Experience of Industry Case Studies: A Comparison of Multi-Case and Embedded Case Study Methods

Julian M. Bass
University of Salford
j.bass@salford.ac.uk

Sarah Beecham
Lero, the Irish Software Research Centre,
University of Limerick, Ireland
sarah.beecham@lero.ie

John Noll
University of East London
j.noll@uel.ac.uk

ABSTRACT

Context: Case studies are a useful approach for conducting empirical studies of software engineering, in part because they allow a phenomenon to be studied in its real-world context. However, given that there are several kinds of case studies, each with its own strengths and weaknesses, researchers need to know how to choose which kind to employ for a specific research study.

Aim: The objective of this research is to compare two case study approaches: embedded, longitudinal case studies, and multi-case studies.

Approach: We compared two actual software engineering case studies: a multi-case study involving interviews with 46 practitioners at 9 international companies engaged in offshoring and outsourcing, and a single case, participant observation embedded case study lasting 13 months in a mid-sized Irish software company. Both case studies were exploring similar problems of understanding the activities performed by members of scrum development teams.

Results: We found that both multi-case and embedded case studies are suitable for exploratory research (hypothesis development) but that embedded research may also be more suitable for explanatory research (hypothesis testing). We also found that longitudinal case studies offer better confirmability, while multi-case studies offer better transferability.

Conclusion: We propose a set of illustrative research questions to assist with the selection of the appropriate case study method.

CCS CONCEPTS

• **Software and its engineering** → **Software creation and management**; *Software development process management*; Collaboration in software development;

KEYWORDS

Case Study Methods, Empirical Studies in Industry, Multi-case Study, Embedded Case Study, Cross-case Analysis

ACM Reference Format:
Julian M. Bass, Sarah Beecham, and John Noll. 2018. Experience of Industry Case Studies: A Comparison of Multi-Case and Embedded Case Study Methods. In *CESI'18: CESI'18:IEEE/ACM 6th International Workshop on Conducting Empirical Studies in Indus, May 28, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3193965.3193967

CESI'18, May 28, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5736-4/18/05…$15.00
https://doi.org/10.1145/3193965.3193967

1 INTRODUCTION

Case study research is one of several methodologies employed for empirical studies of software engineering in industry. According to Yin, a case study is "an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident" [36, p. 18]. Case studies can generate new hypotheses, and test existing hypotheses [16]. Case studies "provide a systematic way of looking at events, collecting data, analysing information, and reporting the results. As a result, case study researchers may gain a greater understanding of why something happened as it did, and what might be important to investigate in future research [35]."

Under the broad heading of case study research are different approaches to conducting case studies, including *holistic* that focuses on an organization as a whole, and *embedded*, which examines different "units of analysis" within the broader organizational context, possibly from different perspectives or using different techniques [32, 36]. Case studies can also vary by size along different dimensions, such as single vs. multiple cases, or longitudinal studies that examine a phenomenon over an extended period of time [30].

Each methodology has strengths and weaknesses. This research investigates similarities and differences between embedded case studies (longitudinal research conducted within a single organisation) and multi-case studies (conducted across multiple organisations). We want to identify criteria for selecting deep embedded approaches or broad cross-company studies. While methods are sometimes dictated by circumstances (such as an opportunity to collaborate with a partner company); we want to understand more clearly the strengths and weaknesses of each approach.

This leads us to identify the following research question: "*What are the similarities and differences between embedded*

case study and multi-case study approaches in industry?" Further we can identify two subsidiary research questions: "*What are the benefits and challenges of embedded and multi-case study approaches?*" and "*Under what circumstances should researchers select embedded and multi-case studies?*"

To answer these research questions, we undertake a cross-case analysis of two previously published embedded and multi-case studies performed by the authors.

This paper is structured as follows: The next section presents previous research on empirical software engineering research methods, followed by a description of our research methods in Section 3. In Section 4, we describe our empirical findings, which is followed by a discussion in Section 5. Finally, we provide conclusions and possible future directions in Section 6.

2 EMPIRICAL STUDIES IN INDUSTRY

A range of methods for conducting research have been identified, including: experiments, surveys, archival analysis and history [36]. Experiments rely on manipulating variables in order to establish causal relationships. Surveys seek to observe the size and direction of relationships between variables [9]. Archival analysis studies provide secondary documentary evidence, rather than primary sources [20]. Historical research does not directly address contemporary issues.

