lot more into software testing that many industrial practitioners realize.
        - *Recommendations*: Communicate the importance of IAC in areas other than only software 'development'.
- RQ 1.5: Correlations and inter-relationship of challenges, patterns and anti-patterns with each other and with success measures
    - *Observation*: We empirically validated that: (1) The higher the amount of challenges, the higher possibility of committing anti-patterns, (2) The higher the rate of committing anti-patterns, the less the chance of success, and (3) When there were more challenges, we were proactive by applying more patterns.

Post-print

RQ 2: Country-specific issues and context-specific issues:

- RQ 2.1: Country-specific issues and differences (Canada versus Turkey)
  - *Observation*: Our initial effort in this regard showed that the project teams in Canada seemed to have similar opportunities of applying patterns while, for the case of Turkey, there seemed to be quite a wide variety in such opportunities. But further investigation is recommended.
- RQ 2.2: Issues specific to software testing?
  - *Observations*: Industry partners seem to have quite different expectation in terms of what researchers could do for them in terms of software testing
    - *Recommendations*: (1) Communicate early on the expectations and what researchers could help in terms of software testing, (2) Do not disclose the test results or number of defects found in the systems in public presentations or in papers if the industry partner is sensitive about it.

5.2 Threats to validity

In this section, we discuss potential threats to the validity of our study and steps we have taken to minimize or mitigate them. The threats are discussed in the context of the four types of threats to validity based on a standard checklist for validity threats presented in [84]: internal validity, construct validity, conclusion validity and external validity.

5.2.1 Internal validity

Internal validity is a property of scientific studies which reflects the extent to which a causal conclusion based on a study and the extracted data is warranted [84]. A threat to internal validity in this study lies in the selection bias (i.e., randomness of the objects or subjects of the study).

As discussed in Section 3.2, to reduce the selection bias, we used the 'stratified' sampling method [51], which is a type of probabilistic (representative) sampling. In this sampling method, the population is first divided into characteristics of importance for the research, which in our case is, by country, industry partner size, project scale, and project success level. Then the population was randomly sampled within each category or stratum. To keep our effort manageable, a pool of 10 IAC projects all in the area of software testing were sampled. We thus believe the internal validity threat is minimized.

5.2.2 Construct validity

Construct validity is concerned with the extent to which the object of study truly represents the theory behind the study [84]. Threats related to this type of validity in this study were suitability of RQs, nature of the projects and categorization / ranking scheme used for the data extraction.

To limit construct threats in this study, the GQM approach was used to preserve the tractability between research goal, RQs and metrics. The RQs and their sub-RQs were designed to cover our goal and different aspects of the IAC projects. The RQs were answered according to a categorization/ranking scheme for challenges, patterns and anti-patterns in IAC projects, adopted from a recent SLR study [3] in this area. That SLR study itself followed a systematic approach in which the threats to the validity were minimized as much as possible [3]. In terms of nature of the projects, they truly represent the 'theory behind the study' (the IAC in software testing).

As discussed in Section 3.2, the assessment of each ranking of challenges, patterns and anti-patterns was entered by one author and was then peer reviewed by another author (or industry partner), both of whom were actually involved in the given project. We ensured that for each assessment, consensus was reached between the first assessor and the peer reviewer. Commenting feature of the online Google Document spreadsheet tool (Figure 4) and Skype tele-conferencing were used to discuss disagreements and reach consensus and, thus, to ensure high quality of our quantitative assessments. All of the above steps ensured that threats to construct validity were minimized as much as possible.

5.2.3 Conclusion validity

Conclusion validity of a study deals with whether correct conclusions are reached through rigorous and repeatable treatment [84]. We attempted to conclude, both quantitatively and qualitatively, that certain challenges, patterns and anti-patterns were more applicable in certain IAC projects compared to others. For each RQ, we attempted to reduce the conclusion validity bias by seeking support from the statistical results. Thus, all the conclusions that we drew in this study are strictly traceable to data.

Post-print

5.2.4 External validity

External validity is concerned with to what extent the results of our study can be generalized [84]. As discussed above in threats to internal validity, all the efforts were made to minimize the selection bias, which is an important factor for both internal and external validity.

Although we believe that our sample project size and geographic distribution of samples (across two different countries) are quite reasonable to make a rough conclusion for the situation of IAC in Canada and Turkey. However, needless to say, it is obvious that IAC projects are different from each other as the participants and context factors influence their outcome to a great extent. Thus, although the results of our study cannot be fully generalized to other projects, they provide 'benchmarks', evidence, recommendations and baselines that would benefit SE researchers and practitioners in conducting successful projects in software testing and in software engineering in general in the future.

Also, note that our findings in this study are mainly within the field of IAC. Beyond this field, we had no intention to generalize our results. Therefore, few problems with external validity are worthy of substantial attention.

