Managing Research Based Software Product Development in Sri Lankan Universities

A thesis submitted for the Degree of Master of Philosophy

D.M.P. Dias
University of Colombo School of Computing
2016

Declaration

The Thesis is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge it does not contain any material published or written by another person, except as acknowledge in the text.

Author's name ……………………………….. Date ………………

Signature ……………………………………….

This is to certify that this thesis is based on the work of Miss ……………………………… under our supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by

Supervisor 1 Name ………………………….. Date ………………...

Signature ………………………..

Supervisor 2 Name …………………………… Date ………………..

Signature …………………………

Acknowledgement

Foremost, I express my gratitude to the University of Colombo School of Computing (UCSC), Sri Lanka for facilitating to conduct this research work. The research group wish to convey sincere thanks to the National Research Council (NRC), Sri Lanka for the research grant. We are thankful to all the participants of this study, for their time and immense help.

Furthermore, immeasurable appreciation and deepest gratitude for the help and support are extended to the following persons who in one way or another have contributed in making this study possible.

Dr. Mrs. Yamaya Ekanayaka, the principal supervisor, subject specialist and qualitative researcher, for her support, advices, guidance, valuable comments, suggestions, and provisions that benefited much in completion of this study. Without her guidance this dissertation would not have been possible.

Prof. N. D. Kodikara, the co-supervisor, who helped throughout the project by sharing his knowledge and experience in software engineering and research project management, and patiently supported in funding related matters.

Dr. Mahen Jayawardena, the former supervisor and the principal investigator of the NRC grant, who initiated this study. His groundwork, ideas, and guidance were immense help in carrying out this research.

Dr. Chamath Keppitiyagama, Dr. Ruvan Weerasinghe, Dr. Shiromi Arunatileka, Dr. Damitha Sandaruwan, and Mr. Viraj Welgama, senior staff members at UCSC, who helped in different ways to make this project successful by sharing their experience and ideas.

Udeni Jayasinghe, Kasuni Weerasinghe, Ravihansa Rajapakse, Samanthi Mahathanthila, Dinusha Thilini, and all the other colleges at UCSC, who encouraged me and supported to carry out this project in many ways.

Last but not least, it gives me a great pleasure to share the value of this research with my beloved parents, this dissertation would not have been possible without them.

Finally, I would like to thank all the readers who would read this dissertation in the future.

Abstract

Research groups of universities around the world have become entrepreneurial during the past few years and by doing so intend to be productive and financially independent entities. The authors of this study identified several research groups in Sri Lanka that could not overcome challenges to become sustainable research and development entities. As successful functioning of such entities lead to the development of a country, investigation of this topic is considered important.

There is little empirical research on managing research based software products within universities which could shed light to enhance the current status within the universities. Therefore, this study is designed to answer three research questions: (1) Why do software development research projects in universities need different management practices from existing software development methodologies? (2) What are the challenges faced by researchers when managing research and software product development within Sri Lankan universities? (3) How do established research groups manage research based software product development within Sri Lankan universities?

Using a qualitative longitudinal study, the authors investigated the management aspects of software development research. The study was conducted using grounded theory approach and case study research methods. Data was collected from members of research groups and industry practitioners in Sri Lanka via semi-structured interviews and open ended questionnaires.

Using the theoretical lens of software engineering, project management, and systems theory, the authors identified the challenges and suggested best management practices to improve the overall research culture in local universities. Challenges occurred due to dynamic teams, lack of committed leadership, uncertain milestones and lack of essential resources. Among the many best practices presented, creating a research culture by research supervisors through actively engaging with the group, community building and revenue generation through small scale industry projects are considered as important.

A framework is developed based on the best practices. It can be used to strategically manage research groups in the universities to achieve long term sustainability and financial independence. Furthermore, findings of this study could be used to create awareness and motivate students to be successful researchers or entrepreneurs. As future research, theoretical aspects of each concept of the framework can be further investigated.

Table of Contents

List of Tables

List of Figures

Acronyms

IS - Information systems

R&D - Research and development

GTA - Grounded theory approach

CSR - Case study research

AR - Action research

DSR - Design science research

FOSS - Free and open source software

ROI - Return on investment

Chapter 1. Introduction

In the late 20th century, with second academic revolution, universities around the world transformed into teaching, research and economic development focused entrepreneurial universities (Gibb 2001; Clark 2001; Etzkowitz 2001; Etzkowitz 2003; Atkinson & Pelfrey 2010). Research transferring and commercialisation activities evolved as a result of this transformation. Although many developed countries have achieved a certain level of entrepreneurial development, some universities in developing countries like Sri Lanka are still struggling to actively participate in research and technological product development activities (Rajapaksa, 2013).

The authors of this study observed, most of the Sri Lankan universities have not yet achieved entrepreneurial transformation. In the area of software development, there were incidents that researchers had tried and failed to implement working products due to various reasons. There are a few well established research groups that have developed and implemented high-end technological solutions. Through that they have achieved a certain level of economic sustainability as individual entities. In this research the authors sought to comprehend best practices used by established research groups to overcome challenges and manage research based software product development in Sri Lankan universities.

1.1. Motivation

Research projects are complex, uncertain, and outcomes are unpredictable in nature (Ernø-kjølhede 2000). When research management itself is difficult, why it is required to implement and deploy products as outcomes of university research projects? According to Clark (2001), nowadays governments as well as general public are interested in more worthy outcomes other than intellectual development, from money allocated for free education and research in a country. Another reason is basic allocation for research funding from state governments is decreasing (Gulbrandsen, Magnus and Smeby 2005). For research projects, initial financial support from industry sponsors is low until some innovative contribution is visible (Bakker 2013). According to Bakker (2013) it is important for universities to make income out of research knowledge, in order to sustain long term in current global economy.

Research commercialisation through new product development in R&D labs have become a trend nowadays (Jessuru et al. 2008; Commonwealth 2007; Commonwealth 2011). Therefore, R&D labs have a better cash flow and expand by accommodating more staff and resources for research. Entrepreneurial universities generate additional income for research through commercialisation of innovative products. From the income they create new employment opportunities, different skilled labour force, and problem solving mechanisms in addition to new knowledge. Likewise, universities around the world are focused on transferring their research knowledge into usable outcomes (Perkmann et al. 2013; Carayannis et al. 1998; Colyvas, Crow, Gelijns, Mazzoleni, Richard R. Nelson, et al. 2002; Tanha et al. 2011; Farsi et al. 2011). Ultimately, production of scientific research contributes to country’s economic and social development.

During past few years, IT industry in Sri Lanka has made a significant growth and ranked among the top 25 destinations globally (SLASSCOM 2014). Export revenue through software development grew from USD 213 million in 2007 to USD 720 million in 2013 (SLASSCOM 2014). Furthermore, SLASSCOM (2014) report states that research and innovative product development is encouraged by both the industry and government in Sri Lanka. According to the report published by the Division of Research and International Cooperation (DRIC) of the University Grants Commission, Sri Lanka, only 7.43% is awarded to engineering and IT faculties from total postgraduate grants awarded in 2013 and 2014. That is not enough compared to grants awarded to other faculties and to fulfil the need of research and innovation expectations by the industry and government in order to win the economic war in academia as well as a country (Sirimanna 2011).

The authors of this research are employed in a Sri Lankan university where software development research collaborations are encouraged through a number of initiatives. This study is motivated by their experiences and observations regarding research projects conducted within Sri Lankan universities. It is witnessed that currently academics are interested in transferring their research findings into workable software solutions. Most of the software products are much needed solutions for critical problems in Sri Lanka, and researchers have developed working systems. As witnessed through preliminary interviews, some of these systems are not implemented due to various reasons. It is understood that there is a lack of products emerge through research activities in Sri Lankan universities. Therefore, it is important to understand and find ways to overcome issues that prevent them from implementation. Practices used for research transferring in other technological fields or developed countries are difficult to apply in this context (Meyer-Krahmer & Schmoch 1998). Hence, this study focuses on investigating best practices used for research based software development in Sri Lankan universities and their impact on successful implementation of products.

1.2. Research problem

The main research problem is identified as, that there is a lack of established research groups in Sri Lankan universities compared to other countries. During preliminary interviews it was witnessed that a considerable amount of projects have terminated or struggling to sustain long term because of poor management practices rather than technical faults in software products. Therefore, the authors have investigated on management challenges of research based software development projects in Sri Lankan universities. The strategy was to identify the best practices employed by established research groups when developing software products based on their research findings. These practices should not eliminate the main purposes of university research; creativity, research, learning, and innovation. To address this scenario the authors came up with three specific questions;

1. Why do software development research projects in universities need different management practices from existing software development methodologies?

Software engineering discipline has a long history of management practices, for examples; Waterfall, Evolutionary prototyping, and Agile methods. During preliminary discussions, the authors understood that it is difficult to manage research and software product development simultaneously in universities, by using those software development methods used in the industry. Therefore, it is decided to identify the key differences, in order to find appropriate management practices.

2. What are the challenges faced by researchers when managing research and software product development within Sri Lankan universities?

Software product development in universities is a new paradigm started with the emergence of entrepreneurial universities. Therefore it is challenging to develop working products within universities, because still research and learning are their main focus. To come up with more relevant management practices for universities, it is needed to understand challenges.

3. How do established research groups manage research based software product development within Sri Lankan universities?

Once key differences and challenges in universities are identified, the authors explored best practices and strategies that are used to manage research and software product development activities effectively within universities. These practices are identified by longitudinally studying well established research groups.

1.3. Significance

It is a timely need for Sri Lanka to improve productive research in universities in order to be competitive with rapidity developing entrepreneurial universities around the world. The authors witnessed several research projects that continued to develop working systems, but had failed to deploy complete solutions for end users to use. Therefore, it is considered important to investigate the obstacles that hinder systems development based on research projects after utilizing a considerable effort and resources. As some research groups have successfully implemented their research products, the authors looked into those groups to know how they overcame the obstacles. Through the study they found specific challenges and best practices that affect managing research based software product development in Sri Lankan universities. Ultimately, this study will help to increase productivity in software development research and overall country's development in the future.

1.4. Context and scope

This study is context specific as the authors investigated software development research projects only within Sri Lankan universities. Several projects from the industry are studied to identify differences in management procedures. The findings could be applied in another developing country to find answers for management problems in research based software development. Some findings might not applicable because of variations in policies, political environment, and research funding procedures.

The scope of the study is limited to management of software development research within universities. The authors have not looked into technological aspects in software product development. Although the study is focused in managing software research, some of the findings may suitable for other research disciplines as well, for examples; physics, medicine, chemistry, or engineering in Sri Lankan universities. Even though this study is conducted within universities, some findings are useful for managing research based software development in other research institutions. The authors invite researchers to mould the suggested best practices carefully, to manage research based innovative product development in other settings.

1.5. Theoretical background

Management of a software development research group includes research management as well as software engineering management. Research project management is problematic than other projects, and only a limited amount of theory available for research management (Ernø-kjølhede 2000). The general systems theory (Von Bertalanffy 1968), which is applicable in management (Johnson, Richard A and Kast, Fremont E and Rosenzweig 1964; Kast & Rosenzweig 1972), information systems development (Gregor 2006), and team organization (Ilgen et al. 2005), is used as the fundamental theory to build this study on. As this study is focused on motivating economically sustainable research groups in Sri Lankan universities; a stable research group, is considered as the unit of analysis. Best possible inputs, and processes used for research based software development in universities are identified through the study.

In literature, previous researchers have discussed about software development methodologies that are used to decide appropriate inputs, and processes. Although they have not proposed a specific methodology for universities, Agile framework (Beck et al., 2001), and free and open source software (FOSS) development (Raymond, 2001) principles are identified as most relevant software engineering methodologies. This study is evolved based on the literature review. Reasons for software project failures and best practices are captured through previous studies. Finally, practices from previous methodologies, which can be inherited to university environments are discussed with the findings of this study.

1.6. Goals and objectives

The ultimate goal of this study is to increase the number of long term sustainable research groups in Sri Lankan universities. These research groups are expected to balance research and economic activities as independent entities. The authors carried out a qualitative longitudinal study with a set of research groups. The thesis comprises best management practices and strategies used to overcome challenges and continue research and economic activities by researchers in Sri Lankan universities. The specific objectives of this study are as follows;

- To do a comprehensive literature review
    - Understand current practices used in university projects
    - Study existing software engineering management methods
    - Elaborate the gap in literature
- Select an appropriate research methodology
    - Understand the nature of the study
    - Select appropriate qualitative research methods for theory building and evaluation
    - Design appropriate data collection strategies and questionnaires
    - Find participants for the study from universities and software companies
    - Develop data analysis and evaluation strategies

- Assess usefulness of current software engineering practices
  - Identify of existing research and software development methods
  - Compare and contrast those factors in university research projects and software industry projects in Sri Lanka
- Identify unique challenges faced and best practices used to overcome those by researchers
  - Find challenges and best management practices by doing recurring data collections
  - Develop findings using grounded theory approach
- Evaluation of the findings using score cards and Case study research
  - Design score cards and the case study
  - Select the most relevant research group to assess
  - Collect data, analyse and document the case study
- Discuss applicability of the findings in other research and innovation focused contexts

1.7. Audience and expected benefits

The findings of this study will be especially useful for academics who would like to initiate research groups and contribute valuable software systems to the nation as outcomes of their research intelligence. Existing research groups that have challenges to sustain long term, can find strategies to overcome the issues, from this dissertation. The results of this study can be used to derive suitable software development methodologies for research groups in entrepreneurial universities in developing countries. These contributions can be used to design university curriculum in a way that motivate students to develop and implement their conceptual research ideas. Sri Lankan researchers from other disciplines can also be benefited from some of the findings to solve their problems in research continuation and product development. Although the authors are novice in qualitative inquiry, IS researchers may find strategies to do empirical studies from the methodology. Last but not least, this study may be useful for companies in the software industry who would like to do research and innovative product development.

1.8. Dissertation structure

The evolution of universities, research projects in universities and software product development practices are discussed as a literature review in Chapter 2. A detailed review of existing research based activities and software development methodologies are discussed in order to illustrate the gap and need for novel practices for research based software development in Sri Lankan universities. Literature review findings are again used in Chapter 4 as a foundation to finalize concepts, categories, and questionnaires along with selective coding during final stages of the grounded theory approach. Chapter 3 explains the nature of this study, research design, and the overall methodology. The grounded theory approach (GTA) is used to develop the study inductively in order to find challenges faced and best practices used by research groups in Sri Lankan universities. Case study research and forced ranking methods and are used to evaluate the findings. Chapter 4 includes the analyses and findings of the GTA. Chapter 5 consists of the case study evaluation report and forced ranking results. Chapter 6 concludes the thesis by summarising contributions, and explaining future directions that evolve through this research.

Chapter 2. Literature Review

Universities were originally formed to meet specialized education needs in military, politics, culture, and religious practices (Perkin 2006). Such institutions were focused on producing high skilled labour through teaching and training activities. With the first academic revolution in the 19th century, universities started to focus on research and innovation as a part of their curriculum (Etzkowitz 2003). During the 20th century, as a result of second academic revolution, entrepreneurial universities emerged and they are now focused on economic development activities, in addition to teaching, and research (Gibb 2001; Gjerding et al. 2006; Atkinson & Pelfrey 2010). Through literature review the authors identified a gap in existing knowledge of managing research based software product development in universities.

The thesis title; "*Managing research based software product development in Sri Lankan universities*," includes management aspects, not only in research projects focused on innovation, learning, and experimentation, but also for software product development, with the idea of research transferring, as commercial products or free and open source products, from Sri Lankan universities. Managing these the two disciplines together is difficult and complex. Figure 2-1 demonstrates this gap using a Venn diagram. Although there is a lack of literature in research management (Ernø-kjølhede 2000), there are many previous studies that explained university research transferring and commercialisation activities. Software development management has numerous methods that evolved throughout the history. However, there is a lack of studies on software development in universities where research is the main focus, and product development is a secondary or a long term goal. It is important as this area has taken attention recently with the emergence of entrepreneurial universities.

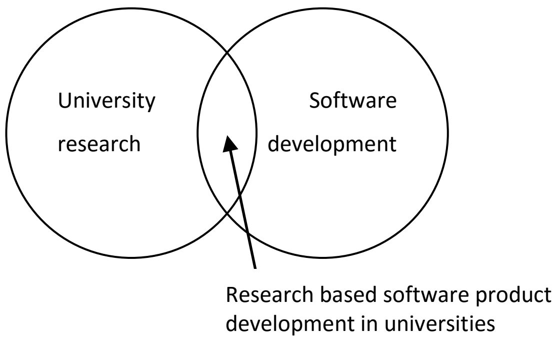

Figure 2-1: Literature gap

A systematic review of existing knowledge in the fields of academic research and software development is carried out. To expand the search, related articles are found from reference lists of most significant publications that appeared first. It is a limitation that the authors had full access only to digital libraries of the Association for Information Systems (AISeL), the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEEXplore). Google Scholar and Science Direct open access articles are also reviewed.

2.1. Managing research projects

A valid research problem, ideas, people, time, funds, and other required resources are considered as basic inputs at the beginning of a research project (Guyette 1983; Hargadon 2003). When a community need becomes a requirement for research and product development, it leads to more desirable outcomes (Guyette 1983). Unity of a set of people from different disciplines and expertise result in innovative ideas, hence research capacity is extended (Hargadon 2003; Cohen & Bailey 1997; Numprasertchai & Igel 2003). Continuous availability of resources including time, funds and required equipment are important for smooth progression of a research (Guyette 1983; Hargadon 2003). Research professors make an effort to build a research culture in universities by providing those inputs (Kroeze et al. 2010; Gulbrandsen, Magnus and Smeby 2005).

Cheng & Yang (2011) have studied about personal capabilities of students in information systems development projects. They state that student teams should have collective motivation, and creativity apart from domain knowledge, methodological knowledge, and implementation knowledge to successfully complete a project. Slater & Mohr (2006) stated: in order to implement a new product it is important to have a set of end users who look forward to use it. As a summary, basic inputs for research projects as explained by previous researchers are given in Table 2-1.