These research methods can be employed to fulfil several purposes: exploratory, descriptive, explanatory or improving [29]. Exploratory research seeks new insights, generates new hypotheses and ideas. Descriptive research articulates the current status of some phenomenon. Explanatory research usually seeks to identify causal factors to explain a situation. Improving research tries to refine or optimise some aspect of the studied phenomenon.

The purpose of a survey is to produce quantitative or numerical descriptions of some aspects of the study population [21]. Information is usually collected by asking questions with responses comprising the data to be analysed. Survey information is usually collected from only a fraction of the population, that is a sample, rather than from every member of the population. Surveys have been criticised for being prone to sampling bias and limited geographical scope [34].

Experimental approaches have long been advocated in computer science [37]. Experiments and quasi-experiments deliberately introduce some intervention in order to observe its effect [9]. In its pure form, the application of interventions are randomised in experiments. However, in quasi-experiments, treatments are not assigned randomly.

Qualitative research methods emphasize words, sounds and images rather than numbers and quantitative measurement in data collection and analysis [26]. In the context of software engineering, there can appear to be a disconnect between research conducted and industry needs [5, 7]. Software engineering research is complex due the intersection of issues around human behaviour during software development as well as human and computer capabilities [33]. Given the complexity and variability of industry settings some researchers call for more 'context specific' research that speaks directly to industry needs and does not set out to generalise results [7].

2.1 Case Studies in Industry

Empirical studies in field settings are conducted in order to explore software development processes and practices as perceived and understood by those involved in performing such work [23].

Hence, the focus of this research is on the case study method, which has been defined as:

> an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident [36].

In order to develop a triangulated and in-depth understanding in a contextual research setting, case studies often involve multiple data sources, including observation, interviews, documents and archival records [14].

Case studies tend to be either explanatory or exploratory in nature. Explanatory case studies often address "how" and "why" research questions [36]. Exploratory case studies, in contrast, are often used to conduct an initial exploration intended to generate new insights or propositions [14]. It has been suggested that case studies in software engineering are exploratory because there is still a need to gather "basic knowledge about the human factors surrounding software development and maintenance" [23, pg. 314].

Single case studies are typically used to inform on typical experiences and provide a holistic, in-depth analysis of one setting, characterized by production of the rich and detailed descriptions[36].

A multiple case study design, in contrast, may potentially sacrifice some level of descriptive richness in each case, in order to make comparisons across several settings [13]. It is argued that multiple case designs are needed for creating a generalisable theory under the replication logic of positivist case research [15].

2.2 Case Study Trustworthiness

There is controversy about validity and reliability criteria for research conducted in real-world settings [29]. On the one hand research in industrial settings should aspire to the same standards of rigour as experimental and quasi-experimental methods. On the other hand, it is not typically possible to manipulate independent variables in order to observe the impact on dependent variables in industrial contexts.

Following the approach of Lincoln and Guba [24, Chapter 11], in this study, we adopt four criteria for exploring the trustworthiness of naturalistic research findings:

- Credibility,
- Transferability,
- Dependability, and
- Confirmability.

These criteria attempt to address questions of validity and reliability in real-world research where experimental variables are not easily manipulated to establish causal relationships [29].

2.2.1 Credibility. The first research quality criterion to be assessed relates to the "truthfulness" of the research. A positivist paradigm recognises a single external reality which researchers seek to uncover. However, naturalistic researchers like Lincoln and Gruba [24] consider the realities created by groups, communities and teams of people to be socially constructed. Hence, they argue [24, Chapter 11] "truthfulness" is achieved when the research is carried out in such a way that the findings are found "credible" by researchers and study participants.

2.2.2 Transferability. The second criterion addresses the applicability of research from one group of study participants to another, where results from one context are applicable to another. A positivist might assume that there is some universal truths, regardless of time and context, out there in nature waiting to be discovered. But, many recognise that a group of study participants work within, and are guided by, a particular set of circumstances. Hence, in order to understand the likely application of research from one study group to another, we need to understand the circumstances affecting that group. Transferability refers to the expectation that results from one context are applicable to another.

2.2.3 Dependability. The third criterion relates to the consistency or repeatability of the research. Conventional studies are considered reliable if replication confirms the results. But replication assumes that the precise circumstances faced by a particular group, community or team of people can be reproduced exactly.

2.2.4 Confirmability. Confirmable research is neutral or objective in terms of researcher interaction with the study context. An independent observer should expect to reach similar conclusions in confirmable research.