6 CONCLUSIONS AND FUTURE WORKS

In conclusion, the authors believe that, for the SE research community to have a meaningful future, there is a critical need for better connections and collaborations between industry and academia. Industry-relevant research and making an impact in the software industry involves more than just producing research results and disseminating them in publications which are, honestly, not read regularly by practitioners. Conducting industry-relevant research requires close cooperation and collaboration between industry and academia throughout the entire research process, from the identification of the problem all the way to delivering the result and publications. To ensure win-win for the two sides, they need to follow lean collaboration models and success criteria briefly discussed in this article, including but not limited to the need for continuous cost-benefit analysis of joint R&D efforts.

In terms of the road ahead and future work, there is much to be done in this direction. We need to find more ways to bridge the gap, learn from other disciplines, e.g., [36-44], especially other engineering fields, on how they approach IAC in their fields, and also to share our SE IAC experiences with them. We need to analyze the award mechanisms and use that information to further motivate researchers and practitioners for collaborations.

Furthermore, further investigations in the direction of success factors, critical success factors (CSF) and failure factors in IAC are needed. The issue of success patterns in this context somewhat relates to critical success factors (CSF) and failure factors and the related body of knowledge in the conventional software project management, e.g. [77-79, 85, 86]. While attempts have been made to offer limited lists of success factors for IAC projects, e.g. in [30, 32], more comprehensive studies in this context are needed. As we realize, IAC projects are quite different in nature compared to conventional software projects and thus the related success factors would be different.

More local and international events are needed to bring the two groups together and foster collaborations. We have observed that, usually, pinpointing mutually-interesting and valuable project topics is one of the hardest phases of these projects, and thus, guidelines on how to "break the ice" and start the discussions on finding mutually-relevant topics are needed. As highlighted by the quantitative and qualitative assessments in this study, the authors and many of their colleagues have empirically observed that IAC can be truly a win-win for both sides if they are planned, operated and finalized/disseminated "properly". So, *let's talk to each other and collaborate more!*

ACKNOWLEDGEMENTS

Vahid Garousi was partially supported by several internal grants by the Hacettepe University and the Scientific and Technological Research Council of Turkey (TÜBİTAK).

REFERENCES

[1] L. Briand, "Embracing the Engineering Side of Software Engineering," *IEEE Software,* vol. 29, pp. 96-96, 2012.

[2] R. L. Glass, *Software Creativity 2.0*: developer.* Books, 2006.

[3] V. Garousi, K. Petersen, and B. Özkan, "Industry-Academia Collaborations in Software Engineering: A Systematic Literature Review," *Technical Report, Hacettepe University Software Engineering Research Group, HUSE-2015-01, https://drive.google.com/open?id=0B6dKdxaNjBENSWRwRlNJbExYUWc,* 2015.

[4] T. Gorschek, C. Wohlin, P. Carre, and S. Larsson, "A Model for Technology Transfer in Practice," *IEEE Software,* vol. 23, pp. 88-95, 2006.

[5] K. Petersen, C. Gencel, N. Asghari, D. Baca, and S. Betz, "Action research as a model for industry-academia collaboration in the software engineering context," in *Proceedings of the ACM International Workshop on Long-Term Industrial Collaboration on Software Engineering*, 2014, pp. 55-62.

[6] A. Sandberg, L. Pareto, and T. Arts, "Agile collaborative research: Action principles for industry-academia collaboration," *IEEE Software,* vol. 28, pp. 74-83, 2011.

[7] J. Franch Gutiérrez, D. Ameller, C. P. Ayala Martínez, and J. Cabot Sagrera, "Bridging the gap among academics and practitioners in non-functional requirements management: some reflections and proposals for the future," in *Modelling and quality in requirements engineering*, M. u. Vannerdat, Ed., ed, 2012, pp. 267-273.

[8] A. M. Connor, J. Buchan, and K. Petrova, "Bridging the Research-Practice Gap in Requirements Engineering through Effective Teaching and Peer Learning," in *International Conference on Information Technology: New Generations*, 2009, pp. 678-683.

[9] S. Jain, M. Ali Babar, and J. Fernandez, "Conducting empirical studies in industry: Balancing rigor and relevance," in *International Workshop on Conducting Empirical Studies in Industry*, 2013, pp. 9-14.

[10] L. J. Osterweil, C. Ghezzi, J. Kramer, and A. L. Wolf, "Determining the Impact of Software Engineering Research on Practice," *IEEE Computer,* vol. 41, pp. 39-49, 2008.

[11] W. Raschke, M. Zilli, J. Loinig, R. Weiss, C. Steger, and C. Kreiner, "Embedding research in the industrial field: a case of a transition to a software product line," presented at the Proceedings of the international workshop on Long-term industrial collaboration on software engineering, Vasteras, Sweden, 2014.