Table 2-1: Basic inputs of a research project

Input	Related practices	References
Requirement for research and product development	If it’s a community need, the research become more beneficial	(Guyette 1983)
People	Cross-functional teams consist of researchers, experts, professors, end users and other communities	(Guyette 1983; Cohen & Bailey 1997; Numprasertchai & Igel 2003; Hargadon 2003)

Personality traits	Domain knowledge, methodological knowledge, implementation knowledge, creativity and collective motivation are important	(Cheng & Yang 2011)
Other resources	Availability of time, funds, and equipment	(Guyette 1983; Hargadon 2003)
End users	A set of potential users for the product to be developed	(Slater & Mohr 2006)

It is understood that constitution of a group of motivated researchers in a university who have required knowledge and resources, when given a compelling research problem to investigate, provides a foundation to a research project. The only process identified from above literature is building of a research culture by professors. The way they do it, challenges face, and how to overcome them are not found through the literature.

2.1.1. Research transferring activities

One of the objectives of an entrepreneurial university is to become a stable entity on its own in this complex and uncertain economy (Clark 2001). According to Clark (2001), universities should find their distinctive ways of achieving this objective by changing their internal activities and external relationships. As a result, research groups started functioning as firm like entities within universities (Colyvas, Crow, Gelijns, Mazzoleni, Richard R. Nelson, et al. 2002; Etzkowitz 2003). Traditional culture of emerging new ideas and innovations through professor-student engagement has not changed, but more functions are added (Etzkowitz 2003). These research centres have identified and solved important research problems and move their outcomes to outside firms and society (Clark 2001). In this new environment professors role has changed from solely supervision, to a team leader, thus sometimes it is like "*running a small business*" (Etzkowitz 2003). Apart from creating a research culture in their areas of expertise, professors have been managing organizational activities virtually full time in these firm like entities. Competition with their peers motivated them to be successful in team building.

As it is understood through literature, software product development based on the research happen in these entities are mainly in two ways. One way is: patenting the innovative idea (Carayannis et al. 1998). That way, if a company develop and distribute the software, the university is given reputation and a share of profits. The other way is: developing the software product within the university by research groups. This way incorporated partnerships with government and industrial firms, networking with practitioners, technology transfer offices (TTO), alumni groups, entrepreneurship, and finally spin-offs (Hashimoto et al. 1997; Siegel et al. 2003; Guerrero & Urbano 2012; Perkmann et al. 2013; Tartari et al. 2014) in order to manage processes and gain collect investments. These activities and partnerships incorporated with research groups are explained further in Table 2-2. According to Numprasertchai & Igel (2003) trust, commitment, and balanced mutual benefits among collaborators are the main success factors in partnerships.

Table 2-2: Partnerships incorporated with research groups

Partnerships	Description	References
Industry partnerships	Partnerships with software industry and other companies. Universities get funds or resources to do research and develop products. Partners get knowledge, innovative ideas or products in return.	(D'Este & Patel 2005; Perkmann et al. 2013; Perkmann 2007)
Government partnerships	Government research focus more towards country's goals. If university projects are funded by the government, outcomes are expected to serve to growth of the entire society.	(Skinner 1917)
Networking with practitioners	Networking with industry practitioners or experts in the field of study brings more ideas and knowledge.	(Levén et al. 2014; Kroeze et al. 2010; Geenhuizen 2011)
Technology transfer offices (TTO)	Administration, contract agreements, marketing and other coordination activities in research and development are handled by these offices.	(Siegel et al. 2003; Perkmann et al. 2013; Tartari et al. 2014)
Alumni groups	Alumni groups support to initiate industry relationships and get donations.	(Clark 2001)
Entrepreneurship	Knowledge based entrepreneurship and spin-off companies are the ultimate outcomes of successful research initiatives.	(Guerrero & Urbano 2012; Gibb 2001)

Industry partnerships, government partnerships, Alumni groups and networking with practitioners bring universities more resources, funds, and innovative ideas to improve their research and development work. TTOs initiate and coordinate all the related administrative activities. Ultimately, some innovative products developed in such research constitutions lead to new companies (e.g., Google).

As this study is focused on software development practices used by university researchers it is important to study the existing software development practices as well to see their applicability in the university context.

2.2. Software product development practices

Evolution of software development (SD) started in 1950's with hardware engineering (Boehm 2006). In 1970's it became formal with the introduction of Waterfall method (Royce 1970). Although Waterfall model addresses all possible activities in a software development lifecycle (SDLC), many problems had been addressed related to the model (Aspray et al. 1996; Highsmith et al. 2002; Laplante & Neill 2004). Most common problem is that it lacks the ability to cope with requirement changes because of the sequential nature (Petersen et al. 2009; Hu et al. 2010). There are many studies about software implementation failures by following sequential approaches (Humphrey 2005; Ogheneovo 2014). Most of the software projects failed because of not meeting goals within budget and schedules (Brooks 1975; Reel 1999). Reel (1999) argued that cost and time overrun due to complexity of managing software projects. Therefore, next generation software developers have come up with dynamic software development practices to overcome those problems (Brooks 1975; Beck & Fowler 2000).

Iterative, incremental, evolutionary development practices emerged to address rapid changes and concurrent processes (Larman & Basili 2003; Boehm 2006). Even in the original Waterfall method, it was mentioned that practitioners have to do the entire process twice in order to get better outcomes (Royce 1970). In 1980's, as suggested in the book of "*the Mythical Man-Month*" by Brooks (1975), evolutionary prototyping had been used to develop large scale software systems in order to reduce the risks of Waterfall (Larman & Basili 2003; Gordon & Bieman 1995). In 2000's Agile methods (Cockburn 2000) came to practice and it is currently used by most of the practitioners in software industry (Parsons & Lange 2007; Dybå & Dingsøyr 2008; Batra et al. 2010).

Conversely, the accidental revolutionary of free and open source software (FOSS) development model introduced a totally different set of software development practices

(Raymond 2001). FOSS development by distributed developers created a free culture which incorporates collective ideas from innovations around the world. The collaborative development approach they use is totally different from existing industry practices. Above methodologies, namely; Waterfall, rapid prototyping, Agile, and open source culture are discussed next with more details about their lifecycles, processes, procedures, and possible conditions to be used.

2.2.1. Waterfall

According to Royce (1970), analysis and coding are considered as the two essential steps in any SD project regardless of size or complexity. Thus, the very first version of Royce's model had only those two phases. For very small and self-made software products this simple model was ideal (Royce 1970). However, large software systems failed by trying to implement using this two-step method (Brooks 1975). Therefore, many steps are added to address the requirements of large scale development projects (Petersen et al. 2009). Subsequently, the standard Waterfall software development approach is introduced with seven steps namely: system requirements, software requirements, analysis, program design, coding, testing, and operations (Petersen et al. 2009). The purpose of this breakdown was to divide the development process into manageable chunks. As explained by later researchers when developing a software product according to Waterfall steps, cost of changing is very high if fundamental design or logical faults occur later in the lifecycle (Aspray et al. 1996; Highsmith et al. 2002; Laplante & Neill 2004). To reduce these problems in Waterfall, they have looked into more iterative, incremental and evolutionary models such as rapid prototyping and Agile.

2.2.2. Rapid prototyping

Rapid prototyping is emerged as a technique to enhance interactive communication between a client and software designers (Brooks 1975). Prototypes are considered as trial versions of the software product to be developed. The difficult parts of design, including interfaces to people, to machines, and to other hardware software systems, are designed and tested using prototypes (Brooks 1986). This enabled early feedback from the users and incremental development of the software. An iterative lifecycle of design, modify and review is involved in prototyping. There are two types of prototypes called throw-away and evolutionary. Throw away prototypes consisted of visual interfaces that can be shown to users for early feedback, but without any real functions or programs. Evolutionary prototypes start with interfaces and develop with real functions that can be tested with end users (Gordon & Bieman 1995). These prototypes are developed as full products at the end. Prototypes can also be used for internal discussions by project teams and to present progress to top management.

The advantages of evolutionary prototyping are reduction of testing time and risks as well as insightful testing of small pieces of the software before final product is released. Feedback gathers through prototypes on what to implement and what not to implement are included in the final product. This improves overall productivity, usefulness and quality of the final output. Main drawbacks of waterfall such as late testing, less customer interaction with the system and vast documentation of the design are reduced by prototyping (Carr n.d.). Still there are issues regarding late identification of design and integration faults and huge cost of prototype development in large systems (Houde & Hill 1997). To reduce the faults had with all previous models, Agile framework is founded by a set of practitioners (Beck et al. 2001).

2.2.3. Agile

Iterative, evolutionary and incremental methods, which are founded in 1960's, put the foundation for Agile methods (Larman & Basili 2003). Extreme Programming (XP) is known as the first well-defined Agile software development approach (Abrahamsson et al. 2002). Consequently, Crystal methods (Cockburn 2004), Scrum (Rising et al. 2000) and Adaptive Software Development (Jayawardena & Ekanayake 2010) came into practice. These methods minimized the problems had with previous methodologies and improved continuous communication. The authors of Agile manifesto (Beck et al. 2001) identified a set of similar principles followed in all those methods and documented as a framework. Agile framework is focused on requirements and solutions evolve through self-organizing, cross-functional teams and their collaborations with end users. The framework explains how to mould the software development processes to suit teams of individuals with different talents and skills (Cockburn & Highsmith 2001).

The main concept of Agile is frequent "*releases*" in short development cycles (Schwaber 2004). Customer is the priority, thus Agile teams deliver working versions of the system early in the process and repeatedly throughout the project. They accept requirement changes even at later stages. Agile teams keen to work collaboratively with business people on daily basis. Organizations that practice Agile motivate individual developers by providing a relaxed and flexible environment to work. The teams are self-organizing; hence they adjust the way they work and discuss regularly to be more productive. They believe that face-to-face communication within the group and with customers solve many issues and lead to clear requirements, better designs, and architectures (Highsmith & Cockburn 2001). Agile embraces changes and try to reduce cost for customizations. Agile project's progress is measured based on the working software. Therefore, they continuously pay attention on maintaining the product quality and good design. Meanwhile simplicity and sustainable development is practiced. All the Agile principles are summarized to four main concepts (Beck et al. 2001; Highsmith & Cockburn 2001; Cockburn & Highsmith 2001) as follows;

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

These principals are focused on minimizing the drawbacks of the Waterfall and the prototyping approaches. Requirement elicitation, analysis, prototype, design, develop, test and deployment steps are performed for small elements of the software as a continuous process, not for the whole product at once (Larman & Basili 2003). Hence, Agile reduces the risks and the cost for re-producing an entire system or a prototype. Each Agile method has its own way of prioritizing features, preparing backlog and estimating efforts. Although Agile is more flexible than Waterfall, and prototyping models, still a question is remained if it is flexible enough to practice in research and innovation focused environment like universities because of its fast paced nature with fixed teams and time-boxed meetings.

The basic requirements for research identified in the previous section (Table 2-1) are assessed related to two of the most used Agile methods; XP (Beck 1999) and Scrum (Schwaber 2004) models, in Table 2-3.

Table 2-3: Processes related to identified inputs of an industry project which practices XP or Scrum

Inputs	Processes in a software company
Requirement for research and product development	Addressed by a customer
Idea(s)	Team discussions and customer feedback
People	Team leader, software developers, customer
Personality traits	Experience and qualifications to the job role
Time to the end user	First iteration in less than one month after negotiating the product backlog
Resources	Allocated by the investors based on the estimations done by the project team

Organization structure	Team members, team leader, top management
Other practices	Regular meetings. Higher ROI is the main objective

From Table 2-3 it is understood that Agile software development methods have well defined methods to address the inputs than university practices (Table 2-1) to manage product development. In the next section FOSS development culture and practices are discussed to assess their applicability in a university context.

2.2.4. Free and open source software development model

This section is formulated mainly based on two books; The Cathedral and the Bazaar (Raymond 2001) and Just for fun: The story of an accidental revolutionary (Torvalds & Diamond 2001). It is said that every good software invention starts with a developer's personal interest and curiosity (Raymond 2001). The FOSS culture started with *Hackers* on the first MIT university network, in 1961. Although now the meaning of "*Hacker*" refers to computer vandals, it was originally referred to a set of network university researchers who experiment and develop software products. With emergence of the internet in 1990's the Hacker culture grabbed the attention of all software developers around the world. They used internet as a platform to provide services to masses. More than the initial Hacker culture, FOSS development model founded by Linus Torvalds through Linux development in 1991, became a trend in the field. In the book of "*The Cathedral and the Bazaar*," Raymond (2001) explains two ways of software development: "*Cathedral - the formal way of commercial software development*" and "*Bazaar - the FOSS culture of Linux development.*" Linux developers used a different software development methodology than the standard practices (Torvalds & Diamond 2001).

Apparently, "*Release often and early*" concept is introduced by Linus's Linux before Agile came into practice (Raymond 2001). Raymond (2001) states that it is a miracle, how a set of part-time hackers build a world-class operating system, which challenges the commercial products. In 1996, Raymond carried out the project "*Fetchmail*" in Bazaar style and developed a theory on effective FOSS development. FOSS development principles, preconditions to practice and social context are summarize below, as explained by Raymond.

- FOSS principles

It is proverbed that "*Every good work of software starts by scratching a developer's personal itch*" (Raymond 2001). Raymond's personal experience on his own POP3 mail project helped him to understand these principles of the Linux world. Linux developers work on their personal interest to develop a better piece of software for their own use. First, they look for similar open source solutions that can be modified to start developing what they want. Raymond has re-phased the software development principle: "*Plan to throw one away; you will, anyhow*" from Brook's (1975) book through his own experience on FOSS development. That means if the developers experiment once, second time they know how to do at least few things right. According to Raymond, the failed first attempt is not a waste, but a lesson. It is important to have a right attitude to embrace interesting problems. However, Raymond has noted, if the leader lose interest, it is his responsibility to pass the program to someone with the interest.

A substantial user base is an important factor for any successful product. To Linux developers the user base is even more important for testing, suggest fixes, and debugging the code. They consider their users as co-developers. "*Release early, release often*" is the slogan of the Linux development model. Previous developers believed that this is a bad practice, as early versions consist of a lot of bugs. However, Linus's open development policy has proved that listening to end users is the most effective way of debugging. Larger the user base become, it is faster and easier to find and fix bugs. When someone finds a bug, it is shared among the community and someone else understands and fixes it. If developers appreciate the contribution of beta-testers of a project, they tend to support more. The other good thing of having a large user base is, they give a lot of breakthrough ideas.

After a while, in his project Raymond (2001) understood that his interpretation of the problem is wrong, thus the solution isn't compatible. The moment he found the right question, he realised the right answer with no time. Then he removed all unnecessary components from the system, and kept the simple pure solution. From that he proved that the perfect design is not the complete one but when there is nothing unwanted. Afterwards, he got a neat and innovative design. Also he learned that he should not ignore requests from his users, although they did not pay for the product, without them there is no value to a product.

- Pre-conditions

According to Raymond, he or Linus did not plan the Bazaar culture. They randomly found co-developer communities. Both Linux and Fetch mail are much needed products for software developers and had attractive core designs. In Raymond's words a "*plausible promise*" caught the attention of a community. As identified at the begriming of this chapter, a certain level of skills and knowledge are required to start a project. However, the FOSS model does not need technical skills as much as the Cathedral style, since it is more of a collective effort. Instead the Bazaar coordinator requires b communication skills and a striking personality.

- Social context of FOSS

Hacker communities formed with the invention of internet. FOSS products and culture initiated based on the developers' everyday problems. Finding a problem that is interesting to developer himself is considered important in Fetchmail and Linux projects. Linux model harnesses the brainpower of a large community more than rich coding of expert programmers. Continuous discussions, releases, and appreciation is needed to keep the community interest for a long time. Linux is the first attempt to harnessing the knowledge from around the world, although there were some other networks within the United States before world-wide web is introduced. Linus was the first to take true value of internet, by using his leadership qualities to grab and maintain the interest of a large community of volunteers around the world. FOSS give a big challenge to commercial software products and their development methods, because of free availability, high quality, flexibility and by better serving of genuine needs to end users. Both Linux and Fetchmail communities have formulated a lot of shared documentation on problem solving, design ideas and other knowledge. Such documentation might not be able to formulate in a Cathedral environment even by paying a lot of money.

> "*I think that the cutting edge of open-source software will belong to people who start from individual vision and brilliance, then amplify it through the effective construction of voluntary communities of interest.*"
>
> *- Raymond, 2001*

The great success of FOSS products started with a motivated visionary thinker and his talents to conveying the idea amongst volunteers with similar interests around the world.

2.3. The gap - Managing research based software development in universities

The objective of this research is to understand the software development practices used by researchers in Sri Lankan universities in order to derive best practices to overcome challenges. Through the literature review, firstly, the basic inputs of a research project are identified (Table 2-1). Another significant aspect of university research is the activities related to research transferring (Table 2-2). Those activities affect the research based software development approach in many ways. Secondly, the authors read through history to learn all types of software development practices that have been used since the beginning of software production. Through literature, Agile practices and open source principles are identified as most applicable methodologies among others, for research focused software development.

The author of "*The Psychology of Computer Programming*", Weinberg (2003) stated that intentionally or unintentionally losing control and missing communication is the key to fail many projects. Without proper communication nobody knows what is happening. Thus "*communication*" is considered as a significant input of a software project. The organization structure, roles, and responsibilities of a project team also impact on project's approach (Hargadon 2003). Thus "*organization structure*" is added as an input. "*Partnerships*" is also identified as another input, since research transferring related partnerships are addressed by many previous researchers.

Unlike the industry (e.g., fixed teams, allocated resources and clients), university environment changes over time in all the aspects. Although Linux model has a similar unpredictable setting, their personal motivations and part-time volunteering culture makes it possible to develop large scale software systems. Inputs of universities, Agile and FOSS models that are found through literature review are summarised in Table 2-4. Although there are similarities between practices, the authors could not find any specific set of principles or models for managing research based software development in universities.

Table 2-4: Similarities and differences in Cathedral, FOSS and university research

Input elements of a project	Cathedral or formal practices	FOSS culture	University research based software development
Requirement for research and product development	Addressed by a customer	A personal itch of a programmer	A community need, innovative ideas
Ideas	Team discussions and customer feedback	From developers around the world	Team discussions and expert knowledge
People	Team leader, software developers, customer	A leader, and part-time volunteer developers	Researchers, experts, professors, end users, communities,

			cross-functional teams
Personality traits	Experience and qualifications to the job role	Passionate leadership to attract and keep a large community	Knowledge, motivation and self-interest on solving the problem
Resources	Allocated by the investors based on the estimations done by the project team at the beginning of the project	Reusable codes, free tools, donations and shared resources by volunteers	University resources, or may obtainable through research funds and partnerships (Table 2-2)
Organization structure	Fixed team, a team leader, and management	Equality among community members	Supervisors and students/fellows
Communication	Regular time-bound meetings	Frequent discussions through internet, aligned with product releases	Informal discussions
Goals and objectives	Return on investment, customer satisfaction	Fulfilling their own needs	Research and innovations
Partnerships	Formal agreements with the customer	Volunteering	Informal relationships with users, Research transfer activities
Other practices	Improved communication	Communication is the key	Technology transfer office (TTO)

It is understood, through experience and preliminary interviews, that there is a significant gap between current software development practices and the practices used by university researchers. Therefore, the authors decided to find answers to the research problem by longitudinally studying software development research groups in Sri Lankan universities.