2.3 Summary

Leonard-Barton [22] compares the single case with the multiple case study, asserting that the efficiency of a single-case, longitudinal study is low, with a danger of data overload, and a lot of unusable data. Objectivity is threatened by researchers becoming too deeply involved, and developing unconscious biases. Pattern recognition tends to be microscopic in terms of examination of highly context specific processes.

Whereas, there are some benefits to the multiple-site method, where efficiency is relatively high, with a focused data gathering effort, and pattern recognition is likely to be good in terms of revealing patterns in the process (regardless of context). However there is still a threat to objectivity, with a danger of unconsciously accepting respondent bias, since the researcher is more detached than in the single embedded longitudinal case study [22].

Table 1: Comparison of multi-case and single-case, longitudinal studies (adapted from Leonard-Barton [22]).

Activity	Single-case, longitudinal	Multi-case
	Data Gathering	
Efficiency	Low	High
Objectivity	Lower confirmability (possible researcher bias)	Higher confirmability (but possible respondent bias)
Pattern Recognition	Detailed	Overall patterns
	Validity	
External	Low transferability	Higher transferability
Internal	High credibility, esp. cause & effect	Lower credibility, esp. cause & effect
Construct	Can test sensitivity to time	Can test stability across situations

When it comes to establishing validity, again we see several benefits to the multiple case study, that has relatively high generalizability as it views a variety of situations, where we could not argue convincingly that observations made in any single case study are generalizable [22].

There are gains to be made for internal validity in the single case, with a good opportunity to establish cause and effect in a longitudinal study. This is clearly lower in the multiple case study, where confusion can exist in cause and effect across the many settings, that are only visited at one point in time [22].

When it comes to construct validity, both types of case have different strengths. In a single-case, longitudinal study, the sensitivity of construct measures (context and changes) can be taken into account over the passage of time. Whereas the multiple case study offers a different form of stability, where the construct can be assessed across situations [22].

Table 1 summarizes Leonard-Barton's comparison, using the terminology of Section 2.2.

Similar to Leonard-Barton [22], we focus on research methodology rather than the research topic under investigation (Scrum activities performed). The motivation for conducting the study on scrum across various companies and contexts is secondary to the purpose of this research, which is to describe how the blending of two methodologies resulted in various synergies, and unforeseen benefits and pitfalls[22].

3 METHODS

In order to discover the contrasting features of case studies methods, a mixed method approach has been adopted comprising two case studies and a cross-case analysis. The first phase (Case Study A) comprised a multiple case study of 46 practitioners from nine multinational companies; employing a Glaserian grounded theory analysis of documentary sources, practice observations and interview transcripts.

In addition to Case Study A, during our second phase, we conducted a 14 month longitudinal embedded case study (Case Study B) in a medium sized software development company.

Finally, we employed a cross-case analysis to triangulate our findings.

3.1 Case Study A

The main source of data in this case study was a series of semi-structured face-to-face research interviews conducted with practitioners supplemented by some use of documentary evidence and workplace observation.

3.1.1 Research Sites. Research sites were selected from a set of multinational enterprises engaged in offshore or outsourced software development using a combination sampling approach comprising snowball and intensity sampling.

Snowball sampling used a network of former co-workers and other professional contacts to provide access to study participants, who then provided access to development teams in different companies [28, p. 237], [27, p. 37]. This initial, exploratory phase of the study, enabled data collection from a broad range of companies.

Intensity sampling was used to obtain a greater richness and depth in the study, by accessing a larger number of interview participants with different responsibilities in the same company or software development programme ([28] pp. 234). Intensity sampling enabled triangulated perspectives from developers, quality assurance testers (QAs), project managers, development programme managers and corporate-level executives

3.1.2 Data Collection. Case Study A made some use of secondary data in the form of corporate process guidelines, project and development programme documentation, and technical reports or white papers. Some of the participating companies also made commercially confidential details of corporate agile practices, roles, policies and recommendations. However, commercially confidential documentation was not obtained from all of the companies, due to its commercial sensitivity.

The main source of data in Case Study A were interviews conducted, between January 2010 and March 2014, using an open-ended semi-structured interview guide and were recorded and subsequently transcribed, as shown in Table 2.

The semi-structured interview guide approach was employed to allow some flexibility to adapt the interview questions to participants with different roles or responsibilities [8]. The open-ended approach employed probing questions to focus in more detail on issues raised by participants (and not included in the interview guide) during interviews.

3.1.3 Data Analysis. The data analysis approach employed in Case Study A involved open coding, constant comparison, memo writing and theoretical saturation advocated in a Glaserian grounded theory approach [17].