[12] C. Wohlin, "Empirical software engineering research with industry: Top 10 challenges," in *International Workshop on Conducting Empirical Studies in Industry*, 2013, pp. 43-46.

[13] M. T. Baldassarre, D. Caivano, and G. Visaggio, "Empirical studies for innovation dissemination: ten years of experience," presented at the Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, Porto de Galinhas, Brazil, 2013.

[14] E. P. Enoiu and A. Čaušević, "Enablers and impediments for collaborative research in software testing: an empirical exploration," presented at the Proceedings of the international workshop on Long-term industrial collaboration on software engineering, Vasteras, Sweden, 2014.

[15] K. Petersen and E. Engstre, "Finding relevant research solutions for practical problems: the serp taxonomy architecture," presented at the Proceedings of the international workshop on Long-term industrial collaboration on software engineering, Vasteras, Sweden, 2014.

[16] A. Kanso and D. Monette, "Foundations for long-term collaborative research," presented at the Proceedings of the international workshop on Long-term industrial collaboration on software engineering, Vasteras, Sweden, 2014.

[17] P. Runeson, S. Minör, and J. Svenér, "Get the cogs in synch-time horizon aspects of industry-academia collaboration," in *Proceedings of the ACM International Workshop on Long-Term Industrial Collaboration on Software Engineering, Co-located with ASE 2014*, 2014, pp. 25-28.

[18] D. Rombach, M. Ciolkowski, R. Jeffery, O. Laitenberger, F. McGarry, and F. Shull, "Impact of research on practice in the field of inspections, reviews and walkthroughs: learning from successful industrial uses," *SIGSOFT Softw. Eng. Notes,* vol. 33, pp. 26-35, 2008.

[19] P. Krishnan, K. Ross, and P. Pari-Salas, "Industry academia collaboration: An experience report at a small University," in *Proceedings of Conference on Software Engineering Education and Training,* 2009, pp. 117-121.

[20] P. Runeson, "It takes two to tango: An experience report on Industry-Academia collaboration," in *Proceedings of International Conference on Software Testing, Verification and Validation*, 2012, pp. 872-877.

[21] S. J. Lamprecht and G. J. Van Rooyen, "Models for technology research collaboration between industry and academia in South Africa," in *Proceedings of the Software Engineering Colloquium*, 2012, pp. 11-17.

Post-print

[22] S. Martínez-Fernández and H. M. Marques, "Practical experiences in designing and conducting empirical studies in industry-academia collaboration," presented at the Proceedings of the International Workshop on Conducting Empirical Studies in Industry, Hyderabad, India, 2014.

[23] P. Morris, M. Masera, and M. Wilikens, "Requirements engineering and industrial uptake," in *Proceedings of International Conference on Requirements Engineering*, 1998, pp. 130-137.

[24] H. Kaindl, S. Brinkkemper, J. A. Bubenko Jr, B. Farbey, S. J. Greenspan, C. L. Heitmeyer*, *et al.*, "Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda," *Requirements Engineering,* vol. 7, pp. 113-123, 2002.

[25] D. Rombach and R. Achatz, "Research Collaborations between Academia and Industry," in *Symposium on Future of Software Engineering*, 2007, pp. 29-36.

[26] R. L. Glass and A. Hunt, *Software Conflict 2.0: The Art And Science of Software Engineering*: Developer.* Books, 2006.

[27] C. Wohlin, "Software Engineering Research under the Lamppost," in *Proceedings of the International Joint Conference on Software Technologies*, 2013.

[28] S. Eldh, "Some researcher considerations when conducting empirical studies in industry," in *International Workshop on Conducting Empirical Studies in Industry*, 2013, pp. 69-70.

[29] C. Wohlin and B. Regnell, "Strategies for industrial relevance in software engineering education," *Journal of Systems and Software,* vol. 49, pp. 125-134, 1999.

[30] P. Grunbacher and R. Rabiser, "Success factors for empirical studies in industry-academia collaboration: A reflection," in *International Workshop on Conducting Empirical Studies in Industry*, 2013, pp. 27-32.

[31] P. Runeson and S. Minör, "The 4+1 view model of industry-academia collaboration " in *Proceedings of the ACM International Workshop on Long-Term Industrial Collaboration on Software Engineering, Co-located with ASE 2014*, 2014, pp. 21-24.

[32] C. Wohlin, A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, *et al.*, "The Success Factors Powering Industry-Academia Collaboration," *IEEE Software,* vol. 29, pp. 67-73, 2012.

[33] A. T. Misirli, H. Erdogmus, N. Juristo, and O. Dieste, "Topic selection in industry experiments," presented at the Proceedings of the International Workshop on Conducting Empirical Studies in Industry, Hyderabad, India, 2014.

[34] S. L. Pfleeger, "Understanding and improving technology transfer in software engineering," *J. Syst. Softw.,* vol. 47, pp. 111-124, 1999.