Chapter 3. Research Design and methodology

The research design and methodology section explains the way this study has been planned and carried out: with regard to what decisions made time to time, how and why the data are collected, organized, analysed and presented. As understood through the literature review, the phenomenon of managing research based software development projects in universities have not received significant attention by previous researchers. In this study, the authors particularly studied about challenges and best practices used in a university context for research and software product development. Similar to other longitudinal studies in the field, this phenomenon is also considered as complex, contemporary and the authors had little or no control over events in the context (Day et al. 2009; Sen 2006; Claybaugh & Srite 2009; Alavi et al. 2006). Qualitative research methods based on empirical data that usually use in social sciences have been increasingly used for these types of studies in sciences recently (Sarker et al. 2012; Baskerville & Myers 2004; Brereton et al. 2014).

Before deciding a suitable research methodology for this study, the authors discussed design factors based on Kothari's (2004) template. These factors, with relevance to this study, are explained in Table 3-1.

Table 3-1: Design considerations

Design questions	Answers - design considerations in this study
What?	Software development research groups
Why?	Lack of product implementations and management practices in this context
Where?	Universities
Data?	Differences, challenges and best practices
Data sources?	Experience of project team members, project sites, internet, documents, articles, books
Time?	March 2012 – June 2015

Unlike natural sciences, social sciences do not have laws based on universal certainty. When carefully studied, fairly regular meaningful patterns are seen in social settings (Kothari 2004). A software development research group is considered as such a social phenomenon (Groenewald 2004). Instead of laws, social scientists develop theories based on patterns that make sense. A theory is defined as "*a general and, more or less, comprehensive set of statements or propositions that describe different aspects of some phenomenon*" (Berg 2001, p. 15). Theories are built upon concepts, which can be objects, properties, features of objects, processes or phenomenon. Concepts provide meaningful information to the readers.

Inductive and deductive are the two fundamental levels of social science research (Charmaz 2008). Simply, they are about developing abstract theories and evaluating them to see how they work in reality (Corbin & Strauss 1990). The aim of inductive research is to generate basic theories from observed data (Charmaz 2008). The main goal of deductive research is to test a theory. It also includes refine, improve, or extend a theory (Bhattacherjee 2012). In this study both inductive and deductive approaches are used to generate a framework and then to evaluate it based on qualitative data.

3.1. Possible qualitative research methods

Ethnography, grounded theory, case study research, action research and design science research are used by many previous researchers for empirical inquiry in IS (Sarker et al. 2012; Baskerville & Myers 2004; Brereton et al. 2014). These qualitative research strategies are explained in Table 3-2.

Table 3-2: Possible qualitative research strategies

Qualitative strategy	Description
Ethnography (Willis & Trondman 2002; Hammersley & Atkinson 2007)	Ethnography is about studying people, a community, or a culture. At the beginning researchers shared his or her own life experiences and its contexts as ethnography research. Later researchers have used Ethnography to understand structures and trends of different cultures by living with them for a long period of time (around a year or so). Hence, Ethnography is used to study people's usual behaviour in everyday contexts, but not under conditions created by the researcher.
Grounded theory approach (GTA) (Glaser & Strauss 1967; Charmaz 2008)	GTA is a creative intellectual activity which is used to generate new ideas and concepts. It is capable of developing a theory out of data by using a specific set of guidelines. Data are collected through interviews, field research, relevant documents or existing literature in the area.
Case study research	CSR is focused on studying an individual case or phenomenon

(CSR) (Yin 2014)	(e.g., a group, a person, an event, a particular society) in the aspects of its structure, dynamics, and context. It is used for inductive theory building as well as deductive theory testing. Depending on the study one in-depth case analysis or multiple case analysis is chosen. CSR has the benefit of studying contemporary events that researcher has a little or no control over.
Action research (AR) (Checkland, Peter and Holwell 1997; Baskerville & Myers 2004; Bhattacherjee 2012)	Action research is used to study a process while creating a change. It is a collaborative activity, thus both researchers and participants are actively involved in changing activities. Depending on the specific method, action research cycle consists of different phases such as diagnosing, action planning, action taking, evaluating and learning. It improves the practical relevance of a research. Action research is aimed at theory testing rather than theory building.
Design science research (DSR) (Hevner et al. 2004)	DSR is used to develop technology based IS solutions to business problems. This is practiced by computer science researchers to design artefacts for problems which needed both theoretical and practical implications. DSR can also be used to solve identified organizational problems when designing and implementing IT solutions.

It was difficult to select an appropriate methodology for this study, as all these qualitative methods have their unique and overlapped features (Groenewald 2004). Therefore, before selecting a suitable methodology, nature of the study is discussed in detail.

3.2. Research design

Qualitative inquiry is started with the study's questions. Therefore, the research questions are revisited here;

1. Why do software development research projects in universities need different management practices from existing software development methodologies?
2. What are the challenges faced by researchers when managing research and software product development within Sri Lankan universities?
3. How do established research groups manage research based software product development within Sri Lankan universities?

Berg (2001) has proposed a set of questions to answer in order to make important decisions when doing qualitative research design. Following answers explain how the authors made those decisions in this study's design.

1. What types of information or data are collected?

Data are mainly consisted of research practices and software development practices. The input elements: requirement for research and product development, ideas, people, personality traits, resources, organization structure, communication, goals, partnerships and other practices, that identified through literature (Table 2-4) are considered as key data categories.

2. Through what forms of data collection techniques?

Semi structured interviews, open-ended questionnaires, participant observation, documents, records and other artefacts are used as data collection methods, depending on accessibility and availability of each research group.

3. Where is the research undertaken, and among what group or groups of people?

The study is undertaken in Sri Lankan universities where active research are happening. Those universities are selected based on relevancy, accessibility and availability of software development research groups. Non probability sampling strategies (Coyne 1997) are used since the whole population of research groups in Sri Lanka is difficult to list. However, a list is found from the internet and emails are sent to request for participation. Since the response rate was low, convenience sampling (Altmann 1974) is used to find more practitioners from university research groups and the industry who are attached to software development projects. Then purposeful and theoretical sampling method (Berg 2001) is used to select more appropriate participants.

4. What strategies are used for data triangulation?

Multiple datasets are collected from both universities and industry in order to improve rigor of the study through triangulation (Bryman 2011). Semi structured interviews (Appendix II - Questionnaire to identify challenges and best practices in university projects, page II) are conducted with more than three (3) researchers from each group, including a supervisor and a research fellow. Depending on the level of accessibility, field observations are done and data are collected through documents, records and other artefacts.

5. Undertake the study alone or with the assistant of others? Multiple investigator triangulation?

Apart from the main author, there were three (3) other investigators who assisted and advised throughout this study. They are senior academics from the fields of information systems and software engineering. All of them have had more than ten (10) years of experience in research and supervision of software development research projects in universities. Thus multiple investigator triangulation is practiced from the beginning of this study.

6. What are the theories that this study is framed by (theoretical triangulation)?

To interpret this phenomenon research practices (Etzkowitz 2003; Ernø-kjølhede 2000) and software engineering practices (Pressman 2010; Boehm 2006) are involved. Theories on Agile framework (Wilson & Doz 2011) and FOSS framework (Lakhani & Hippel 2003) are used extensively, among many software development models, to analyse and elaborate new theory from this research.

7. Whether the project is funded? How much will the project cost in time and money?

The project is fully funded by the National research council, Sri Lanka. Predicted budget is around 1,500,000 LKR and the project duration was three (3) years, from January 2013.

As a summary, this study is started as inductive, because it aimed on generating new theories from unstructured data collected in a natural setting. According to Miles et al. (2014) no matter how inductive or unstructured the research is, qualitative researcher starts fieldwork with some kind of a plan. As the authors are new to the field of qualitative research, the design is kept in between well-structured and emergent. This research is neither social science nor scientific exclusively. It is considered as a social science type of a research from a scientific discipline. Thus, after studying the qualitative research methods (Table 3-2); it is noted that case study research or grounded theory approach is more appropriate for this study than others. GTA is cable of building a theory using a set of defined guidelines (Glaser & Strauss 1967; Charmaz 2008).

Ethnography was not a choice, because the authors could not continuously observe and disturb by engaging with the study participants while they are on the job (Willis & Trondman 2002; Hammersley & Atkinson 2007). Action research was not an option as the authors did not try to change or make constraints to the way researchers work (Checkland, Peter and

Holwell 1997; Baskerville & Myers 2004; Bhattacherjee 2012), but the natural setting is studied. Design science research is more focused on creating new and innovative artefacts; rather than studying processes followed by people in a project (Hevner et al. 2004). Hence, a methodology has to be chosen between CSR and GTA.

Commonly, CSR is used to deductively test conceptual frameworks or hypotheses in IS research (Mockus et al. 2000; Godfrey & Tu 2000; El-haddadeh & Ali 2010). Grounded theory is defined as "*an inductive, theory discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data*" (Urquhart, Cathy and Lehmann, Hans and Myers 2010, p. 357). The grounded theory is used for inductive theory building. It is specifically designed with a well-structured set of guidelines to build a theory from the scratch with a set of unorganized data (Glaser & Strauss 1967). Prior IS researchers have used GTA to answer similar types of empirical research questions (Stilgoe et al. 2013; Urquhart, Cathy and Lehmann, Hans and Myers 2010; Wolfswinkel, Joost F and Furtmueller, Elfi and Wilderom 2013; Gasson 2004; Dedrick, Jason and West 2003; Winter 2008).

The authors could not select between those two methods instantly by looking at previous literature. Hence it was decided to try both inductive[1] and deductive[2] approaches before finalizing the methodology. A theoretical framework is developed through literature analysis (Table 3-3) and themes are generated by using data gathered through preliminary interviews (1 member from each group) (Table 3-4). By evaluating the two tables, the decision is taken to do CSR or GTA. The theoretical framework developed through reviewing literature is as per Table 3-3. The themes and concepts identified through preliminary interview data analysis is in Table 3-4.

Table 3-3: Theoretical framework based on literature

Literature area	Categories	Related content
Research (university view)	Requirements for research	Select research based on market needs and social problems (Tanha et al. 2011). Lack of appropriate evaluation of ideas and innovations of ideas in a national entity (Tanha

[1] The **inductive method** starts with many observations of nature, with the goal of finding a few, powerful statements about how nature works (laws and theories) (Charmaz 2008).

[2] **Deductive reasoning** is a basic form of valid **reasoning**. **Deductive reasoning**, or deduction, starts out with a general statement, or hypothesis, and examines the possibilities to reach a specific, logical conclusion (Charmaz 2008).

		et al. 2011). Insufficient rewards for university researchers (Siegel et al. 2003)
	Product development	There has been very little research on inventions. Most of them are "*proofs of concepts*" or "*prototypes*" (Colyvas, Crow, Gelijns, Mazzoleni, Richard R Nelson, et al. 2002).
	Customer perspective	Researches not completely in line with customer needs (Tanha et al. 2011).
	Project management	Research project management is problematic than other projects, and only a limited amount of theory available for research management (Ernø-kjølhede 2000) Lack of understanding regarding university, corporate or scientific norms and environments (Siegel et al. 2003) Bureaucracy and inflexibility of university administrators (Siegel et al. 2003) Faculty members/administrators have unrealistic expectations regarding the value of their technologies (Siegel et al. 2003) "*Public domain*" mentality of universities (Siegel et al. 2003)
Challenges in software development (industry view)	Resources	Skilled human resources (Munasinghe et al. 2002; Hargadon 2003), Infrastructure support (Munasinghe et al. 2002), Physical equipment (Hargadon 2003), Up to date hardware and software (Munasinghe et al. 2002). Lack of human capital in Sri Lanka (Munasinghe et al. 2002). Few high-end education providing institutions (Munasinghe et al. 2002). Do not have sufficient university-industry linkages (Munasinghe et al. 2002).

	Project management	Sri Lankan projects fail not due to technical issues but due to people and project management issues (Jayawardena & Ekanayake 2010). Application of project management practices and tools which are designed for other disciplines (Jayawardena & Ekanayake 2010).
	Government support	Appropriate policy support (Munasinghe et al. 2002), Domestic market - e-government (Munasinghe et al. 2002).
Technology transfer or research commercialization	Intellectual property rights	Patenting (Colyvas, Crow, Gelijns, Mazzoleni, Richard R Nelson, et al. 2002; Tanha et al. 2011) and firms willingness to search for external knowledge, screening publications (Fontana et al. 2006) Lack of solid rules and regulations for protecting Intellectual Property (IP) rights (Tanha et al. 2011) University are too aggressive in exercising intellectual property rights (Siegel et al. 2003)
	Industry collaborations	Industry partnerships (D'Este & Patel 2005; Fontana et al. 2006; Siegel et al. 2003), Investments, marketing and sales (Tanha et al. 2011). System to share the outcomes in a fair way to motivate researchers (Tanha et al. 2011) Inadequate relationships with regional and global market (Tanha et al. 2011)
	Resources for commercialisation	Lack of adequate venture capital for investment in new technologies (Tanha et al. 2011) Insufficient resources devoted to technology transfer by universities (Siegel et al. 2003) Poor marketing/ technical/ negotiation skills of TTOs (Siegel et al. 2003)

Table 3-4: Themes and concepts identified through analysis of preliminary interviews

Stage	Themes	Concepts	Case Numbers
1 Project initiation	Requirement for research	Community need	1,3,6,7,9
		Software industry need	1,3,4,5,6,7,10
		Specific client(s) need	2, 4,5,8
	Resources	Basic studentship facilities	1,2,3,4,5,6,7,8,9,10
		Studentship time	1,2,3,4,5,6,7,8,9,10
		Project originators' interests and knowledge in the domain	1,2,3,4,5,6,7,8,9,10
		Other funds	2,9
	Community collaborations	Support from the domain experts	2,8,10
		End user collaborations	1,2,3,4,5,6,7,8,9,10
2 System development	Skilled personal	Originators built the first version of the system	1,3,4,5,6,7,8
		Had industry support	2
	Team structure	Supervisors	1,2,4,5,6,7,8,9,10
		Team lead	2,3,4,5,6,7,8,9,10
		Interdependent roles	1,3,4,5,6,7,9
		Computer science students	1,2,3,4,5,6,7,8,9,10
		Researchers from other disciplines	7,10
3 Implementation5	Immediate end users	University students and staff	1,3,4,7,9
		A client/few clients	2,5,6,8,9
	Intellectual property rights	Licenses or agreements	1,2,3,4,5,6,7,8,9,10
	Financial capability	Investors	1,3,10
		Funds	2,5,6,7,8,9,10
		Instant revenue	4,9
	Marketing and communications	Through personal contacts/word of mouth	1,2,3,4,5,6,7,8,9,10
		Publications	1,2,3,4,5,6,7,8,9,10
		Other marketing activities	2,
	Type of systems development entity	Research groups	1,3,4,5,6,7,8,9
		Industry-university partnerships	2,10

After the evaluation, the authors noticed that the factors affecting university based software development research projects are vastly different from the factors addressed in the standard software development lifecycles (Ruparelia 2010). Grounded theory is appropriate to develop theory by using unstructured data collected in a natural setting (Strauss & Corbin 1994). Therefore it was decided to use GTA for deriving the framework inductively, and then CSR to evaluate it deductively based on a single case study. CSR has the benefit of studying or testing the contemporary events that researcher has a little or no control over (Yin 2014). The authors did not try to change the way the researchers work, instead studied what they did every day, what went right, what went wrong and how did they overcome issues. Thus CSR was selected for deductive theory testing, over action research or DSR. Only one case study was possible because of accessibility and time availability of participants. A quantitative forced ranking mechanism is also used to validate the findings from all the participants.

3.3. Data collection

Data are gathered from both secondary and primary data sources. Data for *Google* and *Linux*, well-known research based projects, are taken from secondary data sources. Primary data collection is conducted within Sri Lankan universities. The specific data sources, data collection methods, general information of selected projects and research groups are described in this section.

3.3.1. Secondary data

Two (2) well-known software innovations evolved accidently through research are studied to see if there are any visible management practices. Stories of Google (Google 2014; Vise & Malseed 2008), and Linux (Raymond 2001; Torvalds & Diamond 2001) projects are taken from secondary data sources as they are published by the original founders or their finest followers.

1. Google Inc.

Page and Brin invented the first version of Google search engine while they were research students at the Stanford University. Their problem was "*not having a full-scale search engine that can explore across the internet*" (Vise & Malseed 2008). They had an idea to make "*searching*" the most important mechanism for internet users to surface through the entire web (Google 2014). Page, Brin, and their professor were the initial investigators of the project. Internal university resources were used to implement a prototype of their innovative search engine. The prototype; Google.stanford.edu, sooner became known by the academics at Stanford through word of mouth. That way their end user community increased in a very short time. Later, a set of hardware equipment were funded by some other project at their university to increase speed and capacity in order to serve some more users. They had to keep the interface very simple because they did not have enough funds to hire a graphic designer. There were no particular practices or organization structures at the beginning. There were regular discussions with the professor and agreed upon division of work between the two team members in their areas of interest. By the time they finished their PhD research, they had an innovative idea, a prototype implementation and a group of users. From the Google's story some practices of university projects are identified as: discussions with the professor, agreed upon division of work between two research fellows and awareness creation through word of mouth. Simple interface is a strategy they used to overcome funding problems. They focused only on developing the core functions for an ideal search engine.

As the database and number of users increased very fast they needed more computers and financials (Vise & Malseed 2008). They were funded by the Stanford Digital Libraries project. When they were trying to commercialise the product, it didn't seem to matter that they had something better. While Page and Brin saw the search engine as the most important part of the internet experience for computer users hunting information, others saw it only as a helping tool (Vise & Malseed 2008). They were rejected by many popular investors including Yahoo. After putting a huge effort to find investors, Andy Bechtolsheim, a cofounder of Sun Microsystems liked their idea and agreed to fund the project. They came up with a revenue model for advertisers who are interested in reaching out to the online users (Google 2013) . They created text-based adverts through a self-serve auction-based advertising program. That is how Google has become the most used search engine throughout the world. Currently, Google is a well-established company which has its own code of conduct. Google has started in a very informal way and now carried out its operations as a well-structured world class organization (Google 2014).