Open coding, in this research, involves assigning a short word or phrase to symbolically represent the broader meaning of an action or item in the data [31]. The data analysis software tool NVivo was used to record and formalise the coding process [25].

Table 2: Case Study A– Demographic Information (n=46)

Company	*Company Sector*	*Participant Roles*
A, India	IT Service Provider	Programme Manager, Senior Project Manager, Team Member
B, India	Internet	Engineering Manager, Product Manager
C, India	Software Service Provider	Development Manager
D, India (offshore provider to E)	Software Service Provider	Project Manager, Product Owner, Scrum Master (3), QA Lead, Team Member
E, UK, England	Enterprise CRM	Programme Manager, Project Manager, Director of Engineering
F, India	Industrial Products	Scrum Master
G, India	IT Service Provider	Engagement Manager
H, India	IT Service Provider	Chief Technology Officer, Corporate Lead Architect, General Manager Human Resources, Delivery/Programme Manager (3), Project/Senior Project Manager (3), Scrum Master (2), Technical Analyst, Consultant/Specialist (6), Team Member (9), Business Analyst
I, UK, Scotland	Customer Relationship Management	Chief Operating Officer

Constant comparison is used to compare incidents that apply to each category, integrate categories and their properties, delimit the theory, and write the grounded theory [18, pp.105]. Items in the data are compared within the same organisation or project team and with outside organisations and teams. During constant comparison analysis it is necessary to iterate back and forth between data collection and analysis.

Memo writing was used to clarify, refine and sharpen categories, which evolved as new transcript data is added [17, Chapter 12]. Each memo comprised a short, often informal, essay on each topic including selected quotations to provide illustrative primary evidence.

Theoretical saturation occurs as the study evolves and the richness and detail of the analysis is enhanced by the increasing number of study participants. Gradually, each new participant provides evidence that is consistent with the categories already identified and has less impact on the categorisation.

3.2 Case Study B

Case Study B was a moderate participant-observer study focusing on a development team from a medium-sized Irish-based software company that develops practice and lab management software for the optical industry. PracMed employs approximately seventy staff members in its software development organization, including support and management staff. PracMed's annual sales approach €20 million, from customers across the British Isles, continental Europe, Scandinavia, North America, and China.

Case Study B focused on TeamA, whose responsibility is to tailor the company's product for a large customer in North America; the findings were validated by project management team members from PracMed's development organization (see Table 3). The members of TeamA are distributed over four countries on two continents, with up to eight hours

difference in timezones between locations. They are using Scrum to develop their software, with two weekly sprints.

Table 3: Case 2– Demographic Information (n=9)

PracMed	*Participant Role*	*# participants*
Team A	Product Owner	2
	Software Developer	5
	Quality Assurance	1
	Product Manager/Scrum Master	1

3.2.1 Data Collection. The observations of TeamA in this study were performed from January, 2016 to March, 2017. Specifically, one of the authors observed approximately 200 of TeamA's Scrum ceremonies, including daily standups, sprint planning, backlog grooming, and sprint retrospectives. Due to team members being distributed across Europe and North America, the observations were made via video conference for each ceremony. The same author also conducted semi-structured interviews of each member of TeamA, which were recorded and transcribed. The interviews took approximately one hour and followed an interview protocol available from [6].

The observer also made contemporaneous hand-written notes during both the ceremony observations and interviews. Finally, the interviewer summarized the interviews using a mind-map, and presented the result to five interviewees in an online workshop to validate the insights gained from the interviews.

3.2.2 Data Analysis. Two researchers were involved in the data analysis, using a deductive approach, First, Researcher 1 extracted fragments of text from interview transcripts that related to the research question. Then, Researcher 2 reviewed these fragments, to identify any that seemed out of scope of the research question. This process was repeated to create a set of relevant text fragments.

Next, Researcher 1 defined a set of codes and coded the text fragments accordingly. Then, Researcher 2 independently coded the entire set of text fragments, using the same codes and associated definitions, The second researcher also noted when no codes seemed to match a fragment, and when a fragment appeared to be out of scope with the research question.

Finally, the two researchers discussed disagreements, resulting in some codes being removed or merged, and some new ones being introduced.

3.3 Cross Case Analysis

The case study approach is well established in software engineering [30]. A cross case (or cross site) analysis is used to explore similarities and differences between cases [27]. We can use multiple cases to establish the range of generality and conditions of applicability of each approach [18].