[35] L. C. Briand, "Useful software engineering research - leading a double-agent life," in *IEEE International Conference on Software Maintenance*, 2011, pp. 2-2.

[36] S. Palkar, "Industry-academia collaboration, expectations, and experiences," *ACM Inroads,* vol. 4, pp. 56-58, 2013.

[37] Y. Kato, "Industry-academia collaboration program for commercialization of new dry cleaning system," *AIST Today (International Edition),* p. 11, 2005.

[38] M. T. Harrigan, "Industrial biotechnology education: A model of collaboration between industry and academia," *Biochemistry and Molecular Biology Education,* vol. 31, pp. 142-144, 2003.

[39] M. Fairhurst, "Academia and industry collaboration to drive biometrics boom," *Biometric Technology Today,* vol. 2013, pp. 10-11, 2013.

[40] C. B. Mullins, "A university administrator's view of academia--industry collaboration," *Circulation,* vol. 72, pp. I8-12, 1985.

[41] F. D. Price, "Industry and academia in collaboration: the Pfizer experience," *Circulation,* vol. 72, pp. I13-17, 1985.

[42] C. J. Thomas and J. C. McKew, "Playing well with others! Initiating and sustaining successful collaborations between industry, academia and government," *Current Topics in Medicinal Chemistry,* vol. 14, pp. 291-293, 2014.

[43] A. S. Mujumdar, "Editorial on industry-academia collaboration in R&D," *Drying Technology,* vol. 28, pp. 431-432, 2010.

Post-print

[44] K. Guruprasad, "Multi-faceted industry-academia collaboration," *Evolving Corporate Education Strategies for Developing Countries: The Role of Universities,* pp. 90-102, 2013.

[45] M. Perkmann, V. Tartari, M. McKelvey, E. Autio, A. Broström, P. D'Este*, et al.*, "Academic engagement and commercialisation: A review of the literature on university–industry relations," *Research Policy,* vol. 42, pp. 423-442, 3// 2013.

[46] S. Ankrah and O. Al-Tabbaa, "Universities–industry collaboration: A systematic review," *Scandinavian Journal of Management,* vol. In press, 2015.

[47] R. v. Solingen and E. Berghout, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*: McGraw-Hill, 1999.

[48] M. S. Deutsch, "An exploratory analysis relating the software project management process to project success," *Engineering Management, IEEE Transactions on,* vol. 38, pp. 365-375, 1991.

[49] K. Jugdev and R. Müller, "A retrospective look at our evolving understanding of project success," *Project Management Journal,* vol. 36, pp. 19-31, 2005.

[50] M. Leszak, D. E. Perry, and D. Stoll, "Classification and evaluation of defects in a project retrospective," *Journal of Systems and Software,* vol. 61, pp. 173-187, 4/1/ 2002.

[51] T. R. Lunsford and B. R. Lunsford, "The Research Sample, Part I: Sampling," *J. Prosthetics and Orthotics,* vol. 7, pp. 105-112, 1995.

[52] C. Wiederseiner, V. Garousi, and M. Smith, "Tool Support for Automated Traceability Analysis in Embedded Software Systems," in *Proc. of IEEE International Conference on Embedded Software and System (ICESS)*, 2011, pp. 1109-1117.

[53] C. Pinheiro, V. Garousi, F. Maurer, and J. Sillito, "Introducing Automated Environment Configuration Testing in an Industrial Setting," in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, Workshop on Software Test Automation, Practice, and Standardization*, 2010, pp. 186-191.

[54] C. Wiederseiner, S. A. Jolly, V. Garousi, and M. M. Eskandar, "An Open-Source Tool for Automated Generation of Black-box xUnit Test Code and its Industrial Evaluation," in *Proceedings of the International Conference on Testing: Academic and Industrial Conference - Practice and Research Techniques (TAIC PART)*, 2010, pp. 118-128.

[55] S. A. Jolly, V. Garousi, and M. M. Eskandar, "Automated Unit Testing of a SCADA Control Software: An Industrial Case Study based on Action Research," in *IEEE International Conference on Software Testing, Verification and Validation (ICST)*, 2012, pp. 400-409.

[56] Y. Amannejad, V. Garousi, R. Irving, and Z. Sahaf, "A Search-based Approach for Cost-Effective Software Test Automation Decision Support and an Industrial Case Study," in *Proc. of International Workshop on Regression Testing, co-located with the Sixth IEEE International Conference on Software Testing, Verification, and Validation*, 2014, pp. 302-311.

[57] Z. Sahaf, V. Garousi, D. Pfahl, R. Irving, and Y. Amannejad, "When to Automate Software Testing? Decision Support based on System Dynamics – An Industrial Case Study," in *Proc. of International Conference on Software and Systems Process*, 2014, pp. 149-158.