2. Linux

Unix and Linux operating systems are developed by a set of early *Hackers* (Raymond 2001). Hacker culture is emerged through a high-speed computer network introduced by the United States defense department to connect all universities and research laboratories around the country in 1969. It was a great opportunity for researchers to share knowledge and resources with a larger group of people with similar interests(Godfrey & Tu 2000). As a result they started developing, reusing, and consuming software components collaboratively. In 1991, Linus Torvalds initiated Linux operating system by using reusable tools from the Free Software Foundation and made it free to use and customise the source code (Raymond 2001). By this time the internet was launched and available to the general public. The success of Linux is the idea of a free operating system with the source code of the first version of the software available over the internet, and a group of hackers who would like to use, and contribute in further development. It is also freely available to the general public to download through internet and use. Linux developers did not need expensive resources because of their shared development culture. At the beginning there were no hierarchical structures or time restrictions for volunteered developers. However, Linux development model opened up many pathways to the developers and the researchers in the field.

3.3.2. Primary data - set 1

To assess the applicability of existing practices in the industry for software development projects in universities, an open ended questionnaire (Appendix I - Questionnaire to find differences between industry and universities, page I) is distributed among practitioners in Sri Lankan universities and software companies. The personal details of the participants are anonymised because of confidentiality issues.

Random sampling method is used to select participants from the industry (Hippel 2002), but from people who are interested enough to participate on this study. A list of software companies in Sri Lanka is taken through an industry guide[3]. Software development team members of those companies are considered as the population for the study. Second, the contact emails are collected from each website and sent a standard email asking to distribute the link to the questionnaire within the company. There were 41 companies in total. Fourteen (14) responds from fourteen (14) different projects in eight (8) software companies are received. This is considered as a fairly representative sample (Creswell 2007), as different team members responded such as software engineers, project managers and business analysts. Hence collective interpretations of all types of industry practitioners are gathered. Details of the respondents from the industry including their destinations and years of experience are given in Table 3-5.

Table 3-5: Industry respondents’ details

Company	Project description	Respondents	Designation	Experience (years)
C1	Stock management systems	1	Senior software engineer	4
		2	Senior business analyst	4
		3	Project manager	11
C2	Financial systems	4	Senior software engineer	7
		5	Software engineer	5
C3	HR management systems	6	Software engineer	3
		7	Software engineer	3

[3] http://srilankasoftwarecompany.blogspot.com/

C4	Marketing websites	8 9	Chief technology officer Senior software engineer	21 7
C5	eLearning systems	10 11	Technical lead Software engineer	8 4
C6	Embedded systems	12	Software engineer	3
C7	ERP systems	13	Business analyst	8
C8	Word processing software	14	Network administrator	6

14 questionnaires are distributed among 14 a representative sample (Coyne 1997) of researchers from two of three top ranking universities in Sri Lanka (i.e. University of Moratuwa and University of Colombo)[4]. These two universities are selected based on their technological advancement and number of research projects listed on their websites (Anon 2015). In order to discover the collective interpretations of researchers, viewpoints are sought out from all senior and junior levels (Isabella 1990). Participating researchers represented different roles such as lecturer, supervisor, investigator, student, research assistant, develop and so forth. At a broader view two clearly distinctive roles are captured i.e. research supervisors (e.g., professors) and assistants (e.g., students). Some of the university respondents have dual roles. These roles are part-time or full-time basis depending on their lecturing and research workload. University respondents' details including their destinations and years of experience are given in Table 3-6.

Table 3-6: University respondents' details

University	**Number of respondents**	**Projects**	**Destinations**		**Experience (years)**
			Full time	**Part time**	
University of Moratuwa	6	A1 A2 A3 A4 A5 A6	Lecturer Senior lecturer Research assistant Researcher MPhil student Consultant	MPhil student Investigator Developer Developer Developer -	5 12 3 4 2 5

[4] Webometrics Ranking of World Universities - http://www.webometrics.info/en/Asia/Sri%20Lanka%20

University of Colombo	8	B1	Research assistant	Developer	3
		B2	Senior lecturer	Investigator	16
		B3	Senior lecturer	Investigator	14
		B4	PhD student	Lecturer	8
		B5	Research assistant	Developer	3
		B6	PhD student	Lecturer	7
		B7	PhD student	Developer	5
		B8	PhD student	Developer	6

3.3.3. Primary data - set 2

Similar to other qualitative studies this study uses nonprobability sampling strategies to select the study sample (Creswell 2007). Nonprobability samples offer the benefits of not requiring a list of all possible elements in a full population and the ability to access other highly sensitive or difficult to research study populations (Berg 2001). Convenience sampling is one of the nonprobability methods. Convenience sampling is sometimes referred to as an accidental or availability sample. This category of sample relies on available subjects-those who are close at hand or easily accessible (Berg 2001). For example, it is fairly common for college and university professors to use their students as subjects in their research projects. Afterwards, from the convenience sample chosen, more appropriate and representative sample is selected for the study using theoretical sampling strategies (Coyne 1997). Therefore, most relevant, sustainable and economically and research wise successful software development research groups are chosen from the two main computer science and IT universities in Sri Lanka. Criteria for the selection is tabulated in Table 3-8. The population for the study – software development research groups - is collected from the university websites. The projects list that is taken from University of Colombo, University of Moratuwa, University of Peradeniya and University of Ruhuna websites is given in Table 3-7.

Table 3-7: Population of research groups for the study from Sri Lankan universities

University	**Institution/Faculty**	**Project**
University Of Colombo	UCSC	Vidusayura
		Harbor management system
		PAN Localization
		Online Handwriting Character Recognition
		Data-driven Speech Translation between Sinhala

		and Tamil
		Railway Traffic Optimization System
		Social Sensor Networks for Opinion Analysis
		Automated Fingerprint Identification System
University Of Moratuwa	Dept. of Information Electronic & Telecommunication Engineering, Faculty of Information Engineering Electronic Systems Research Laboratory	Machine Vision Based Intelligent Surveillance System for Expressways
		Premium International-UoM Research Laboratory for Biomedical Technologies
		Computer Vision, Machine Vision
	Dialog-UoM Mobile Communications Research Laboratory	
	Department of Electronic & Telecom. Engineering, University of Moratuwa	Project Multipurpose Self-Configurable Indoor Wireless Sensor Network for Green Buildings
University Of Peradeniya	Dept. of Education, Faculty of Arts	Mobile phones on a private network for science teaching and learning in schools
	Department of Statistics & Computer Science, Faculty of Science	Towards Better Performance: Development of an Automated Tool for Performance Evaluation in Sports using Trajectory Analysis
		Intelligence Led Policing Using Data Mining Techniques
University Of Ruhuna	Department of Chemistry	Investigation of structure, dynamics and energetic of mixed transition metal clusters

However, after few preliminary interviews, University of Colombo and University of Moratuwa are selected to conduct the study extensively as they had the most number of projects, diverse research groups, and because of accessibility of information and convenience. Among the project groups those who agreed to participate in the study by sharing information and spending time for interviews, 10 groups are selected for the final study based on their consecutive years of operation as a research group (i.e., more than 2 years). Therefore, 2 year research master degree projects are excluded from the final sample. Selected projects from those two universities are given in Table 3-7.

To investigate the challenges and best practices in university projects, this data collection had been done using semi structured interviews (based on Appendix II - Questionnaire to identify challenges and best practices in university projects, page II). There are two types of software development projects carried out by research groups in Sri Lankan universities. The main type of projects is research and learning oriented, whereas some of the projects are commercial oriented. The authors have interviewed participants from ten (10) research groups. More than three (3) interviewees from each project except for one which had only two (2) members are interviewed to improve the accuracy and validity of information, therefore, triangulation methods are applied. There are at least one senior researcher and one junior fellow from each group. Productivity, continuity and stability related measurements of the research groups are given in Table 3-8. More information on those research groups are discussed afterwards.

Table 3-8: General details of the research groups

	Measures	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**	**G9**	**G10**
1	Consecutive years of operation	3	2	8	7	4	5	10	12	4	4
2	Job opportunities (#)	6	2	20+	20+	8	20+	20+	20+	2	20+
3	Research opportunities (#)	8	2	3	20+	8	1	20+	20+	1	7
4	Projects undertaken (#)	1	1	16	9	3	12	8	10	1	1
5	Research grants (#)	2	1	0	6	0	0	10	0	1	3
6	Investments/payments from private sector (#)	0	0	16	6	3	12	1	10	0	0
7	Publications (#)	3	2	3	20+	6	3	20+	20+	4	20+
8	Systems implemented (#)	1	0	14	7	1	10	5	8	1	2
10	Formal customers (#)	0	0	13	5	1	1	1	6	0	1
11	End users (#)	~50	0	1k+	1k+	~80	1k+	1k+	1k+	0	~100
12	Participants (#) / members (#)	3/3	2/2	3/18	8/23	3/9	3/12	6/7	3/13	3/9	3/22

G1 a software development research group from the University of Colombo. They started a project based on a research idea to advance current learning system of university students.

The group consists of three members including an MPhil student and two supervisors. The system is a gaming component for university students to learn computer science. The MPhil student did the research, developed a prototype system and evaluated the system with a set of students. Although the concept is evaluated and found successful, they did not have funds to deploy it for a larger user base. They have tried, but did not receive any research grants, except from university allocation for publications. Currently, the project is struggling to continue because of lack of resources and time, for the MPhil student to develop a full scale functioning system, although there is a potential user base of more than thousand (1000) university students.

G2 is a geographic information systems (GIS) research group from the University of Colombo. Their objective was to find a solution to a problem in current GIS using maps in Sri Lanka. The group consists of two members including an MPhil student and a supervisor. The system is developed as a plug-in to an existing application. After developing and testing the component, they have written a funding proposal to implement the solution. It did not get accepted; therefore project is terminated after fulfilling the MPhil degree.

G3 is a software development entity within the University of Colombo which is started with the intention of training university students and helping to develop research based software products. It cannot be explicitly mentioned as a research group. Similar to a software company, G3 undertakes software projects from corporate customers. Also they collaborate with researchers to develop software products. If they need research to be done for commercial products, they outsource it to researchers. When research groups need skilled people for software development, G3 helps them. They have a formal as well as a mutual relationship with the university, its students, staff and researchers. They provide expertise knowledge and consultation to government and private organizations with the help of academic staff and researchers.

G4 is a modelling and simulation research group from the University of Colombo. They started a project based on a customer request to develop a ship simulation system. Although the customer withdrew the request, a group of academics continued to research on modelling and simulation systems. The initial team consisted of two (2) researchers and five (5) supervisors. They developed collaborations with established centres in university and receive financial help by providing services in their expertise area. After sometime, they receive several research grants and had industry partnerships. They got more resources using those funds. Later, they started to undertake short term modelling and simulation projects from corporate customers. With the financial gain, the team grew over time with more resources and people. At present, G4 is an established research group that do local and international projects collaboratively while maintaining a smooth cash flow.

G5 is a software development research unit attached to the University of Moratuwa, founded and invested by a private company. They take on research students to develop research based software solutions for computer networks related technical problems of the company. The students are supervised by the senior academics of the university. They collaborate with other local and international research groups to get expertise technical knowledge and support. Once deployed the software component is maintained by the company technicians. Research student passes the knowledge and train the technicians to do so. That constitution contributed good and productive research to the nation, for four (4) years and continuing to the fifth year.

G6 is another software development research unit, funded by a private company as a partnership with the University of Moratuwa. The company outsource their software development work to the lab, based on a revenue sharing model. At the beginning there were research components, as the lab was initiated to do research. However, it has changed to a software development unit without much research. With the software development work load, and deadlines given by the company, sometimes it is hard to conduct research. Therefore, right now it is questionable if it is a successful partnership for research, or just a software development training unit for university students.

G7 is a research lab, at the University of Colombo, dedicated to do language technology research in the country. Initially the entity was created by a set of local researchers with a grant received by an international association for language research. They had completed the research and developed the software components, in the research proposal. Now they are using the lab and equipment to do more language related research and commercial projects, with new funding sources. Currently the group consists of three (3) PhD students, two (2) language technicians, a research assistant and their supervisor. There are local and international collaborations with other language technology research centres and beneficiary organizations.

G8 is a research unit partnered with Sri Lankan universities (i.e., University of Colombo and University of Moratuwa) and few companies in the software industry. The companies sponsor research based product development done by students and researchers in universities. They collaborate with international academics to get new technology related knowledge. This constitution has been doing productive for more than twelve (12) years consecutively, because of the dedicated partnership from academics and industry companies.

G9 is a research group from the University of Colombo started with the intention of developing a software solution to a critical problem in rural villages of Sri Lanka. They developed a low cost solution with a limited amount of technology which is affordable to the rural community. They could not keep close relationships with the actual beneficiaries of the system, because of travelling and communication shortages. Shortly, a research grant and support was received from some government organizations to implement the solution. With the support they could carry out a pilot implementation, with the actual users. However, the users did not continue to use the system in that location. Although funds and equipment were available, some researchers were dissatisfied from the failure and the group did not keep up to implement the system in more locations.

G10 is a distributed research group, which is attached to the University of Colombo, Sri Lanka and work together with universities in Australia, UK, and Italy. They are developing an information system for farmer communities in villages of developing countries. The group has built end to end relationships with end users and related government departments. More than twenty (20+) jobs and seven (7) research opportunities are provided all together. Three (3) research grants have been received from Sri Lankan funding agencies. The authors could not find information on international funding sources. Currently they are doing final evaluations and pilot implementations of web and mobile versions of the system, with Sri Lankan farmer communities.

From the above research groups; G1, G2 and G9 are identified as less successful groups in doing productive and continuous research. They had functioned only for 2-3 years, because they could not overcome challenges faced in developing research based software products. Data collected through those participants are used to identify those challenges. G4, G5, G7, G8 and G10 are identified as more sustainable and established research groups. G3 and G6 were more like commercial software development entities within universities. They provided training and internships for university students, and sometimes helped researchers to develop software products.

3.3.4. Ethical concerns

As the authors conducted research with human subjects, ethical concerns are important. During the entire study ethical priorities such as honesty, openness of the study's intent, respect for subjects, privacy, anonymity and confidentiality are obeyed. All the participants of the study were willing to participate voluntarily. A description about the project is given and gathered consent to use data in forms of publication (Appendix VI – Consent form, page XIV).

3.4. The overall methodology

Firstly the research problem is brainstormed among the study's investigators. Few preliminary discussions with senior researchers from Sri Lankan universities are conducted, in order to understand the problem in depth. The research questions are formulated to carry forward the research. Through a comprehensive literature review, a gap in existing knowledge is identified and explained. Therefore, the authors decided to address the research problem by doing a longitudinal study. The overall methodology is illustrated in Figure 3-1.

3.4.1. Grounded theory approach for inductive theory building

The Grounded Theory Approach (GTA) is used to discover theory from systematically obtained and analysed data (Glaser & Strauss 2006; Corbin & Strauss 1990). Relevant predictions, explanations, interpretations, and applications, of the collected data can be derived. Especially, it can be used to build and explain a study from the very beginning with unstructured data.

The literature review is used as the instrument to start with GTA. Input elements (Table 2-4) related to managing university research projects and software development projects are gathered through literature to define boundaries of the study. Although Glaser & Strauss (1967) originally argued against using literature as the foundation to start a research study, Dunne's (2011) justified the stance of the literature review. Since both university research and software engineering methodologies had a long history, it was not reasonable to completely ignore the existing knowledge. However, only the basic set of input categories were selected through literature. There were very little knowledge on research management and no literature found on research based software development in universities. Therefore, the authors argue, that the literature review have not interfered the originality of findings, but had used to define the basic input elements (Table 2-1) of a research.

Table 3-9: Inputs identified through literature

Inputs (literature)	Requirement for research and product development, ideas, people, personality traits, resources, organization structure, communication, goals and objectives, partnerships, other practices

With the preliminary interviews; possible processes, outputs, other important facts and uncertain factors are identified. Using open coding, literature inputs and content from preliminary interviews were analysed. Through that basic concepts were identified divide under four categories as given in Table 3-10. This categorisation is used throughout the study for content analysis and report findings.

Table 3-10: Basic inputs, processes, outputs and other factors identified through open coded preliminary interviews

Inputs	Requirement for research and product development, goals and objectives, financing and resources, project team, communication
Processes	Software development methodologies, life cycle phases, milestones, interactions with end users, relationships with community groups, design methods and programming languages
Outputs	Outcomes
Other	Important facts, risks and uncertain factors

Afterwards, the authors followed the a selected set of GTA procedures and canons explained by Corbin & Strauss (1990) to give rigor to this study. The overall methodology (Figure 3-1) is decided based on these canons and procedures.

1. Data collection and analysis are performed as interrelated processes

Data collection and analysis are done simultaneously since the literature review to final case study analysis. Coding and memoing are used to make meaningful interpretations of data from the beginning of analysis (Miles et al. 2014). Based on that, questions, participants, interviews, documents, or areas for observation, are decided for the next dataset. Likewise, analysis of each dataset directed the next round of data collection. The basic concepts identified at the beginning are repeatedly addressed in all forms of data collections to see if they have any significance or not. Systematic field notes are taken by the main author during all forms of data collection techniques.