Some authors have focused on synthesis of data from multiple case studies [11, 12], three main approaches have been identified: narrative synthesis, thematic synthesis and the cross case analysis [10]. Narrative synthesis focuses on using words and textual data to "tell a story" based on the case study data. Thematic synthesis is a method for identifying, analysing and reporting patterns within case study data. Cross case analysis facilitates the comparison of commonalities and difference in the events, activities, and processes considered important for each case [19].

The approach we have employed for this research is a cross case analysis from a methodological perspective [27]. We have purposively selected Case Study A and Case Study B in order to contrast features of the methodology.

4 RESULTS

Our main purpose here is not to present original findings from either case study itself, but rather to focus on the cross case analysis from a methodological perspective. However, establish credibility for the methods used, previously published results from each case study are summarised.

4.1 Case Study A

The multi-case study approach has been used to explore the activities performed by product owners on large-scale offshore software development programmes [1, 2]. That research found product owners organised into hierarchical teams. For the large scale teams in that study, governance and risk assessments were conducted in addition to more familiar agile product ownership activities.

The multi-case study approach has also been used to explore the artefacts produced by cooperating software teams on large-scale development programmes [3]. In addition to producing working software, teams also produced detailed release plans in order to manage dependencies between cooperating agile teams.

4.2 Case Study B

The embedded case study approach has been used to explore the activities of the Product Owner role in Scrum [4]. In particular, it was found that Product Owners perform a wide variety of activities, that can lead to stress and potential conflicts of interest, resulting in negative impact on team performance.

4.3 Cross Case Analysis

The research questions from the case studies we have investigated here are shown in Table 4.

The main features of the case studies we have investigated are summarised in Table 5. These features of each case study were selected to investigate each research question.

We found that embedded case studies offer the opportunity to observe events (such as agile ceremonies: stand-up, retrospectives, etc.) repeatedly during a study period. This allows researchers to assess the frequency of best practices and observe teams under the varying project conditions they face over time. However, in order to gain the benefit of witnessing events repeatedly in the study company, researchers must expend a considerable time period.

Table 4: Research Questions Used in Case Studies

Research Question	*Study*
"How do practitioners describe the tailoring of agile method roles and practices in large scale offshore enterprise software development programmes?"	[2]
"How do practitioners describe enhancement and expansion of functions within the product owner role, to meet the needs of large scale offshore enterprise software development programmes?"	[2]
"How do the artefacts map to software development processes used in large-scale offshore software development programmes?"	[3]
"How do these practitioner descriptions contribute to our understanding of artefacts in agile method tailoring in large-scale offshore software development programmes?"	[3]
"What activities do Product Owners perform according to the empirical literature?"	[4]
"What activities do Product Owners perform in practice?"	[4]

Table 5: Key Features of Selected Case Studies

Criterion	*Case Study A*	*Case Study B*
Duration	one day to 2 weeks	15 months
Number of Subjects	46	9
Number of Companies	9	1
Incident Observation	Once	Repeated
Time Required to Conduct Data Collection	Modest	Extended

Table 6: Strengths and Weaknesses of Case Study Approaches

Method	*Opportunities*	*Challenges*
Embedded Case Study	Repetition of events	Long time commitment
Multi-case Study	Triangulation across companies	Challenges to get access

In contrast, the multi-case approach provides the opportunity to triangulate findings across companies. This creates the opportunity to broaden the researcher's repertoire of observed events and practices. However, it remains a challenge to get access to companies, that tend to be secretive about their practices in order to protect commercial advantage. These opportunities and challenges are summarised in Table 6.

Table 7: Cross Case Analysis

Method	*Strengths*	*Weaknesses*
Embedded Case Study	Confirmability	Transferability
Multi-case Study	Transferability	Confirmability

5 DISCUSSION

This research has investigated the question: "*What are the similarities and differences between embedded case study and multi-case study approaches in industry?*" We contrasted two case study approaches previously, and separately, conducted by the authors. In Case Study A a multi-case study approach was used to investigate practitioner perceptions of product ownership and development artefacts across nine international companies engaged in software development using agile methods.

Case Study B, in contrast, used an embedded case study approach to explore practitioner perceptions of activities within the scrum master role.

We follow Lincoln and Guba [24] and use quality indicators for empirical qualitative research: credibility, transferability, dependability and confirmability. We observe that Credibility is enhanced when research questions are appropriate for the method selected. Dependability is enhanced when data is collected with care and diligence as well as being subjected to rigorous analysis.