[58] V. Garousi and D. Pfahl, "When to Automate Software Testing? A Decision-Support Approach Based on Process Simulation," *In Press, Wiley Journal of Software: Evolution and Process,* 2015.

[59] D. Akdur and V. Garousi, "Model-Driven Engineering in Support of Development, Test and Maintenance of Middleware Communication Protocols: A Preliminary Approach " in *International Conference on Model-Driven Engineering and Software Development (MODELSWARD), Doctoral Consortium*, 2015, pp. 11-19.

[60] V. Garousi and J. Zhi, "A Survey of Software Testing Practices in Canada," *Journal of Systems and Software,* vol. 86, pp. 1354-1376, 2013.

[61] V. Garousi and T. Varma, "A Replicated Survey of Software Testing Practices in the Canadian Province of Alberta: What has Changed from 2004 to 2009?," *Journal of Systems and Software,* vol. 83, pp. 2251-2262, 2010.

Postprint

[62] V. Garousi, A. Coşkunçay, A. B. Can, and O. Demirörs, "A Survey of Software Engineering Practices in Turkey," *Journal of Systems and Software,* vol. 108, pp. 148-177, 2015.

[63] V. Garousi, A. Coşkunçay, A. B. Can, and O. Demirörs, "A Survey of Software Testing Practices in Turkey," in *Proceedings of the Turkish National Software Engineering Symposium "Ulusal Yazılım Mühendisliği Sempozyumu" (UYMS)*, 2013.

[64] D. S. Cruzes and T. Dyba, "Research synthesis in software engineering: A tertiary study," *Information and Software Technology,* vol. 53, pp. 440-455, 2011.

[65] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*: John Wiley & Sons, 2012.

[66] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen, "Action research," *Communications of the ACM,* vol. 42, pp. 94-97, 1999.

[67] G. M. Sullivan and A. R. Artino, "Analyzing and Interpreting Data From Likert-Type Scales," *Journal of Graduate Medical Education,* vol. 5, pp. 541-542, 2013.

[68] V. Garousi, K. Petersen, and B. Özkan, "Challenges, Success Patterns and Anti-Patterns in Industry-Academia Collaborations in Software Engineering: a Systematic Literature Review," *Technical Report, Hacettepe University Software Engineering Research Group, HUSE-2015-01,* 2015.

[69] V. Garousi, "Success Stories in Industrial Software Testing Research and Education: A Canadian Perspective," *Marmara Research Center, Scientific and Technological Research Council of Turkey (BILGEM, TÜBITAK), Istanbul, Turkey,* July 2012.

[70] V. Garousi, "Incorporating Real-World Industrial Testing Projects in Software Testing Courses: Opportunities, Challenges, and Lessons Learned," in *Proceedings of the IEEE Conference on Software Engineering Education and Training (CSEE&T)*, 2011, pp. 396-400.

[71] V. Garousi, "An Open Modern Software Testing Laboratory Courseware: An Experience Report " in *Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training*, 2010, pp. 177-184.

[72] G. V. B. Subrahmanyam, "A Dynamic Framework for Software Engineering Education Curriculum to Reduce the Gap between the Software Organizations and Software Educational Institutions," in *Software Engineering Education and Training, 2009. CSEET '09. 22nd Conference on*, 2009, pp. 248-254.

[73] C. O'Leary, D. Lawless, D. Gordon, H. Li, and K. Bechkoum, "Developing a Software Engineering Curriculum for the Emerging Software Industry in China," in *Software Engineering Education and Training, 2006. Proceedings. 19th Conference on*, 2006, pp. 115-122.

[74] V. Garousi, "A Genetic Algorithm-based Stress Test Requirements Generator Tool and its Empirical Evaluation," *IEEE Transactions on Software Engineering, Special Issue on Search-Based Software Engineering (SBSE),* vol. 36, pp. 778-797, 2010.

[75] V. Garousi, L. Briand, and Y. Labiche, "Traffic-aware Stress Testing of Distributed Real-Time Systems Based on UML Models using Genetic Algorithms," *Elsevier Journal of Systems and Software, Special Issue on Model-Based Software Testing,* vol. 81, pp. 161-185, 2008.

[76] V. Garousi, "Empirical Analysis of a Genetic Algorithm-based Stress Test Technique for Distributed Real-Time Systems," in *Proc. of the Genetic and Evolutionary Computation Conference, Search-Based Software Engineering (SBSE) track*, 2008, pp. 1743-1750.

[77] A. Ahimbisibwe, R. Y. Cavana, and U. Daellenbach, "A contingency fit model of critical success factors for software development projects," *Journal of Enterprise Information Management,* vol. 28, pp. 7-33, 2015.

[78] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software,* vol. 81, pp. 961-971, 6// 2008.

[79] J. S. Reel, "Critical success factors in software projects," *IEEE Software,* vol. 16, pp. 18-23, 1999.