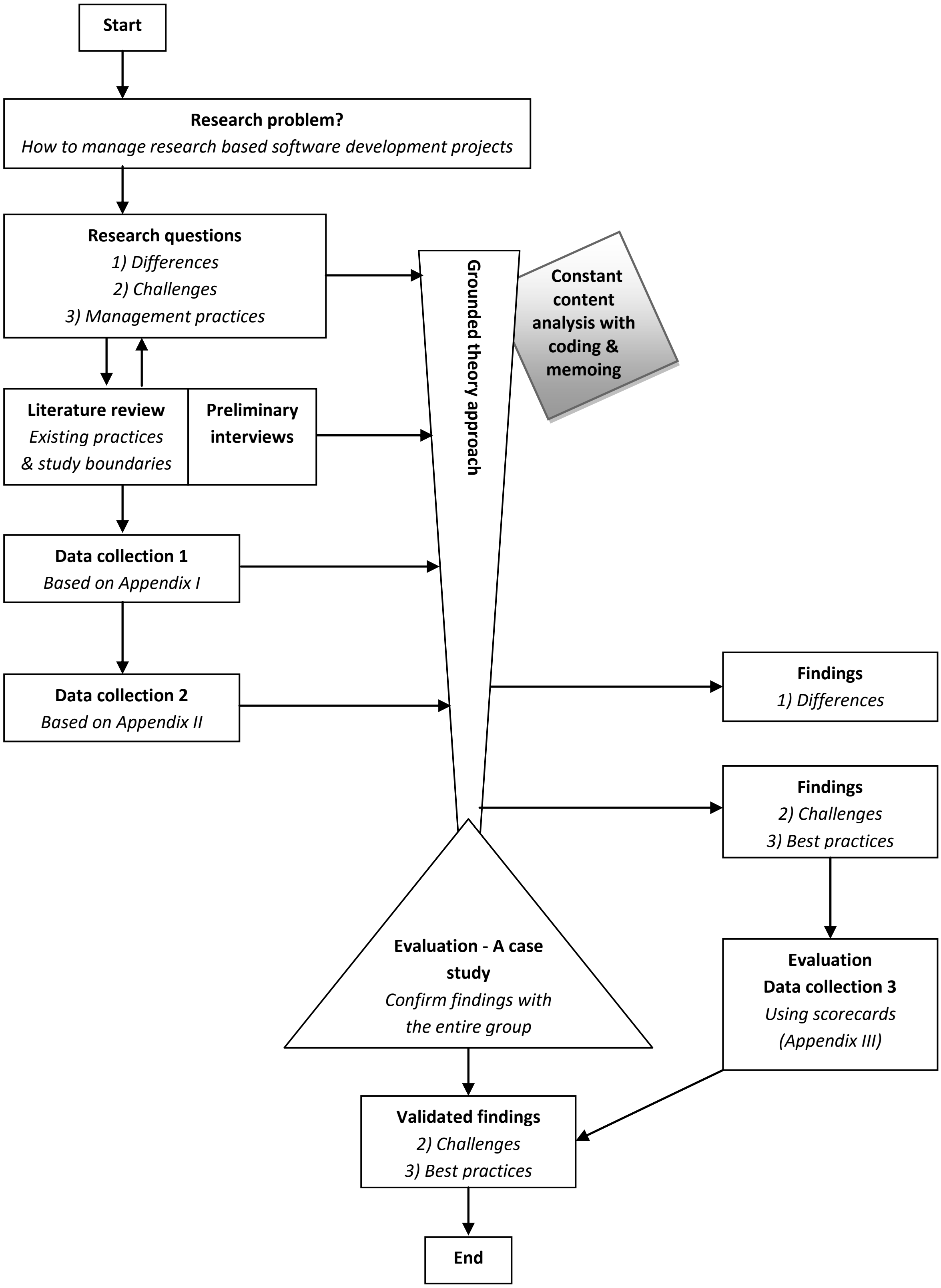

Figure 3-1: The overall methodology

2. Concepts are considered as the basic units of analysis

Theories of this study are built upon the concepts. Concepts could be events, happenings, actions, or incidents, of a particular phenomenon. In this study, concepts were identified at the beginning through literature review and preliminary interviews. Those concepts are used as labels to categorize the raw data when coding.

3. Categories are developed and interrelated

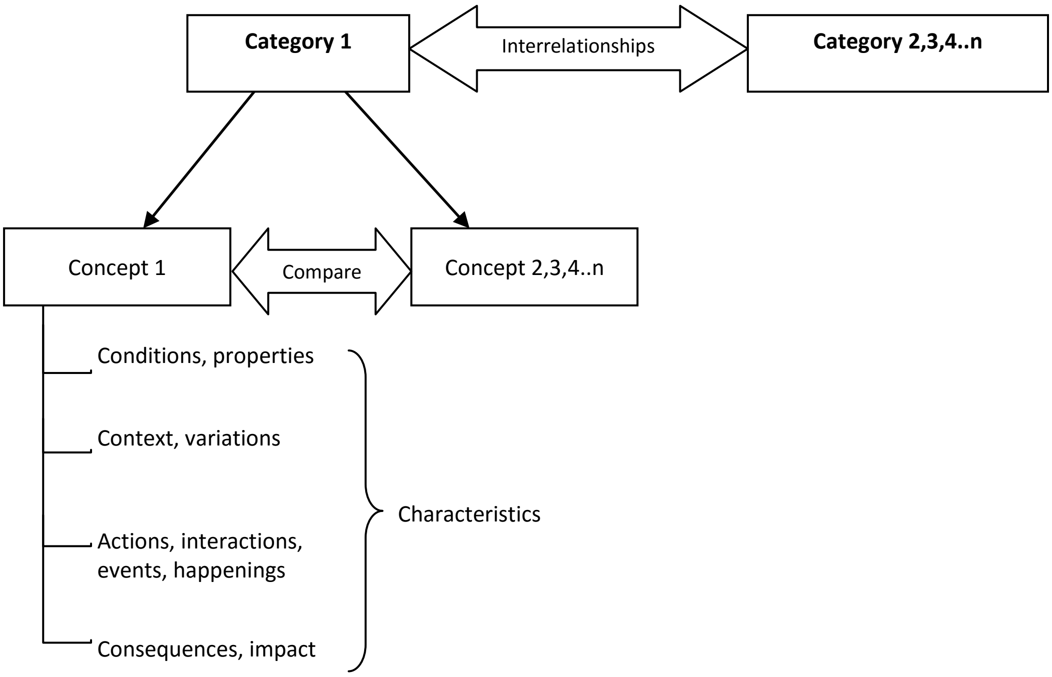

Figure 3-2: Formation of categories, concepts and relationships in GTA

Categories are considered as the "*cornerstones*" of a developing theory. Meaning of a theory is provided by its concepts and categories. Concepts with similar meanings are grouped to form higher level categories. Once a category or abstract concept is identified, its characteristics are investigated in order to develop its properties, dimensions, variations, interactions, consequences and conditions to be applied. The authors' formation of GTA concepts, categories, and relationships are demonstrated in Figure 3-2.

4. Sampling proceeded on theoretical grounds

The authors focused on representativeness of concepts rather than persons. Appropriate individuals, groups, and organizations are selected to match the identified concepts, and their characteristics of the phenomenon being studied. University researchers are studied, in the aspect of how they manage to do research, develop and implement software products, as a group. University research labs, industry research labs attached to universities, and individual researchers, professors who worked in universities are studied to understand how they manage and what challenges they face. Also some data are gathered from industry practitioners through semi structured interviews to understand their practices in order to compare, and contrast, the characteristics. The incidents, events, and happenings that shape what the participants do; the conditions or barriers that provide or interrupt; and the resulting consequences are investigated. The number of occurrences of events are counted from the field notes if the frequency is useful. Once identified a concept has been studied continuously in each data collection and analysis phase in order to maintain consistency and guard the authors against biasness. Categories, and related concepts, are prioritized considering their impact on the study phenomenon. As an example; what types of work do they do mostly, such as reading, designing, concept development, software development, formal meetings, informal discussions, etc. Do they engage in research work all the time, or software development work is also done part time, what conditions that enable or prevent them doing so. The nature of rare and unusual events are investigated. Sometimes, these events were more important when formulating findings.

5. Analysis is done using constant comparisons

First the characteristics of identified incidents are compared to see the similarities and differences. Constant comparisons help to group the concepts more precisely with a minimum impact of researcher bias.

6. Patterns and variations are accounted for

Regular patters and unusual variations of data are identified and addressed whilst analysing the content. In this case, more researchers are interested in brainstorming sessions where research ideas are generated, only few researchers were interested in business meetings with the industry partners. These patterns helped to formulate findings.

7. Processes are built into best practices

A process can be emerged by dividing a phenomenon into several phases, stages, or steps. Some researchers may develop software products whilst research is progressing, whereas, some might finish research first and then do software development work. Thus in this study, the processes are explained individually in the form of best practices.

8. Writing theoretical memos used as an integral part

Memos are kept in order to track all the work, throughout the research from the literature review. They are used in coding, theory formulation, and revising. During coding, code notes are kept to memorize the significant aspects in codes. Memos and code notes are used as a base for writing the research. Sometimes they are elaborated more into writing itself.

9. The main author did not work alone

Multiple investigator triangulation methods are practiced amongst the group members, thus worked collaboratively for planning, analysis and interpreting findings, although the data collection is done entirely by the main author.

10. Broader structural conditions must be brought into the analysis, however microscopic in focus is the research

As shown in the Figure 3-1, GTA started with broader conditions and incoming data collections, and later narrowed down to the specific challenges and best practices in the context. GTA steps that demonstrated in Figure 3-1: The overall methodology, is summarised in Table 3-11.

Table 3-11: Steps followed in GTA

Steps	GTA procedures
Step 1	Basic concepts are identified from Literature review and content from preliminary interviews. Memos and code notes are maintained throughout the project.
Step 2	Categorized them under abstracts headings. Open coding is applied.
Step 3	Data collection 1 is done based on Appendix I - Questionnaire to find differences between industry and universities with participants explained in Table 3-5 and Table 3-6.
Step 4	Similarities and differences in university projects and industry projects are identified by applying Axial coding to analyse content. Findings are formulated under the concepts (section 4.1. Analysis between industry projects and university projects, page 63) and proved the need to investigate challenges and best practices unique to universities.
Step 5	Semi structured interviews are conducted with the participants from selected research groups (Table 3-8) to understand challenges and best practices in universities by using Appendix II - Questionnaire to identify challenges and best

	practices in university projects.
Step 6	Selective coding is used to identify challenges and best practices unique to universities. Findings are reported in two sections as; 4.2. Challenges faced by (page 74) and 4.3. Best practices in managing research based software product development in universities (page 77).

3.4.2. Data analysis and interpretation

Once collected, data are organized in the forms of field notes, audio recorded interviews, and photographs. Transcribing is used to convert them into write-ups or tabular formats. NVivo software package is used to store and get some interpretations of data. Ratings or judgments are made to the transcribed data using different analytical techniques. Coding and memoing methods explained by Basit (2003) played a significant part in the process of analysing textual data. As described by Miles et al. (2014) codes are used to provide a symbolic meaning to data collected by assigning identifiable labels based on identified categories. Memoing is used to keep notes on research ideas, questions, and code description as explained by Corbin & Strauss (1990). Both coding and memoing processes are used to make meaningful concepts out of data when doing categorization and analysis.

When coded, the authors could easily get together all the data chunks relevant to a particular category, concept, or a research question. An example for a coded piece of formatted interview recorded data is given in Table 3-12:

Table 3-12: An example for a coded piece of interview data

[a] Budget restrictions are there as software had to be developed in a low budget. For a simulator there should be a fast frame speed. But within low budget computers and equipment, it was a risk. As this was a research project, it was hard to meet the exact deadlines.	[a]FINANCING AND RESOURCES

Open, axial, and selective coding types explained by Corbin & Strauss (1990) are used for GTA content analysis. Those coding types are explained below;

1. Open coding

The goal of open coding was to categorize data based on the selected categories and standard ways of thinking by the authors. Multiple investigator triangulation method is used here. The basic forms of events/actions/interactions are grouped by considering the similarities and differences. Then the abstract concepts are identified and labelled the groups with similar concepts. This technique helped to reduce the subjectivity and bias of the researcher.

A content analysis example of open coding from this study data is given in Figure 3-3. "*Research problem*" is the main concept and four balloons include the content segments that supported to come up with the main concept.

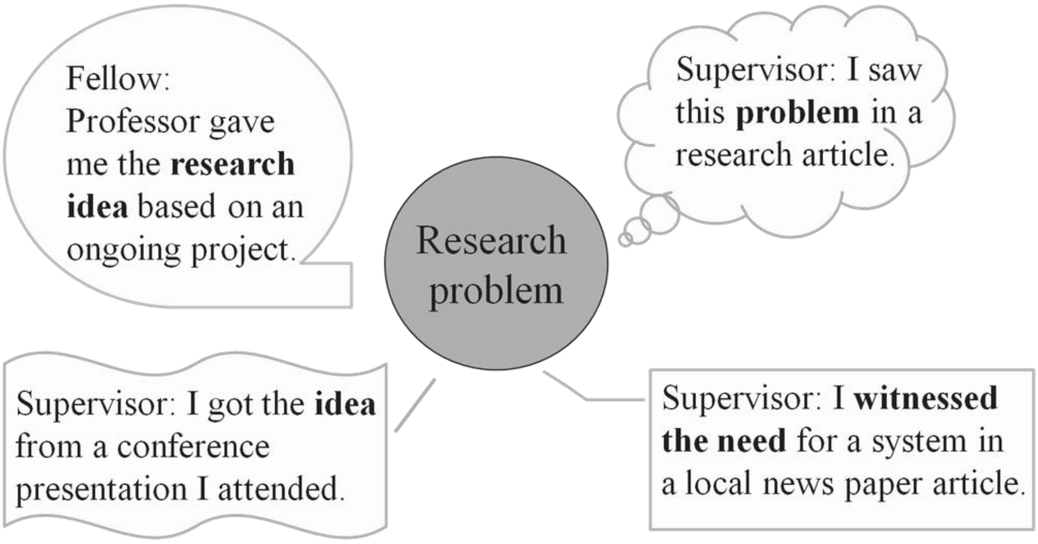

Figure 3-3: Example of content analysis in open coding

Selected concepts based on the frequency of relevant content are shown in (Figure 3-4). From the projects which had more than three participants, more descriptive three responds are demonstrated in the figure. Content frequencies are illustrated under seven (7) themes namely: research problem, people, software development methodologies, project management, funding, communities and meetings. The similar content that considered as relevant to those themes are mentioned within brackets. Number of times those fragments are found in interview transcripts of each participant is calculated and total is taken from all the participants from all groups for each theme. As seen in Figure 3-4, most participants have talked about "*people*" and "*funding*" factors. In the next level of coding (i.e., Axial coding), more meaningful concepts are derived from these abstract themes.

- Research problem (idea, requirement, need)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	23	35	12	22	13	26	45	8	15	31
P2	28	18	24	28	7	33	25	24	12	9
P3	14	-	8	24	4	42	15	12	24	16
Total = 597	65	53	44	74	24	101	85	44	51	56

- People (retention, researchers, fellows, professors, supervisors)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	27	34	28	38	37	41	19	21	24	43
P2	13	24	27	24	26	16	8	24	27	23
P3	24	-	35	43	27	7	22	35	38	27
Total = 782	64	58	90	105	90	64	49	80	89	93

- Software development methodologies (method, Agile, FOSS, random, lifecycle)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	31	26	15	17	24	26	23	15	19	6
P2	25	12	27	26	18	16	18	17	35	29
P3	8	-	19	14	21	31	28	18	14	26
Total = 604	64	38	61	57	63	73	69	50	68	61

- Project management (management, timeline, deadline, team, leader, coordinate)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	15	23	18	26	28	19	24	27	34	12
P2	19	16	13	28	29	27	17	26	33	29
P3	16	-	19	22	14	29	11	28	16	23
Total = 641	50	39	50	76	71	75	52	81	83	64

- Funding (agency, salary, resources, proposal, award, payment,)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	22	19	12	17	28	36	26	24	29	32
P2	15	18	23	29	18	5	28	14	24	27
P3	24	-	28	18	22	25	19	27	26	31
Total = 666	61	37	63	64	68	66	73	65	79	90

- Communities (partnerships, industry, collaboration, international, government)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	16	24	27	17	19	27	15	29	12	23
P2	28	26	13	21	23	28	16	14	19	28
P3	21	-	24	18	16	22	18	25	16	24
Total = 609	65	50	64	56	58	77	49	68	47	75

- Meetings (discussion, Skype, panel)

Participants	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
P1	7	15	16	18	11	14	9	19	13	16
P2	19	8	18	19	22	16	13	8	16	21
P3	17	-	12	24	15	18	4	14	17	11
Total = 430	43	23	46	61	48	48	26	41	46	48

Figure 3-4: Open coding content analysis word frequency figures

2. Axial coding

Axial coding is concerned on conditions, context, actions, and consequences of concepts grouped under one labelled category. More data are collected to address those factors about identified concepts in the next dataset. Concepts are analysed and re-grouped based on those factors in order to create more meaningful categories. Axial coding in GTA looked for all possible variations in the phenomena. This way the authors identified new management practices, due to variations of conditions, context, actions, and consequences.

More meaningful seven (7) themes are identified after axial coding. They are:

- People retention
- Economic uncertainty
- Access to publications
- Software product
- End users interactions
- Industry partnerships
- Product maintenance

For an example, few content segments of the first code "*people retention*" is given in Figure 3-5.

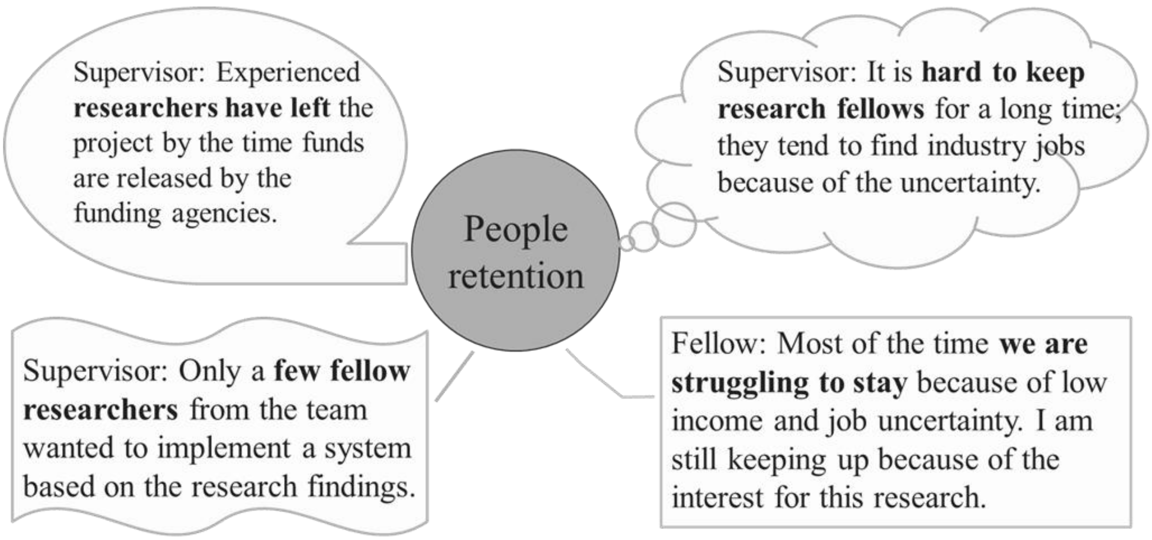

Figure 3-5: Example of content analysis in axial

3. Selective coding

Selective coding was the final stage of GTA data analysis (Scott & Howell 2008). Findings are summarized in this stage of the study. The findings consisted of unique features of university projects, possible challenges, best practices, and finally a framework is developed based on the conclusions. Matured memos are also used to finalize the findings as suggested by Glaser (2012). Theoretical sampling strategies are incorporated at this stage to generalize the findings and include global perspectives to the framework. Some more data segments and analysis tabulations are given in Appendix V - Example data collection and analysis sheets (page VI).