Drawing on the findings from our cross-case analysis, to answer our second research question "*What are the benefits and challenges of embedded and multi-case study approaches?*" we found that transferability is enhanced by using the multi-case approach whereas confirmability is enhanced by using the embedded case study approach, as shown in Table 7.

5.1 Implications for Theory

The case study method encapsulates a broad range of research styles and approaches. Case studies can range from interpretive, constructionist and ethnographic qualitative research to positivist experimental quantitative studies. Case studies in industry can be executed using a range of approaches.

As shown in Table 1, we can identify strengths and weaknesses of each of the case study styles considered in this research. There are trade-offs to be considered when selecting a case study approach.

Research questions and units of analysis must appropriately correspond to selected methods. Embedded cases lend themselves to detailed investigations of the project particularities, with potentially high levels of confirmability but lower efficiency and higher risk or researcher bias. Whereas broad short-duration studies lend themselves to segmentation of research sites within a specific industry sector, business size range or project style and hence can favour transferability.

Table 8: Recommended Research Questions

Case Study	*Research Questions*
Embedded Case Study	How does the way [X] is performed evolve (over time)? Why does [X] vary over time?
Multi-case Study	How is [X] performed in different (units of analysis)? Why does [X] vary in different (units of analysis)?

5.2 Implications for Research Practice

Gaining access to perform studies in industry is, in our experience, challenging. Researchers are not always in a position to be selective about the type of access they have been granted. However, this research can inform the types of request for access researchers make. If developing a partnership with a specific company, then seeking longitudinal embedded access is desirable.

When an opportunity arises for access to industry, then researchers need to derive Research Questions that correspond to the type of access granted. Conversely, if negotiating industry access, ensure that the access requested corresponds to the planned research questions.

While it has been argued that all software engineering case study research is exploratory [23, pg. 314], the enhanced confirmability of the embedded case study may make this technique suitable for explanatory research. The longitudinal nature of the embedded case study could offer an opportunity for hypothesis testing. The multi-case study approach, if combined with shorter durations in each setting, is confined to exploratory research.

To answer our third research question: "*Under what circumstances should researchers select embedded and multi-case studies?*" we can now used our findings to develop illustrative case study research questions, as shown in Table 8. Drawing on the enhanced confirmability of the longitudinal embedded case study approach, research questions focus on 'how' and 'why' phenomena evolve over time. In contrast, drawing on the enhanced transferability of the multi-case study approach, research questions focus on comparative studies within different units of analysis (such as different companies with a similar context).

5.3 Limitations

This research draws methodological conclusions from two previously published empirical case studies, conducted by the authors, in industry settings. The cross case analysis is used to identify similarities and differences between the approaches taken in the case studies.

The data collection in Case Study A was performed in UK and India; while for Case Study B, data collection was performed in Europe and North America. The companies in Case Study A tended to be larger multi-national enterprises, while Case Study B focused on a single medium-sized Irish company.

The authors addressed the risk of bias in this study, by each advocating their preferred case study approach while also acting as a critical friend for the other case study style. In this way, the overall research team was able to hold itself to account when appearing to favour one or other approach. We are also satisfied that our findings for empirical studies in software engineering support those proposed within the organisational studies domain [22].

6 CONCLUSIONS

In this research we have compared two approaches to conducting case studies in industry: the participant observation-based embedded case study and a multi-case study involving practitioner interviews.

With access to sufficient resources, it would be attractive to undertake longitudinal studies across multiple cases. However, such an approach requires multiple researchers (embedded in multiple units of analysis) over a protracted period, requiring an abundance of resources not available in our experience.

Case Study A used a combination sampling approach comprising snowball and intensity sampling to investigate nine international companies engaged in geographically distributed software development predominantly using agile methods.

Case Study B, in contrast, was a participant observation embedded case study lasting 13 months in a mid-sized Irish company producing medical practice management software using a geographically distributed development team.

A methodological cross case analysis was employed to compare commonalities and differences between Case Study A and Case Study B. The cross case analysis contrasted features of the two case study approaches.

For the analysis in which we compared the two case study approaches, we selected four naturalistic qualitative research quality indicators: credibility, transferability, dependability and confirmability [24]. Using the findings from our cross-case analysis, we found that whereas the multiple-case approach enhances transferability, the longitudinal, embedded case study approach enhances confirmability. Conversely, ensuring transferability is a challenge for the longitudinal, embedded case study approach and confirmability is a challenge for the multi-case approach.