[80] J. C. Maxwell and J. Dornan, *How to Influence People: Make a Difference in Your World*: Thomas Nelson Inc, 2013.

Post-print

[81] C. Wiederseiner, S. A. Jolly, and V. Garousi, "AutoBBUT - Automated Code Generation for Black-box Unit Testing," http://code.google.com/p/autobbut/, Last accessed: Sept. 2012.

[82] S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, and P. Grünbacher, *Value-Based Software Engineering*: Springer, 2006.

[83] M. Adler and E. Ziglio, *Gazing Into the Oracle: The Delphi Method and Its Application to Social Policy and Public Health*: Jessica Kingsley Publishers, 1996.

[84] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*: Kluwer Academic Publishers, 2000.

[85] R. N. Charette, "Why software fails," *IEEE Spectrum,* vol. 42, pp. 42-49, 2005.

[86] K. T. Yeo, "Critical failure factors in information system projects," *International Journal of Project Management,* vol. 20, pp. 241-246, 4// 2002.

APPENDIX-EXCERPTS FROM THE SLR STUDY

CHALLENGES IMPEDING SUCCESS

We show in Table 12 the categories and the detailed list of all challenges. In the following we provide an overview of the findings for the categories.

Table 12 - List of all challenges (from the SLR study [3])

Category	ID	Challenge description
Lack of research relevance	C01	Results produced through research are not relevant for practice
	C02	Researchers do not understand the relevant problems from an industry point of view
	C03	Results produced by research are not measurable and exploitable (mechanisms for exploiting them are missing)
	C04	University education not focused on industrial relevance
	C05	Research topic selection not driven by relevance
Research method related	C06	Addressing the validity of the research when industry is involved: Generalizability, control and confounding factors, biases, subjectivity, repeatability, sample size, and repeatability
	C07	Running a flexible research project/method is challenging
	C08	Research in its nature is risky
	C09	Difficult to evaluate whether research addresses future needs in practice making it challenging to decide on solutions
	C10	Integrating existing solutions in the already existing context
Lack of training, experience, and skills	C11	Deficiencies in software engineering education
	C12	Lack of training, experience, and skills (general)
	C13	Deficiencies in skills of practitioners to work with the research solution
	C14	Deficiencies of knowledge by the researcher of the company context and technologies used in practice
	C15	Deficiencies in research skills from practitioners
Lack or drop of interest / commitment	C16	Lack of commitment to provide access and time
	C17	Lack of commitment to assess research results and forums (such as conferences)
	C18	Lack of commitment to invest money
	C19	Lack of commitment due to human factors (inertia, admit the need for external collaboration, not invested here syndrome)
	C20	Lack of commitment due to competitive business
Mismatch between industry and academia	C21	Different time horizons between industry and academia
	C22	Different interests and objectives
	C23	Different perception of what solutions and outcomes are useful
	C24	Different terminology and ways of communicating
	C25	Different reward systems
	C26	Different communication channels and directions of information flow
	C27	Different cultures
	C28	Different expectations on quality of evidence in research
	C29	Different focus on scale of solutions

Post print

	C30	Different types of knowledge available (industry vs. academia)
	C31	Willingness for technology transfer from academia larger than acceptance of transfer from industry
	C32	Different contexts
	C33	Different business models
	C34	Different perception of challenges
	C35	Different requirements on novelty
Communication-related issues	C36	Communication gaps between researchers and practitioners
	C37	Difficulty of managing multiple research partners
	C38	Difficulty to elicit information from developers
	C39	Fulfilling the need of communicating on time-frames, topics, and responsibilities
	C40	Lack of prior relationships between a company and academia
Human and organizational factors	C41	Resistance to change and inflexibility
	C42	Lack of organizational stability and continuity
	C43	Difficulties in training practitioners due to high training cost and lack availability of time due to market pressure
	C44	Intangible human factors with organization-wide impact
	C45	Competition between industrial and external researchers
	C46	Finding a champion
	C47	Solution incompatible with organizational culture
Management-related issues	C48	Difficulty to achieve clear and realistic ambitions and goals in projects
	C49	High investment in time and effort needed
	C50	Difficult to find the right project infrastructure (management, collaboration environments)
	C51	Difficulty in competence management to integrate external competences
	C52	Time-critical windows of opportunity for product research
	C53	Lack of openness to disclose weaknesses
	C54	Loss of champions in projects
	C55	Unwillingness to disclose weaknesses (areas needing improvements)
Resource-related issues	C56	Lack of resources due to high investment in terms of resources (people's time and effort) – both from industry and academia side
	C57	Financial investment risky from industry side
	C58	Financial investment risky from academic side
	C59	Licensing restrictions on tools
	C60	Lack of resources to provide technical support for research solutions
Contractual, and privacy concerns	C61	Intellectual property rights and privacy limit access to data
	C62	Difficulty in managing and handling intellectual property rights (skills, definition of requirements, handling of transfer of rights)
	C63	Missing trust and respect
	C64	Incorporating new methods and solutions in research contacts