3.4.3. Forced ranking for evaluation

Two types of forced ranking forms are formulated to validate finding in challenges section (4.2. Challenges faced by academics when managing research based software development projects, page 74) and best practices section (4.3. Best practices in managing research based software product development in universities, page 77). The forms are given to all the participants and asked to identify all the challenges faces and best practices used by them in their research groups. Also they were asked to prioritise the challenges and best practices, by numbering them in a sequential order from 1. Number 1 is considered as the most significant challenge or practices to a participants study. The evaluation results are given in (5.1. Scorecard, page 87).

3.4.4. Case study research for evaluation

Generating theory goes hand in hand with verifying it. Case study research (CSR) can provide insight and add more rigor into the context that had been studied in a large-scale with GTA. Although scorecards and multiple investigator triangulations are used to verify the findings, one case study is discussed here to add more rigor to the study by providing in depth information of a research group. CSR is used as a evaluation method in comprehensive studies by previous researchers to further verify their findings (Yin 2009; Laws & Mcleod 2004). Through the case study the authors have captured the complex scenarios over time in the real world context by collecting comprehensive data from research group members who play different roles in university research projects. The study is designed according to the Yin’s (2014) book of Case Study Research; specifically CSR as a part of larger evaluation (Appendix B, pp. 220-222 of the book). Six sources of evidence; i.e. documentation, archival records, interviews, direct observations, participant-observation, and physical artefacts, are used for data verification in CSR. Same transcribing, coding, and memoing methods are used in GTA are applied in CSR as well.

The overall mission of this study is to propose best practices for novice researchers who would attempt to initiate research groups that hope to deploy working software products as outcomes of their research findings. The case study aimed to study a successful and complex research group in depth, and report the findings in order to create awareness amongst Sri Lankan research community. Here, the meaning of successful is the balance between research and product development, specifically; the development and deployment of several innovate software products through academic research, while publishing findings in high quality journals and conferences. The complexity is measured by the number of consecutive years of operation, employment and research opportunities given, projects undertaken, grants and investments received, publications, systems implemented, formal customers, and end users (Table 3-8: General details of the research groups).

The case study is focused on second and third research questions; challenges faced by the researchers, and their ways of overcoming them, as industry analysis is not applicable. Therefore, the case study's questions are as follows;

1. What are the challenges faced by the research group?
2. How did they overcome those challenges? What are the best practices used?

In this section, the findings derived through GTA are evaluated extensively with more rigor, validity, and relevance. Relevance was not an issue when selecting a research group to be studied, as all the projects were selected from Sri Lankan universities. However, the most relevant and accessible research group is selected by rating the groups based on complexity measurements and accessibility to multiple sources of evidence in particular; interviews with all the group members, workplace and participant observation, documents, records, and other artefacts (Table 3-13). Case study evaluation confirms and corroborates the findings by triangulating the evidence, thus improving validity of the previous results.

Table 3-13: Accessibility measures applied for case selection

	Accessibility measures	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**
1	Access to interviews with all group members (Y/N)	N	N	Y	Y	N	N	Y	N
2	Access to documents, records, and other artefacts (Y/N)	Y	N	N	Y	N	N	Y	Y
3	Access to observation (Y/N)	Y	N	Y	Y	Y	N	Y	Y

From the measures in the Table 3-13, it is evident that the authors had full-access only to G4 and G7 research groups. On the other hand, those two groups don't have any null (0) values, compared to other projects when referring to Table 3-8. Therefore, the balance of research and commercial components are assessed to select the most relevant research project from those two. As you can see, research related components are similar in the two research groups. G4 had five (5) formal customers, delivered seven (7) systems, and received six (6) private sector payments whereas, G7 delivered only five (5) systems and had only one commercial customer. Therefore, G4 is selected to study in depth as a case study.

From G4 research group, interviews are carried out with three (3) senior academics, two (2) research fellows worked from the very beginning, two (2) past interns who later worked at the government organization through industry partnership, five (5) recent interns, and the current project coordinator, once a month throughout a year (from Jan-Dec, 2014). The authors had access to all websites, published papers, research and commercial project proposals, funding applications, internship and employment agreements, and systems developed to view and gather non-confidential data. The confidential data such as persons' and organizations' names are anonymised throughout the report. The authors observed the work environment and participants while interviewing them throughout the year. Case study report is formulated based on the gathered data (5.2. A case study from a Sri Lankan university, page 89).

3.5. Generalising from Qualitative Research

Generalizability is a common critic of qualitative research studies (Sarker et al. 2012). Therefore there are many studies about how to generalize qualitative findings (Lee, Allen S and Baskerville 2003; Yin 2014; Sarker et al. 2012). According to Walsham (1995) there are four concept of generalization: 1) development of concepts, 2) generation of theory, 3) drawing of specific implications, and 4) contribution of rich insight. One can approach these types of generalization by stating that a) findings are not generalizable, but the theory or the concepts can be applied to other settings; b) generalizability is limited, but is balanced against other advantages of qualitative inquiry; c) future research is needed to enhance generalizability (Sarker et al. 2012). For example, the theory of evolution by natural selection, is still evolved with new species as lifespan of some species are not observable by only one researcher throughout his lifetime (Smith 1987), likewise, it is impractical to address all possible context of a study by only one research. Therefore, in this study the authors approach generalizability by stating that other researchers can apply the developed framework and concepts in other settings after deductively evaluating for the new context. The way they approach it and make the research global is explained subsequently in this section.

At the beginning of this research GTA is focused on generating themes. At the third stage of analysis (i.e., selective coding) theoretical sampling strategies (Glaser & Strauss 2006) are incorporated in order to make the study globally accepted by application of existing theories (Goldkuhl, Göran and Cronholm 2003). Generalisation is performed as the GTA design has been appropriate informed by theory, and the established theory is added to the findings (Glaser & Strauss 1967). The method of generalisation for qualitative research is not statistical generalisation, but analytical generalisation in which a previous knowledge is incorporated with the empirical results of the case study (Scott & Howell 2008; Lee, Allen S and Baskerville 2003). Furthermore, the authors have included two cases based on secondary data sources in the analysis of this study. Therefore, the framework is generalizable to some extent, hence, the framework and concepts can be applied to other contexts.

Chapter 4. Analysis and results

Literature review showed that previous knowledge on managing research based software development in universities is lacking. Nevertheless, recently emerged software development methodologies; Agile methods or FOSS model, may be able to apply in these projects. To find out if these methods can be applied in the university context, semi-structured interviews are conducted with practitioners from industry projects and university projects in Sri Lanka. The objective was to understand similarities and differences in software development practices between university research projects and industry projects.

4.1. Analysis between industry projects and university projects

The analysis is done under number of aspects related to inputs, processes, and outputs of software development projects and research projects identified through the literature review. Other important facts and uncertain factors are also discussed at the end (Table 3-10). Communication methods and meetings are addressed within other aspects when applicable. These aspects are assessed with regard to software industry and university contexts by using the data collected from twenty eight (28) respondents from twenty eight (28) projects. During the analysis, it was noticed that the projects from a same company have similar characteristics, thus we discussed them collectively as a company. University projects are mostly different from one to another, thus they are discussed as individual projects.

1. Requirement for research and product development

Most of the industry projects started based on client requirements. Only two of them started based on market trends and customers have been found later. When a client requested a system, a formal agreement was signed between two parties. University projects have started based on community problems, problems raised by corporate customers, university's internal problems that lead to research and software development. Formal agreements were not found in any university projects. A graphical representation of requirement for research and product development types in industry and universities are given in Table 4-1. It is understood that industry projects had certain promise on their investments than university projects because of formal agreements.

Table 4-1: Requirement for research and product development in industry and universities

Requirement for research and product development														
Industry (N=14)	C1			C2		C3		C4		C5		C6	C7	C8
Client requirements	■	■	■	■	■	■		■	■	■	■	■		■
Market trends							■						■	
University (N=14)	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	B7	B8
Customer problems		■		■		■						■	■	■
Community problems			■		■			■	■	■	■			
University's internal problems	■						■							

2. Goals and objectives

All the companies are focused on revenue on investments for providing products and services to their customers. Universities are mainly focused on research and learning, while financial benefits are secondary. Although researchers generated income through commercial activities, most of the time excessive funds are used to acquire resources and retain people for R&D.

3. Financing and resources

All the companies financed through client payments, although initial investments are beared by investors or company owners. The investors have funded at the beginning to build required infrastructure and resources in companies. Therefore, project teams are well equipped to carry out projects. In return the investors are paid return on their investments. All the university projects have got some kind of a research grant at the beginning or middle of the project. The very basic infrastructure facilities are provided by the universities to initiate research projects. Research grants are used to get resources and pay salaries for researchers and developers. Some projects are funded by industry companies.

Table 4-2: Differences in financing and resources between industry and university projects

Financing and resources														
Industry (N=14)	C1			C2		C3		C4		C5		C6	C7	C8
Investments & income	■	■	■	■	■	■	■	■	■	■	■	■	■	■
University (N=14)	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	B7	B8
Research grants	■	■	■		■	■	■	■	■	■	■	■	■	■
Industry support		■		■	■	■		■	■	■	■	■	■	■
Customer payments		■		■	■	■		■	■	■	■	■		

One (1) of the projects is fully sponsored by a private company. They did not have any resource or funding shortages. It is identified that the universities faced more challenges than the industry. Industry projects had considerable investments at the beginning and predictable income because of formal agreements. Most of the university projects had only few short term funding sources. From Table 4-2 it is visible that all the companies have better financials than universities.

4. Project team

Software development teams in companies mostly consisted of a project manager, architects, software developers, business analysts, and QA engineers. Database administrators, UI engineers, and network administrators played supporting roles whenever necessary. Some teams had a position called technical writer instead of business analyst. C5, the Scrum practiced company, had a Scrum master, product owner, and a QA lead, instead of project manager and business analysts. C8’s team had a sales person as well.

University teams consisted of Phil/PhD students who do research and software development, and supervisors who work as senior lecturers at universities. Some teams had research assistants, interns as well as software developers to do research and software development work under the supervision of senior researchers. Team A4, that is sponsored by a corporate company, had a director to overlook the entity, however, there were no research students in that team. Team B6 was distributed team consisted of researchers, supervisors, software developers and interns from different countries such as Sri Lanka, Australia, UK, and Italy.

Table 4-3: Project teams in industry and university projects

Project teams														
Industry (N=14)	C1			C2		C3		C4		C5		C6	C7	C8
Project manager														
Architects														
Software developers														
Business analysts														
QA engineers														
Database administrators														
UI engineers														
Network administrators														
Technical writer														
Product owner														
Scrum master														
QA lead														
Sales person														
University (N=14)	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	B7	B8
Research student (MPhil/PhD/MSc)														
Software developing														
Lecturing														
Investigators														
Lecturing														
Research assistants														
Software developing														
Interns														
Software developers														

Director														
Lecturing														
Consultant														

Project wise team details are shown in Table 4-3. Full time workers are shaded in dark grey colour whereas, part time work is shaded in light grey colour. In industry database administrators, UI engineers and network administrators are full time employees in the company, but, working only for some parts in a project and change projects frequently. In universities part time employments are shown in italic letters. For example; research students were doing software development as a part time job in A1, A2, A3, A5, B4, B7 and B8 projects. Main difference between university and industry teams is there were no dedicated people for research in the industry. Industry practitioners have been doing full time product development work, while some experimentation is done to find new technologies. The time they spent on research is very low compared to university teams.

5. Software development methodologies

C1, C2, C4, and C7 companies used a combination of Agile and Waterfall methods. They were going through the steps of Waterfall method in each Agile iteration. C3, and C6 were using Agile practices. C5 used Agile based Scrum. Therefore, except C8 all the other companies used some kind of an Agile method. C8 used Waterfall method. The participant from C8 mentioned that they developed relatively small mobile applications, thus Waterfall phases suited well. Two teams from university A namely; A2, and A5 used an evolutionary approach as their software development methodology. Team A6 used iterative prototyping. Team B6, which had a distributed team, used an incremental development approach based on design science research and action research. All the other respondents stated that they did not use any particular software development methodology, but some self-constructed methods.

Most of the software companies practiced Agile methods to manage software development processes and teams. Although Agile methodology are specifically used to manage software projects, university teams have not used it. Later, when the authors asked why they do not use Agile practices, the answer was Agile is suitable for face paced development projects with fairly defined requirements and predictable outcomes. It was noted that existing software development methods are rarely used in university projects, because of their unpredictable nature.

6. Project life cycle phases

The phases of traditional Waterfall method; planning, requirement analysis, design, development, testing, implementation, and maintenance, are followed by many industry practitioners. Some companies followed these phases iteratively within Agile approach. Company C, and F had a high level requirement gathering first and estimations and project plans were prepared with that information. Then detailed requirement gathering, design, development, testing, and implementation are done in several iterations. After final implementation of the complete solution continuous maintenance is carried out. Company E that practiced Scrum method followed the standard Scrum life cycle.

Most of the teams from universities had a similar life cycle to Waterfall method to develop software. The differences are identified as; they did a literature review, before requirement analysis, and instead of testing, they did an evaluation to critically assess the system. However, none of the university teams followed Waterfall steps in the sequential manner. They practiced the steps randomly when it is needed. It is a weakness, that Except A4, other teams are not much concentrated on the maintenance phase. Some teams created prototypes after literature review. Then requirement analysis is done, and kept on designing, and developing it iteratively while feedbacks are gathered from end users until they are satisfied. Then implemented the system, but maintenance phase is not mentioned. Project B-6 followed a totally different methodology. They started with AR which followed; diagnosis, action planning, action taking, and evaluation. To execute the action planning phase they used DSR steps; objective identification, design and development, demonstration, and evaluation. As DSR is used for artefact development and evaluation, they argued it can be used to develop software systems.

Anyhow, all the companies followed a precise life cycle for every project they perform, whereas universities adjusted their practices and came up with new methods time to time, depending on their research possibilities.

7. Project milestones

To the Scrum practiced company (C5), there was a project milestone at the end of each Sprint. They delivered a working product after each Sprint. The time duration of a Sprint was a month. This was similar in all Agile methods. Agile practiced companies had daily and weekly milestones to review their work internally and monthly with clients. They selected a component of a full product which can be completed within a month. Then divided weekly and daily work accordingly within the team. There were daily weekly time bound progress review meetings, and monthly demonstration to all project stakeholders, including customers. Some companies practiced all phases of Waterfall method within Agile iterations. In Waterfall method, project milestones are as same as its life cycle phases. The scheduled milestones are set by project teams and are approved by clients after discussing the scope of a product.

Similar to private software companies, some university projects got client deadlines and their internal deadlines to meet client requirements. Except from project A4, all the other projects followed conference deadlines, as they published manuscripts. There were deadlines set by the funding organizations. Further, PhD/MPhil students had deadlines set by university regulatory for higher degrees. Project A3 had its own deadlines for publishing apps to Google play app store. Project B2, B3, B4 and B5 had exhibition deadlines as well, as they demonstrated their prototype products in public gatherings.

From the above information it was evident that industry teams were fast paced and milestone driven in order to meet customer requirements. University teams had many different predictable and non-predictable milestones as mentioned above.

8. Interactions with end users

Back to back relationships with end users are considered as important in all the companies. They used face to face meetings, emails, and phone calls as communication mechanisms to gather requirements or feedback almost every week. The entire project team maintained a close relationship with end users, starting from gap analysis for a new system, to clarify requirements, getting feedback, till troubleshooting, and continuous maintenance. However, depends on the client, level of interactivity is varied. Some companies used Skype conferences and emails to gather requirements as they had international clients. Some complex projects involved sending a business analyst to overseas sites to get clear requirements. Scrum teams are more standardized because of Sprint reviews. After each Sprint they did a product demonstration to end users and gather feedback before proceeding to next Sprint. Some companies had an onsite customer. Development team could discuss with him anytime for any reason. However, in all the Agile practiced companies, monthly reviews were there to meet project stakeholders. Informal relationships are built apart from the standard demonstrations to clearly understand and fulfil the end user needs.

Four (4) university projects had end to end relationships with end users similar companies. They used face to face meetings, Skype conferencing, email convocations, and phone calls to get end user involvement. Feedback are gathered in daily, weekly, or monthly basis. Some teams collected feedback thro' questionnaires to evaluate requirements and prototype systems. Team A3 collected feedback via Google play in each stage, as their audience was a group of public users in Google. Other teams involved with end users only in design and development stages. Project B6 engaged with end users from requirement gathering (diagnosis phase) to evaluation phase in AR. They used all the above communication methods as well as questionnaires to gather requirements and continuous feedback. Participants from projects B7 and B8 said that end user interactions were not applicable in their research projects.

From the above information it is clear that universities had less interactions with the end users compared to the industry. Industry companies had formal agreements with their customers. Universities had mostly informal partnerships with volunteers, thus there were a lot of uncertainty. This might have been a problem for university researchers when transferring their research findings to the society.

9. Relationships with community groups

External relationships with community groups were very rare in the companies except from their potential customers. Some companies had collaborations with state government and regulatory bodies. Company C, E, F, and G did not have any collaborations with external communities. Company D has given research funds for state universities, through corporate social responsibility initiatives.

Most of the community groups, the university teams collaborated informally with, were their end users. Similar to software companies some university teams had formal and informal partnerships with private organizations as their clients or potential clients. Except the project A4 all the others had relationships with funding organizations, state government and regulatory bodies. Some projects had engaged with university students as their pilot sample to get feedback from, when actual end users were hard to approach. Furthermore, team A3 collaborated with public users of Google play application. University teams discussed with other research groups, faculties, universities as well as international universities to get expert knowledge, support, and test data sets. Universities had relationships with a larger group of communities to get support in different activities. Industry companies did not have relationships with external parties except from the state government and their potential customers.