We found that the multiple-case and embedded case study methods are suitable for exploratory research; both can be used to develop new hypotheses regarding the phenomena under investigation. However, we argue that the longitudinal case study approach might also be suitable for explanatory research in which hypothesis testing is performed. However, research objectivity can be compromised by long-term interaction with study participants. This can be alleviated to a certain extent through the involvement of researchers (not directly involved in the data collection) testing the reliability of the findings.

Further, we have identified research questions particularly suited to each approach. Research questions targeting change and evolution of the phenomenon under investigation are suited to the longitudinal embedded case study approach. Whereas research questions exploring variations of the phenomenon under investigation within similar contexts are better investigated by employing the multiple-case approach.

Future research will include extending the comparison of the two case study approaches to include the positivist criteria: internal validity, external validity, reliability and objectivity. We also intend to consider mixed method approaches, where the two case study types are supported with cross case analysis to offer methodological triangulation.

7 ACKNOWLEDGMENTS

The authors are grateful to participants in Case Study A. Thanks also go to IIIT-B and Company H who provided hospitality during several research visits. The research benefited in part from travel funding from the UK Deputy High Commission, Bangalore; Science and Innovation Network.

We also thank the members of TeamA and for their generous and thoughtful collaboration on this study, and PracMed, for allowing us to study their software development efforts. This work was supported, in part, by Science Foundation Ireland grants 10/CE/I1855 and 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie).

REFERENCES

[1] J. M. Bass. 2013. Agile Method Tailoring in Distributed Enterprises: Product Owner Teams. In *Proc. IEEE 8th Int. Conf. on Global Software Engineering*. IEEE, Bari, Italy, 154–163. https://doi.org/10.1109/ICGSE.2013.27

[2] J. M. Bass. 2015. How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering* 20, 6 (2015), 1525 – 1557.

[3] J. M. Bass. 2016. Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology* 75 (July 2016), 1–16. https://doi.org/10.1016/j.infsof.2016.03.001

[4] Julian M. Bass, Sarah Beecham, Mohammad Abdur Razzak, Clodagh Nic Canna, and John Noll. 2018. An Empirical Study of the Product Owner Role in Scrum. In *Proceedings of International Conference on Software Engineering (ICSE 18)*. ACM.

[5] S. Beecham, P. O' Leary, S. Baker, I. Richardson, and J. Noll. 2014. Making Software Engineering Research Relevant. *Computer* 47, 4 (2014), 80–83. https://doi.org/10.1109/MC.2014.92

[6] Sarah Beecham, John Noll, and Mohammad Abdur Razzak. 2017. Lean Global Project Interview Protocol. *Available at http://bit.ly/2nPxaXH*. (2017).

[7] L. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh. 2017. The Case for Context-Driven Software Engineering Research: Generalizability Is Overrated. *IEEE Software* 34, 5 (2017), 72–75. https://doi.org/10.1109/MS.2017.3571562

[8] S. Brinkmann and S. Kvale. 2014. *InterViews: Learning the Craft of Qualitative Research Interviewing* (3 edition ed.). Sage Publications, Inc, Thousand Oaks, CA, USA.

[9] T. D. Cook, W. R. Shadish, and D. T. Campbell. 2001. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference* (2nd ed.). Wadsworth Publishing, Boston.

[10] D. S. Cruzes, T. Dybå, P. Runeson, and M. Höst. 2015. Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering* 20, 6 (Dec. 2015), 1634–1665. https://doi.org/10.1007/s10664-014-9326-8

[11] D. S. Cruzes and T. Dybå. 2010. Synthesizing Evidence in Software Engineering Research. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. ACM, New York, NY, USA, 1:1–1:10. https://doi.org/10.1145/1852786.1852788

[12] D. S. Cruzes and T. Dybå. 2011. Research synthesis in software engineering: A tertiary study. *Information and Software Technology* 53, 5 (May 2011), 440–455. https://doi.org/10.1016/j.infsof.2011.01.004

[13] B. Doolin. 1996. Alternative Views of Case Research in Information Systems. *Australasian Journal of Information Systems* 3, 2 (1996). https://doi.org/10.3127/ajis.v3i2.383

[14] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. 2008. Selecting Empirical Methods for Software Engineering Research. In *Guide to Advanced Empirical Software Engineering*. Springer, London, 285–311. https://link.springer.com/chapter/10.1007/978-1-84800-044-5_11 DOI: 10.1007/978-1-84800-044-5_11.

[15] Kathleen M. Eisenhardt. 1989. Building Theories from Case Study Research. *The Academy of Management Review* 14, 4 (1989), 532–550. https://doi.org/10.2307/258557

[16] Bent Flyvbjerg. 2006. Five misunderstandings about case-study research. *Qualitative inquiry* 12, 2 (2006), 219–245.