BEST PRACTICES (SUCCESS PATTERNS): WHAT TO DO TO ENSURE SUCCESS

Table 13- List of all best practices (success patterns) (from the SLR study [3, 68])

Category	ID	Best practice description
Knowledge management (communication, terminology, transfer, training and skills)	BP1	Provide examples of challenges and solutions
	BP2	Need for continuous learning and for training on both sides
	BP3	Improvements to university and research communities
	BP4	Researchers should tune their social skills
	BP5	Establish common and simple terminology (vocabulary)
	BP6	Researchers should better open up knowledge to practitioners
	BP7	Run workshops and seminars
	BP8	Use existing works (than just inventing yet other approaches)
	BP9	Need for prior expertise
	BP10	Effective communication
	BP11	Create user documentation
	BP12	Establish a steering group
	BP13	Effective proprietary data management
	BP14	Promote the solution and its ease of use using evidence
Ensu	BP15	Ensure management engagement in the industry side

	BP16	Need for champions and their attitudes
	BP17	Make long-term commitments
	BP18	Proper presentation and communication by researchers in early meetings
	BP19	Proper topic selection
	BP20	Create and encourage buy-in from industry side
	BP21	Researchers shall take responsibility and commit resources for the whole research lifecycle
	BP22	Prior Positive Experience
	BP23	Researchers shall industry partners properly
	BP24	Keep the team focused during the project
	BP25	Transfer ownership of approach to industry folks
	BP26	Encourage access to industry systems and data
	BP27	Industry shall acknowledge value of research ideas
	BP28	Attention to company needs
Consider and understand industry's needs, challenges, goals and problems	BP29	Base research on real-world problems
	BP30	Use systematic approaches, e.g., ‘problem frames, to classify and analyze software engineering problems
	BP31	Involve practitioners in problem formulation
	BP32	Attend to not only industry needs, but also goals
	BP33	Continued contact of researcher with industrial demands during the project
	BP34	Find the most problematic pain points
	BP35	Control formulation of problems to be research and not consulting
	BP36	Formulate non-trivial problems
	BP37	Consider industry's long-term needs
	BP38	Define coherent sets of challenges
	BP39	To do for Practitioners: Practitioners should assist researchers in studying and understanding diffusion theory
Ensure giving explicit industry	BP40	Show benefits of the research solutions for the industrial partner
	BP41	Important quality aspects of the solution (e.g., sustainability, adaptability, highly customizable, scalability)
	BP42	Use industrial data in research
	BP43	Solution should be cost-efficient (ROI)
Have mutual respect, understand	BP44	Establish trust
	BP45	Establish common objectives between industry and academia
	BP46	Friendliness and reciprocal respect
	BP47	Appreciate each other’s strengths
	BP48	To academic researchers: value practitioners experience
Be Agile	BP50	Be Agile (use iterations/increments)
	BP51	Convert large projects to several smaller ones
Work in (as) a team and involving	BP52	Work in (as) a team
	BP53	Find the right team and time-scale for collaborations
	BP54	Change roles over time and involve different people over time
	BP55	Involve the "right" practitioners
	BP56	Write papers together (joint authorship)
Consider and manage risks and	BP57	Consider the organizational stability of the industry partner as a risk factor
	BP58	Address risks and weaknesses in the collaboration proactively
	BP59	Realize limitations of the lab experiments
	BP60	Manage time-related risks
	BP61	Share risk-taking
Researcher's on-site presence and access	BP62	Researchers should be co-located and be present on the industry site
	BP63	Provide easy and frequent access for the researchers (to data and to practitioners)
	BP64	Participate in activities beyond the research project in the company
	BP65	Have frequent interaction through meetings
	BP66	Get access to corporate meeting forums
Follow a proper research/ data	BP67	Use the case study method
	BP68	Use retrospective analysis of experiments
	BP69	Use situational method engineering
	BP70	Use the design science method