10. Design methods and programming languages

As design methods, the companies used object oriented programming (OOP), Model-View-Control (MVC) architecture and sometimes Service Oriented Architecture (SOA). As programming languages, they used Java, C++, Objective C, PHP, Python, C#, NodeJS, ASP.NET and a wide variety of basic and new programming languages depending on the product component. Universities also used OOP and MVC architecture as design methods. Java is identified as the most popular programming language among university researchers. C++, Python, Perl, PHP, C#.net, ASP.NET and Android are also been used sometimes.

Design methods were almost similar in the industry as well as the universities. It is understood that the industry developers chose from a wide variety of programming languages depending on their skills and nature of given projects. University researchers had a narrow selection of programming languages compared to the industry.

11. Outcomes

Table 4-4: Outputs in industry and university projects

Companies	**Outputs**	**University**	**Outputs**
A, G	Enterprise management systems	A1	A gaming component, publications
B, H	Web, mobile applications, backend services to three (3) industry segments	A2, A5, B1, B7, B8	Web based systems, publications
		A3	Web proxy server, publications
C	Middleware applications to seven (7) industry segments	A4	Web and mobile applications
		A6	WPF support system, publications
D, E	Web applications	B2, B3, B4, B5	Simulation systems, publications
F	Customer relationship management (CRM) systems	B6	Web and mobile application, publications

Outputs in university and industry projects are given in Table 4-4. The main difference in project outputs between universities and industry was publications. Apart from that some university projects had system components, whereas companies always developed full solutions. These components had to merge with another system to be presented as a full product, therefore problems were arisen when introducing to end users.

12. Important facts

All the respondents from companies stated end user interactions, continuous feedback and customer satisfaction were important. As other important facts, respondents from Company A have said on time delivery, accuracy, speed, security aspects of systems and continuous clarification discussion with all project stakeholders were critical. A respondent from Company C mentioned flexibility in processes was easy for the team. Respondents from Company F believed identifying correct requirement, analyse risks, time estimation and budgeting were the most critical aspects of a project.

In universities, planning, distribution of tasks, and team management is identified as an important task by all the teams. It was essential to keep team members motivated and interested towards the research, so that they would not leave because of high uncertainty. Except team A4, all the others stated publishing the work was mandatory. Except B7 and B8, other teams said end user interactions, continuous feedback, and end user satisfaction were vital. Data gathering was a major part for team B7 and B8, as they research on data driven processes. Apart from those things, managing international collaborations and use of action research and design science research together are mentioned as critical factors for team B6.

13. Problems, risks and uncertain factors

All the respondents from companies stated completing a project within allocated time and budget was always challenging and risky. Company A faced more problems with these restrictions as they had to develop bug free systems for a mission critical industry. Respondents from Company B complained that time bounds restricted the ability of developers to catch up new technologies, because complications could occur when experimenting new technologies. Apart from time and budget issues, customers failing to hold their ends of the contracts, and competitors are identified as risk factors to Company C. Quick changes of product milestones based on market trends is identified as an uncertain factor in Company E. The respondent from Company H has identified several risk factors including; conflicts between users and developers, users with negative attitudes toward a project, continuously changing requirements, involvement of new technologies, change in organizational management during a project, and unstable organizational environment.

Lack of funds, resources and access to publication libraries were the main issues faced by university teams. Time was vital, because research could be outdated very fast and had to meet deadlines from universities and funding agencies. Delays on releasing funds by donor agencies, made it hard for researchers to carry out research work as they planned. Sometimes, experienced people left projects by the time funds were released. Lack of knowledge in high tech areas was an issue with novice researchers. Keeping end user interest, interactions and acceptance were always at a risk, because their participation was voluntary. Some teams had restrictions on publishing their work because of confidentiality problems raised by industry partnerships or commercial projects. Scarcity of test data was a critical issue in some research projects. Most of the products developed by researchers are not maintained properly even if they are deployed satisfactorily.

4.1.1. Overall similarities and differences

From the above analysis it is understood that only design methods and programming languages had similar attributes in university and industry projects. All the other factors had slight or major differences.

University research was more towards community problems, whereas companies had particular customers with formal agreements. Companies focused on revenue generation and customer satisfaction. University researchers are interested in generating new knowledge, techniques or creating inventions. Economic development was secondary. For industry projects, required budgets, people, and resources are allocated using capital investments, at the beginning. Research projects did not have direct investments. Researchers needed to convince some external funding agency or partner to invest in their projects. Therefore, university projects had a lot of uncertainty. People on those projects needed to make a considerable effort to be motivated and keep interest towards research and innovation.

Industry teams are well defined with their roles and responsibilities, whereas university teams are dynamic. Researchers change their duties time to time, to do research and software development. There was a tendency of people leaving from university projects because of short term agreements and job uncertainty. Moreover, industry practitioners are used some kind of a software development methodology, such as Agile or Waterfall. Most of the university researchers did not practice any defined methodology. They chose different stages of the software development life cycle whenever suitable and perform research or development tasks. In the industry, project life cycle stages and milestones are aligned with the software development methodology they practice. Milestones are decided after discussing with the customer. Therefore, most of the management aspects of a commercial software development project is covered by the software development methodology they use. Management of end user interactions and outcomes are also included in Agile methods. University projects had various milestones, based on their regulations, funding agencies as well as customers. They developed a range of outputs, for examples; research publications, dissertations, posters, exhibits, prototypes and software products. Most of the time end user interactions are limited as they are not bounded by agreements, but voluntary.

As the authors figured out during interviews, university researchers were not capable of using well defined software development methodologies, because of the uncertainty nature of research goals, funding, project teams, project lifecycle, milestones, end user interactions, and research outcomes. It was difficult to manage research based software development projects in universities, by using the existing software development methodologies. Therefore, the authors decided to investigate on unique challenges faced by university researchers in order to find suitable strategies and best management practices.

4.2. Challenges faced by academics when managing research based software development projects

Research based software development is considered as complex due to several factors. Researchers needed more time and freedom for experimentation. Predicting and developing management methodologies for research projects are critically difficult because of distinctive challenges occurred due to uncertainty and complexity. Therefore, it was important to identify those challenges before investigating best practices. The challenges identified through content analysis of data collected from university participants (Table 3-8) are explained in this section. The authors did not pinpoint each project when discussing challenges, because some researchers who were not try to overcome the challenges might found it at fault, thus the participants' confidentiality is protected.

1. People retention

With immense uncertainty in research projects, it was hard for many researchers to thrive a long time in this field. Most of the senior academics from Sri Lankan universities complained, it was difficult to find committed students or research assistants to undertake fulltime research work. It was more difficult since Sri Lanka is a developing country. The senior academics witnessed a tendency of young people who are interested in research went overseas for higher studies as soon as they got a little exposure. Other people left for better jobs because of low wedges, facilities and job insecurity in research units. Some students have struggled to complete research projects, mostly when they selected a topic out of their interested areas. According to interviews, lack of interest has become a serious issue because of too much freedom and less restrictions in research projects. Conversely, supervisors faced difficulties when they select inappropriate and uninterested students to run their research. Therefore, people retention is identified as a serious challenge in Sri Lankan research groups.

2. Lack of funds and economic uncertainty

Research projects do not usually get direct investments. Researchers need to write proposals for funding agencies, based on their conceptual thinking. It is difficult to convince investors or funding agencies because of the vagueness of research ideas at the beginning. Although some students start research without funding, there is a greater tendency to terminate those projects in the middle, than projects with funding. Researchers need funding for various activities, other than a regular stipend. Without financials it is not possible to get resources or people to carry on research. If manuscripts get accepted, researchers need funds to participate on local and international conferences. It is better to have some kind of a funding source to start the research with. Then it is required to have a better cash flow in order to expand and get more resources. Economic uncertainty is a reason for people to leave research projects. Lack of funds and economic uncertainty is identified by many academics as a factor for giving up on research projects.

3. Access restrictions to publication libraries

This is a major issue faced by many students when doing research in Sri Lankan universities. It is a basic facility provided by most of the higher education institutions around the world. It is not an established facility in Sri Lanka yet. All the researchers face difficulties in getting new knowledge to do systematic reviews. Some students get help from their research partners or friends in overseas universities to find materials. Other students only use open access publications, as they cannot afford individual membership fees. This is a problem raised by all the research students, thus can be considered as a countrywide problem to students in all the Sri Lankan universities, irrespective of their area of studies.

4. Late plans on product development

Most researchers do not plan from the beginning to implement working products based on their research outcomes. Once they have a technically proved conceptual solution only they plan on product development. There are various things that should be concerned about from the beginning, for implementations. It is important to get requirements of the end user at least few times, because technically perfect solutions might not work in practical situations. Thus, before developing technical solutions, it is needed to assess actual problems by engaging with people who face it. Some researchers have just conceptually assessed the problems and developed solutions. Later, when they try to implement, some practical problems arose that lead to project termination. Another issue is funding and resource requirements for implementation. It was noticed that funded projects had both research and long term product development aspects in their research proposals. Thus, not planning for a productive outcome can lead to less attractive funding proposals as well. Later plans for product development can lead to development of incompatible solutions and project terminations.

5. Getting end user interactions

Unlike for industry projects, it is very hard to get end user support for university projects. Many researchers complained that they could not talk to end users as much as needed. Since end users are not bounded by any agreements, it is difficult to get their time with regular work. Although university researchers try to help them free of charge, because of the uncertainty of research projects, end users may not believe that they would benefit. Thus less supportive. Sometimes they do not give all the information because of confidentiality issues. This leads to poor design of the solution. Therefore, the authors find difficulties of getting end user interactions as a problem in university projects.

6. Problems due to industry partnerships

Some industry partnerships that was started with the intention of doing research, have later become fully commercialised entities. Universities do not have much control over the work they propose to do because of the agreements they signed. When a research unit is fully funded by a private company, it is dedicated to do any software development work they propose. Researchers are not able to choose between projects, even if the projects are not research oriented. Even though some projects have research components, companies do not allow publishing because of confidentiality issues. These projects have deadlines given by the customer, thus restricting the freedom for innovations. Some companies fight for product and patent ownerships after a while they have been supported. Hence, there are problems arose because of conflict of interest in industry partnerships.

7. Problems with funding agencies

Many researchers from Sri Lankan universities complain about long and time consuming procedures of government funding organizations. Some researchers stated that they had completed a half of the project, by the time funds are released. Research assistants spend half of their time on filling forms and getting approval for purchases. The authors witnessed few projects that the researchers had given up because of funding shortages, by the time funds approved. A proper infrastructure for research funding is yet to establish. However, some professors stated now there services are much efficient than in the past. Yet, less efficiency and legacy procedures in funding organizations is identified as a problem for research groups in Sri Lanka.

8. Product maintenance issues raised by users

Most researchers did not mention maintenance as a phase in the software development life cycle they use. This is a reason for lacking end user trust and support for university projects. It is seen that researchers tend to leave projects once they completed the work. There are only few established research groups that do regular upgrades and maintenance. Without maintenance, users might get more trouble than in manual systems. If help is not available when they need, they might as well stop using the product. Lack of continuous maintenance is identified as a problem for not using fully implemented software products.

Above challenges are linked to each other in several ways. Economic uncertainty is a reason for difficulties in people retention. Late plans on software development lead to delays in getting end user attention and arrangements of funding. End user support is lacking due to uncertainty of product implementation and maintenance. However, it is required to find solutions to overcome these challenges in order to increase the number of long term sustainable and productive research groups in Sri Lankan universities.

4.3. Best practices in managing research based software product development in universities

Drawing from the study, this section summarizes best practices for research supervisors and researchers in managing software development research groups in Sri Lankan universities. It is important to have a synergy between these two parties to create a flourishing bond. Further, two strategies are proposed to universities as a whole and the regulatory. The authors suggest that use of these strategies can improve productivity, continuity and stability of a research group while creating opportunities to do more research and product development.

4.3.1. To research supervisors

1. Awareness of attractive research happening in their field

A researcher advisor's responsibility is to be updated about local and international knowledge and recurring problems. Compared to Kroeze et al. (2010) study, Sri Lankan research supervisors are more updated and supportive. As the authors observed, supervisors of more sustainable research groups are already good at supervision and creating a research culture by participating in conferences, organizing workshops and other proposed strategies for South African universities (Kroeze et al. 2010). They do not only participate in research conferences, but also share knowledge and experience among fellow researchers. Active researchers are alerted on attractive research happening in the field. Not only community needs (Guyette 1983), but also government and industrial problems are taken into consideration. For examples; tender calls, government cassettes, newspaper articles, and other sources of general information are used to identify day today problems. By doing so, they can motivate junior researchers to find and solve interesting problems as well as grab the attention of end user communities.

2. Community building

As mentioned by previous researchers, networking with practitioners from different disciplines, and expertise, result in innovative ideas which extend the research capacity (Hargadon 2003; Cohen & Bailey 1997; Numprasertchai & Igel 2003; Levén et al. 2014; Geenhuizen 2011). From this study, the authors found more benefits that can be drawn from community building. Established research groups like G4, G7, G8 and G10 are collaborated with external communities and private companies who interest in their field of research. Linux is built entirely by a community, because of Linus's talent as a community leader (Raymond 2001). Once a senior academic got an idea to form a research group, first he could talk with fellow academics with similar interests. These fellow academics can be found from same faculty, outside faculties, other local or international research institutions. Through mutual collaborations, they can spread the word among junior researchers who would like to take on research in the field. In addition to innovative ideas, research supervisors could find motivated individuals to form a group, as well as get investments and commercialisation opportunities, by building relationships with right communities. Therefore, research supervisor's responsibility is to initiate collaborations with suitable communities and encourage the group to uphold them long term.

3. Work together with established R&D centres in the same university

Any research needs equipment and finances to pay research allowances. Along with community building, research supervisors should look for possible funding sources. From the partnerships mentioned in Table 2-2; industry partnerships, government partnerships and networking with industry practitioners are used to raise funds in Sri Lanka. As the authors found during this study, there are several other ways of acquiring funds for research in Sri Lankan universities. The easiest and most practiced way of starting a new research group in Sri Lankan universities is; first work closely with an established centre or another research group which has common interests. The establish group or centre can help by sharing some equipment and funds, while the new group members can do part time research, development work and shared publications in return. That was practiced by G2, G4, G5 and G8 research group. In Google's case Page and Brin (Vise & Malseed 2008) had some help from internal centres of their university at the beginning. Linus also started developing Linux when he attached to another project called HURD, while he was a student at Helsinki University (Raymond 2001). Until a research group makes some credibility to function on its own, acquiring support from other entities can be the best possible way to start with.

4. Gather a group of passionate people

Almost all the senior academics who participated in this study, has gathered their fellow researchers through personal contacts, no formal interviews conducted. If a principal investigator first built collaborations with fellow senior academics, they help in finding junior fellows from external institutions. That way, a set of passionate and skilful junior researchers can be found to work as research students or assistants or product developers in a software development research group. More than other groups, G7 participants specifically explained how they joined and why they were still willing to be a part of the group, although there was a great amount of uncertainty. From their own words;

> "*First I heard about a research group, which is going to form, from a senior lecture of my faculty, when I was looking for jobs after graduation. I got some employment opportunities, but I was much interested in this research. So I decided to join and see. Since then I have worked more than 10 years with the group. After sometime, our principal advisor suggested to register for an MPhil in order to have an academic career. I completed it after struggling for more than 7 long years, but did not leave the research group. Today I am a senior lecturer in this university, and still enjoy working in the research group voluntarily, whenever I get some free time during work.* "

From this statement and more similar thoughts from researchers, it is evident that a passionate researcher would not leave the group irrespective of how hard it can be. A good research supervisor could be able to identify and draw together such personalities through experience.

5. Prompt an inspirational leader

Research professors role as a team leader (Etzkowitz 2003), was not practical in Sri Lankan universities because of lecturing and supervision workload. Although in other countries supervisors has been managing organizational activities virtually full time, in this study the authors observed a naturally occurred team leader amongst fellow researchers. Similar to Linus Torvalds's Linux revolutionary (Torvalds & Diamond 2001), research oriented projects started with personal interest of a researcher or a set of researchers. Except from G5 and G9, all the other research groups on this study had a person who has become a leader eventually, amongst others, when the group evolves. Unlike in industry projects, there is no appointed project manager role in these projects. Most of the time, this headship role is taken by a PhD student. Commitment for the PhD and interest towards research motivated them to find ways to achieve their targets. Therefore, research supervisors could identify such people, and guide them towards leadership. In the studied examples, with the guidance of supervisors, these leaders have kept the group together. If research supervisors can find at least one person in a group, who has enough confidence to take that inspirational role, the group is more likely to sustain long term. Here is it also important to remember the Raymond's lesson, if the leader lose interest, the position has to be passed to someone interested (Raymond 2001). A set of good assistants guided by an inspirational leader can direct the group towards getting more research funds and publications, community building, industry partnerships and developing commercial products.

6. Stimulate an overall R&D culture

In literature it is mentioned that research professors make an effort to build research culture in universities through their experience (Kroeze et al. 2010; Gulbrandsen, Magnus and Smeby 2005). Through this study, the authors explained how to do it practically within universities. As the authors observed, supervisors of more sustainable research groups are actively engage with the team routinely. The supervisors do not only supervise but also conduct research with the group. They look for funding opportunities regularly and collectively write research proposals. Brainstorming and review meetings are conducted weekly. Research seminars, workshops, and other collaborative activities are organized in order to gather interested communities, create awareness and share knowledge gained through research. They write publications, participate on conferences and exhibitions, whilst encouraging fellow researchers to do the same. Supervisors of more productive research groups, have communicated the long term product development goals from the beginning of the projects. Therefore, researchers and developers are concerned on end user satisfaction as well, apart from technical viability. When compared to countries like Australia (Commonwealth 2007; Commonwealth 2011), research commercialisation is not given much attention. Through this study, the authors suggest senior academics to create the culture within their universities by actively participate in R&D activities.

The research supervisor's role and responsibilities are concluded with one of the Raymond's sayings on FOSS community building;

> "*This should be obvious. In order to build a development community, you need to attract people, interest them in what you're doing, and keep them happy about the amount of work they're doing. Technical sizzle will go a long way towards accomplishing this, but it's far from the whole story. The personality you project matters, too.*"
>
> - *Raymond, 2001*

4.3.2. To research groups

1. Having a long term vision, goals and objectives

They have long term higher level objectives of the final outcomes. They do not have direct commercial purposes, time bound deliverables with specific requirements, and confidentiality issues. For large scale development projects, researchers who are interested in the area can join to R&D a part of the solution. Therefore, research projects grow naturally with incoming contributions, while providing learning opportunities to undergraduate and postgraduate research students. In Raymond's words a "*plausible promise*" can catch the attention of a community.