[17] B. G. Glaser. 1998. *Doing Grounded Theory: Issues and Discussions*. Sociology Press, Mill Valley, USA.

[18] B. G. Glaser and A. L. Strauss. 1967. *Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine, Chicago, IL., USA.

[19] S. Khan and R. VanWynsberghe. 2008. Cultivating the Under-Mined: Cross-Case Analysis as Knowledge Mobilization. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 9, 1 (Jan. 2008). https://doi.org/10.17169/fqs-9.1.334

[20] B. Kitchenham. 2004. *Procedures for undertaking systematic reviews*. Technical Report TR/SE-0401. Department of Computer Science, Keele University and National ICT, Australia Ltd, Joint Technical Report. http://www.scm.keele.ac.uk/ease/sreview.doc

[21] B. A. Kitchenham and S. L. Pfleeger. 2008. Personal Opinion Surveys. In *Guide to Advanced Empirical Software Engineering*. Springer, London, 63–92. https://link.springer.com/chapter/10.1007/978-1-84800-044-5_3 DOI: 10.1007/978-1-84800-044-5_3.

[22] Dorothy Leonard-Barton. 1990. A Dual Methodology for Case Studies: Synergistic Use of a Longitudinal Single Site with Replicated Multiple Sites. *Organization Science* 1, 3 (1990), 248–266.

[23] T. C. Lethbridge, S. E. Sim, and J. Singer. 2005. Studying Software Engineers: Data Collection Techniques for Software Field Studies. *Empirical Software Engineering* 10, 3 (July 2005), 311–341. https://doi.org/10.1007/s10664-005-1290-x

[24] Y. S. Lincoln and E. G. Guba. 1985. *Naturalistic Inquiry* (1st ed.). SAGE Publications, Inc, Beverly Hills, Calif.

[25] QSR International Pty Ltd. [n. d.]. NVivo 11 for Windows. http://www.qsrinternational.com/nvivo/nvivo-products/nvivo11-for-windows. ([n. d.]).

[26] L. McLeod, S. G. MacDonell, and B. Doolin. 2011. Qualitative research on software development: a longitudinal case study methodology. *Empirical Software Engineering* 16, 4 (Aug. 2011), 430–459. https://doi.org/10.1007/s10664-010-9153-5

[27] M. B. Miles and A. M. Huberman. 1994. *Qualitative Data Analysis: An Expanded Sourcebook* (2nd ed.). Sage Publications, Inc, Thousand Oaks, CA, USA.

[28] M. Q. Patton. 2002. *Qualitative Research & Evaluation Methods* (3rd ed.). Sage Publications, Inc, Thousand Oaks, CA, USA.

[29] C. Robson. 2011. *Real World Research* (3rd ed.). John Wiley and Sons Ltd., Chichester, UK.

[30] Per Runeson, Martin H ost, Austen Rainer, and Bj orn Regnell. 2012. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, Hoboken, NJ, USA.

[31] J. Saldana. 2015. *The Coding Manual for Qualitative Researchers Third Edition* (3 edition ed.). Sage Publications Ltd, Los Angeles ; London.

[32] Roland W. Scholz and Olaf Tietje. 2011. *Embedded Case Study Methods*. Sage Publications Inc., Thousand Oaks, CA, USA.

[33] C. B. Seaman. 1999. Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering* 25, 4 (1999), 557–572. https://doi.org/10.1109/32.799955

[34] M. Torchiano and F. Ricca. 2013. Six reasons for rejecting an industrial survey paper. In *2013 1st International Workshop on Conducting Empirical Studies in Industry (CESI)*. 21–26. https://doi.org/10.1109/CESI.2013.6618465

[35] J. M. Verner, J. Sampson, V. Tosic, N. A. A. Bakar, and B. A. Kitchenham. 2009. Guidelines for industrially-based multiple case studies in software engineering. In *2009 Third International Conference on Research Challenges in Information Science*. 313–324. https://doi.org/10.1109/RCIS.2009.5089295

[36] R. K. Yin. 2009. *Case Study Research: Design and Methods* (4th ed.). Sage Publications, Inc, Thousand Oaks, CA, USA.

[37] M. V. Zelkowitz and D. R. Wallace. 1998. Experimental models for validating technology. *Computer* 31, 5 (May 1998), 23–31. https://doi.org/10.1109/2.675630