Pre-print

	BP71	Use the reflective systems development approach
	BP72	Use evidence-based software engineering
	BP73	Use flexible research designs
	BP74	Use systematic approaches to build taxonomies supporting communication
	BP75	Investigate different contexts for generalizability
	BP76	Use established guidelines and data collection methods (interview, survey, etc.)
	BP77	Collect different kinds of data (quantitative - qualitative, triangulation)
	BP78	Personally interact with the practitioners during data collection
	BP79	Place more emphasis on empirical research in realistic contexts
	BP80	Agree on confidentiality before collecting data
	BP81	Aim for "just enough" rigor
	BP82	Assure relaxed feeling of participants (e.g. in surveys)
	BP83	Collect archival data prior to conducting the research project
	BP84	Discuss and record observations immediately
	BP85	Evaluate your role as a researcher (Software engineering researchers should stop seeing themselves as computer scientists)
	BP86	Report negative results
Manage funding/recruiting/Partnerships and contracting privacy	BP87	Manage intellectual property rights (flexible and simple approach)
	BP88	"Employ" the researcher (e.g. put in status of intern, part-time leave from university, etc.)
	BP89	Collaborate with few high-quality external partners
	BP90	Embrace research negotiations (contractual)
	BP91	Employ researchers (graduate) with industry background
	BP92	Establish a partnership/joint project with the industry
	BP93	Establish a research institute to facilitate collaboration and transfer
	BP94	Fund small research projects
	BP95	Involve industry partners in research education (PhD)
	BP96	Research should not be free
	BP97	Build joint transfer test labs as a bridge for technology transfer
	BP98	Choose a partner complementing the innovation process of the company well
	BP99	Create long term/high cost research and development project proposals
Under stand the contex	BP100	Be aware of and identify context factors that influence and constrain the research results
	BP101	Gain an inside view of the practices used at the company
	BP102	Learn the domain and vocabulary
Efficient research project management	BP103	Plan the research project (time planning, estimation, collaboration, alignment with project goals)
	BP104	Decrease overhead and waste in research project administration
	BP105	Assure consistent reporting across documentation produced in research (reports, posters, etc.)
	BP106	Assure the availability of time for adequate roles represented by practitioners to participate in research activities
	BP107	Design effective reward structures for good practice
	BP108	Ensure that end research results hit the right trade-offs (e.g., quality and cost)
	BP109	Integrate research into daily work
	BP110	Save time of practitioners participating in research (e.g. in experiments)
	BP111	Utilize Ph.D. students as resources in projects
Conduct measurement/ assessment	BP112	Establish a measurement program and define measurable objectives
	BP113	Measure Return of Investment (ROI)
	BP114	Combine quantitative and qualitative information to evaluate projects
	BP115	Develop a set of guidelines to evaluate bodies of evidence
	BP116	Evaluation criteria should support the R&D project
	BP117	Measure innovativeness (innovation benchmarking)
	BP118	Measure solution stability as an indicator for applicability
Test pilot soluti ons	BP119	Test the solution in the lab/academic environment
	BP120	Pilot the solution with industry practitioners
	BP121	Test the solution through a proof of concept

	BP122	Build research prototypes
	BP123	Have a separate academic solution branch from an industrial solution branch to further evolve the solution
Provide tool support for solutions	BP124	Provide technical support and documentation for academic tools
	BP125	Assure the usability of the user interface - provide interfaces familiar to practitioners
	BP126	Assure the flexibility of the tools
	BP127	Agree on the licensing model for the tools produced
	BP128	Encourage the use of CASE tools

ANTI-PATTERNS: WHAT NOT TO DO TO ENSURE SUCCESS

Table 14- List of all anti-patterns (from the SLR study [3, 68])

Category	ID	Anti-pattern description
Self-Centric Approach	AP1	Building solutions alone
	AP2	Vested Interest
	AP3	Using scientific publications / evidence / taxonomies as the basis for communication
	AP4	Assuming one-way knowledge transfer
	AP5	Skewing the scientific results
	AP6	Too much emphasis on industrial-need
	AP7	Assuming that research projects can only be initiated by the academic side
	AP8	Assuming that research is free
Stakeholder Commitment and Benefits Presentation Related Anti-Patterns	AP9	Insufficient Benefits Presentation
	AP10	Accepting Problem Descriptions from few stakeholders
	AP11	Missing management support
	AP12	Using experiments for argumentation before convincing stakeholders
	AP13	Taking Management as the best Research Target
	AP14	Assuming an early optimism from industry
	AP15	Not relating evidence to an effect
Research Design Related Anti-Patterns	AP16	Omitting the State of the Art and existing practices
	AP17	Not distinguishing experimental environment from real life situations
	AP18	Overlooking Context dependency
	AP19	Insufficient time allocation for exploitation of the results
	AP20	Using anecdotal evidence
	AP21	Collecting one type of evidence
	AP22	Incoherent empirical studies
Unstructured Decision	AP23	Poorly defined or unbalanced decision-making structures and roles for collaborative research
	AP24	Careless selection of the champions
Poor Change Management	AP25	Start big and work waterfall
	AP26	Avoid the need to revise data measurements and data collection plans
	AP27	Assume a stable context
	AP28	Discourage opportunism
	AP29	Forcing change
Ignoring Project Organizational Product Characteristics	AP30	Underestimating Complexity
	AP31	Not spending sufficient time at the organization
	AP32	Overlooking required personnel skills
	AP33	Overlooking the ability to change
	AP34	IPR Over focusing
Ineffective Communication	AP35	Unorganized communication of research activities and results
	AP36	Not involving new members into face-to-face interviews

Preprint