2. Embracing the uncertainty and changes

In academia, once G4 changed the entire project architecture because of an idea came through a student research project. G4, G5, G7 and G8 have been changing and expanding the project scope time to time, to increase research capacity to take new research students. In POP3 mail project, after some development Raymond understood that his interpretation of the problem was wrong, thus the solution wasn't compatible (Raymond 2001). This could happen to every researcher, the moment he found the right question; he realised the right answer with no time. However, none of the above groups stopped research or product development because of sudden and major changes. In fact, the changes have improved the solutions and fastened the product development. The established research groups have believed that changes can make better contributions to the product as well as the research community. Therefore, they do not reluctant to change in the future as well, by embracing what went wrong as lessons.

3. Dividing large research problems into meaningful and manageable components

This practice is adopted from the Agile practices. Most of the research problems could not handle by an individual research students. In some projects the scope was not decided at the very beginning because of poor planning and management. Trying to do it all by the main researcher himself without proper planning had jeopardise the whole project in some cases. If the group could measure the scope of the research at the beginning, it can be divided into manageable chucks. By giving manageable research components to research students, established research groups have managed to achieve a complete research solution at the end.

4. Acquiring several funding sources

In literature review, partnerships with government and industrial firms, networking with practitioners, technology transfer offices (TTO), alumni groups, entrepreneurship, and spin-offs are explained as different ways of getting funds and managing a research entity (Hashimoto et al. 1997; Siegel et al. 2003; Guerrero & Urbano 2012; Perkmann et al. 2013; Tartari et al. 2014). Since technology transfer offices (TTO), alumni groups, entrepreneurship, and spin-offs are not yet developed in Sri Lanka, researchers themselves have to find several funding sources in order to sustain long term. Through this study is it found that partnerships with government firms are popular than industry firms in Sri Lanka. It is observed that government research agencies have utilized their funds on projects that address country’s development. Even though the government provided funds for research, practitioners faced difficulties in getting them because of long and legacy procedures, as explained in the previous section, under challenge number 7. To avoid that established research groups have collected funds through many initiatives. They have written different funding proposals to various agencies by using divided research components. Some researchers including Page and Brin (Vise & Malseed 2008) have provided services to the university itself, and acquire funds and equipment to expand research capacity.

5. Undertaking short term commercial software development projects in their research area

Main success factor of a research group is identified as committed human capital. During this study the authors came through some projects that have useful research findings, but had failed to develop or implement any solutions, due to knowledgeable and experienced people leaving the project. Enthusiasm and interest can keep a person bind to a research group for some time. However, after a while researchers tend to leave, because of financial issues and uncertainty. Successful research groups uses income from above mentioned commercial projects as a strategy for employee retention and resource issues. These financial benefits give hope and keep researchers for a long period of time. Researchers work in both research and commercial projects simultaneously. Salaries are paid mostly from the income of commercial projects. Sometimes everyone works on commercial projects if there are strict deadlines. However, they try to balance the time spend in research based projects and totally commercial projects. Although main objective of university projects is research and learning, it is understood that commercial projects are important to long term existence of research groups. That way, time restrictions for research can be loosening, and more freedom could be granted for experimenting new technologies. Overall uncertainty can be decreased and confidence can be increased by developing software products using the acquired skills through research done.

6. Keeping up community relationships

Keeping up continuous relationships with the communities are considered important as researchers were lacking end user support. They got end user support mostly from the communities they were engaged in. According to Numprasertchai & Igel (2003) trust, commitment, and balanced mutual benefits among collaborators are the main success factors in partnerships. Most of the established research groups in the universities had strong and long term mutual relationships with relevant community groups.

7. Support new research initiatives in the university

Supporting new research initiatives is considered a good practice by established research groups, because of their past experiences. That way overall research culture in Sri Lankan universities can be increased by supporting each other’s activities.

8. Outsource maintenance to commercial software development entities attached to the university

To solve the long term maintenance problems raised by end users many Sri Lankan research groups have outsourced those activities to established commercial software development centres attached to universities. Some fulltime employees from those units are trained by the researchers to handle product upgrades and continuous maintenance. Most of the research groups have been working together with these units (G3 and G6 in two selected universities) when developing software products, thus knowledge sharing happened from the beginning.

4.3.3. To universities and the regulatory

Through this study, the authors strongly recommend to Sri Lankan universities and the regulatory to get institution wide access to scholarly research publication libraries. Moreover, they can organize events with the software industry to increase productive research through mutual partnerships.

4.4. Framework

Following framework is derived based on the above findings on considerations to research supervisors (Table 4-5), researchers (

Table 4-6) and regulatory (Table 4-7).

Table 4-5: Framework based on considerations to research supervisors

Considerations to supervisors	Description and examples
Research awareness	It is important to create awareness of interesting research happening in their field in order to attract good students and communities.
Community building	Building a community with other faculties, industry, government, overseas researchers and other related parties to generate ideas, find people and funds. Keeping up community relationships maintains a wide voluntary user base throughout the project.
Work collaboratively	Work together with established R&D centers in same university which have common interests helps through shared equipment and funds, while the new group members can do part time R&D work in return.
Recruiting passionate people	A group of passionate researchers tend to remain during hard times because of the passion for research. New research fellows are

	interviewed (informal talking) for their interest for research interest and job satisfaction, instead of salary and job security.
Strong leadership	Prompt an inspirational leader within the research group. A strong and continuous leadership throughout a research project is identified as a success factor. Good leadership can keep a research group together for a long time.
R&D culture	An overall R&D culture can be stimulated by encouraging publications and organizing collaborative events with communities and industry. Hence research groups get required funds, resources and industry exposure.

Table 4-6: Framework based on considerations to research groups

Considerations to **research groups**	Description and examples
Clear objectives	Having a **long term vision**, **goals** and **objectives** is important in order to cope with uncertainties. Specially about product development and commercialization.
Embracing uncertainty	**Embrace uncertainty** and **changes**. Apply changes to developed products or designs in order to have **innovative outcomes**.
Division of research ideas	**Divide large research problems** into **meaningful** and **manageable components**. Hence collectively create a complete solution.
Several funding sources	Good to have projects run by **government funding** as well as **industry partnerships**.
Commercial software projects	Undertaking **short term commercial software development projects** in their research area bring a **better income** for research projects. **New knowledge** acquired from research can be **applied** in such projects.
Maintain community relationships	**Keeping up close relationships with communities** is considered important as researchers lack **end user support**. They get end user support mostly from the communities they engage in.
Support new research initiatives	Support new **research initiatives** in same university as a return of favor. Overall research culture in a university can be increased by **supporting each other'**s activities.
Outsource	Product maintenance can be outsourced to **commercial software**

maintenance	**development entities** or **other research groups** attached to university if a research group is not capable by themselves.

Table 4-7: Framework based on considerations to regulatory

Considerations to **regulatory**	Description and examples
Publication access	Getting **institution wide access** to research publication libraries.
Inspire industry partnerships	Research events can be organized with the industry to increase productive research through **collaborations** and p**artnerships**. That way industry brings more **funds** to research groups.

The framework (Table 4-5, Table 4-6, Table 4-7) is divided into three main segments based on the actors who involve in the mentioned activities. First columns of the tables give a description to the proposed factors to university supervisors, research groups and regulatory bodies. Second columns explains the ways of those activities are achieved with examples and their importance.

Chapter 5. Evaluations

5.1. Scorecard results

Table 5-1: Scorecard results for challenges faced by research groups

	Challenges	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
1	People retention	√	√	√	√	√	√	√	√	√	√
2	Lack of funds and economic uncertainty	√	√		√			√	√	√	√
3	Access restrictions to publication libraries	√	√	√	√	√	√	√	√	√	√
4	Late plans on product development	√	√							√	
5	Getting end user interactions	√	√		√			√	√	√	√
6	Problems due to industry partnerships			√	√		√				
7	Problems with funding agencies		√		√			√		√	√
8	Products maintenance issues raised by users	√	√		√	√		√		√	√

From the results in Table 5-1 it is understood that people retention and access to publication libraries are the most critical problems faced by almost all the research groups participated in this study. Lack of funds and economic uncertainty was a critical challenge to all the research initiatives, which did not have continuous industry support. Complaints on poor product maintenance is also identified as a significant problem. Getting end user interactions was an issue for many researchers as it was voluntarily and no formal agreements or promises were made. Except in G5 and G8, other groups which had industry partnerships, were restricted for publications the findings came through those projects, because of confidentiality issues. Other issues faced by research groups can be identified as poor planning and problems due to long procedures in funding agencies.

The best practices proposed to minimize those challenges, are also evaluated using scorecards. Those results are illustrated in Table 5-2 and Table 5-3.

Table 5-2: Scorecard results for best practices used by research groups and suggestions to universities

	Best practices - by successful research supervisors	G4	G5	G7	G8	G10
1	Awareness of attractive research happening in their field	√	√	√	√	√
2	Community building	√	√	√	√	√
3	Work together with established R&D centres in the same university	√	√	√		√
4	Gather a group of passionate people	√	√	√	√	√
5	Prompt an inspirational leader	√		√	√	√
6	Stimulate an overall R&D culture	√	√	√	√	√
	Best practices - suggested to universities					
7	Getting institution wide access to scholarly research publication libraries	√	√	√	√	√
8	Organize events with the industry	√	√	√	√	√

It is visible in Table 5-2, except from 3rd and 5th best practices all the others are validated by all the members from established research groups. That means the supervisors in those projects have been used those practices. Not only the participants in scorecards, but all the researchers from Sri Lankan universities, including the authors of this study, strongly argued it is a best practice to get institution wide access to scholarly research publication libraries. Universities could help research development by organizing collaborative activities with the industry as well.

Table 5-3: Scorecard results for best practices used by established research groups

	Best practices - by established research groups	G4	G5	G7	G8	G10
1	Having a long term vision, goals and objectives	√	√	√	√	√
2	Embracing the uncertainty and changes	√	√	√	√	√
3	Dividing large research problems into meaningful and manageable components	√	√	√	√	√
4	Acquiring several funding sources	√		√	√	√
5	Undertaking short term commercial software development projects in their research area	√	√	√	√	
6	Keeping up community relationships	√	√	√	√	√
7	Support new research initiatives in the university	√	√	√	√	
8	Outsource maintenance to commercial software development entities attached to the university	√	√	√		√

It is viewed in Table 5-3, that G5 did not acquire several funding sources because they are funded by an industry organization continuously, without making any restrictions on research activities or publications. G10 have not yet undertaken any commercial projects or supported new research initiatives in the university, because they also got stabilised recently. Long term product maintenance were outsourced to commercial entities like G3 and G6 by all the groups except G8. They have done maintenance themselves and earned profits. All the other best practices are validated has been used by the established research groups in Sri Lankan universities.

5.2. A case study from a Sri Lankan university

Recently, the way of doing research has been modernised, thus research groups act more collaboratively and exposed to economic activities compared to traditional research groups. They try to be financially independent in order to sustain long term as an independent entity within universities, while not fully depend on government funds. Yet, balancing the research and financial gain has become a challenge. The authors found out there are fully commercialised entities which originally founded as research oriented systems development groups in universities. In contrast, there are extraordinary researchers who had not considered any financial gain or productive outcomes through research. This case study provides up-close information about a balanced research group in a Sri Lankan university, in the aspects of research and economic activities.

The modelling and simulation group was initially formed in 2008 to work on a ship handling simulator by a group of academics in a Sri Lankan university. Computer based ship simulators are popular, cost effective and safe method which can be used for training of navel marine engineers and officers. The product was originally requested by a Sri Lankan government organization and later cancelled due to various reasons. However, one of the fellow researchers of the group started a MPhil project, thus continued to develop the product as an academic research project. This joint research effort involved academics from several faculties in the university and had informal relationships with the government organization who would like to consume the product. Consequently, the modelling and simulation group expanded and currently carries out various research and development activities for and with national and international organizations.

The research group has been doing both research and commercial projects simultaneously. Throughout the seven (7) years from 2008 to 2015, they have successfully implemented and maintaining one research based software product and six (6) commercial products. The research based product had number of components developed by many researchers all through the years which combined together to make a complete working product. Except few members, including the initiated student who had completed his PhD recently, other researchers are temporary stakeholders of the project. Software developers and interns are hired time to time for product development activities. Commercial projects which run for 1-2 years are in the same research area. The group makes use of the knowledge and skills gained through research in these commercial projects. Financials came from them helped in obtaining resources and employee retention for research product development, since research grants were not sufficient. Likewise, they have managed research and commercialisation activities while taking advantages from the benefits of each other. After seven (7) years of research and development, they have recently deployed the research product at a user site, and maintaining it with regular updates.

The report discusses insightful information on the research group's activities using question answer format (Yin, 2014, Chap. 6, pp. 185). These questions cover throughout the research project from its initiation to successful implementation, whilst, commercial projects and activities are also discussed with greater information.

1. How was the research group initiated?

A group of academics from the university collaboratively formulated a project proposal for a tender called by the government organization. The tender was cancelled due to the criticality of war at that time. However, similar to Linux accidental revolutionary (Raymond 2001), a fellow member from the tender project team started an MPhil research, later continued as PhD, based on the submitted proposal, as he was interested in the modelling and simulation area. To start with, a senior academic encouraged him to send a manuscript to a conference held in Singapore. The accepted publication did not only keep the group together, but speeded up the MPhil registration, in order to participate in the conference as a student. This validates the findings on research supervisor's role to identify an inspired student, encourage and provide guidance to creative research culture within the group.

2. How were the commercial activities started and currently proceeding?

After about two (2) years of full time research, some commercial projects are undertaken from external customers, in the same research area, since the research group acquired necessary experience, knowledge, skills, and equipment. Thus, only modelling and simulation projects which could complete within less than two (2) years are undertaken from outside organizations. These projects consist of agreed upon requirements, timelines, and payments with formally signed agreements.

Sometimes, research groups undertake totally commercial oriented projects as well. Similar to industry projects, these projects incorporate formal agreements, time bound deliverables, presentations, confidentiality issues, restrictions for publications, and so forth. Customer organization gives requirements, and research group is committed to come up with a solution within a given time period. Research groups commit on such projects only if they have enough resources, infrastructure facilities, experience, knowledge, and skills to build the solution. If the human resources are not enough, they are not reluctant to hire more people. In commercial projects researchers do not try out new things because of the mentioned restrictions and time limitations. Research groups take projects based on their prior research experience in their areas of expertise. As examples, a virtual reality research group takes only simulation projects, a language technology research group takes only language related projects, a telecommunications research group conducts only relevant project in their field. Research groups do not undertake projects out of their areas of expertise. Although they do not experiment on such projects, they use findings of previous research projects to build these commercial solutions. They do not publish papers on these projects because of confidentiality issues and publication restrictions. These projects bring funds for research groups to expand and sustain long term. Therefore, research groups' objective from commercial projects is monetary purposes, so that they can retain people for research projects.

By today (dated June 5, 2015), the ship simulator is deployed at new customer sight and being using for teaching and learning purposes. The university has given the product free of charge, upon necessary equipment and labs are provided by the customer organization. However, the group makes income by providing further modifications and regular maintenance.

3. What were the goals and objectives?

Although, the early goal of the ship simulation project were to develop a commercial product and deliver customer requirements for revenue generation, later it was changed to developing a low cost product by research. Objectives were changed to academic purposes, as examples; undergraduate and postgraduate research projects, academic publications, exhibitions, awards, and so forth. Scope of the project re-defined time to time according to the student projects. Aligned with the research objectives, product development has been happening constantly, along with requirements stated in the initial proposal, while having a long term goal of commercialisation, which has achieved by now.

Contrarily, objectives of the commercial projects are to deliver the customer requirements on time, and to gain revenue in return. The hidden goal of doing these projects is to keep a regular income to retain people and get resources while research funds are lacking. Instinctively, they could make use of knowledge gain through research as well.

3. How did the group handle financing and resources?

At the beginning, the research project was funded by two local grant organizations, whilst basic lab facilities were provided by the university. Some more funds are given by established service providing centres in the university, for examples; multimedia and e-learning service centres, to hire interns for software development activities. In return, the research group helps the centres to provide services and generate income, as they have knowledge and skills in those areas.

As the product evolved, some equipment and shared employment opportunities were provided by the customer who formerly had the need, similar to industry partnerships discussed by many previous researchers (Perkmann et al. 2013). Employees were hired by the customer organization and allowed them to work fulltime in the research and product development activities.

Lately, when the research group became matured in the field of modelling and simulation development, they started undertaking short term (1-2 years) product development projects from outside organizations with revenue in mind. That way the group could pay higher salaries to the researchers, hire more developers, and buy more equipment, hence sustaining long term.

4. Who are the group members and their contribution to the group?

The fellow, who started the research project, still keeps preceding the group, after completing his PhD recently. During interviews, one of the senior academics stated that;

> "*It is his (the PhD student's) effort, that kept the group together this long.*"

There was no formal project manager or any other professional roles and duties. Five (5) senior academics and two (2) junior researchers were the very first research group and it grew over time. Two junior investigators have been doing the research work, one as the PhD work, and other for his MSc research, under senior academics' supervision and advices. The first version of the research product was, a prototype consisted of a moving dot for a given point in a 2D space, researched and developed by those two fellows.

In early 2009, an intern was hired to carry out product development, based on the findings of research that has been doing. The intern was able to finish the task during six (6) months of his training period, while overseen by the two junior researchers. This effort upgraded the product to a computational model with a multiple display. The intern was promoted as a full time developer. Hire interns for software development activities, helps the university to train student interns in an actual working environment.

Undergraduate and postgraduate students from University of Colombo were involved in research and software development activities under senior lecturers' supervision. Most of them involved solely because of their personal interest on modelling and simulation research. At the end there are many researchers who contributed to the project in different ways. Nobody forces or provide requirements for researchers about what to do next. Research students can select a part of the problem as their interest and based on advices of the senior researchers.

5. What is the software development methodology followed?

The research group have not followed any software development methodology for research product development. Module wise development is practiced for the research project, as its people change time to time. Researchers undertake parts of the system to work on as their interest. An iterative development model, similar to rapid prototyping (explained in Chapter 2: Literature review; section 2.2.2) is practiced. They started developing prototypes and experimentation on new techniques after a comprehensive literature review on modelling and simulation development methods. Literature on maritime software systems have been analysed and found new cost effective approaches to develop a selected module of the product by a researcher. Then the knowledge has passed to the developers to code that module as a system component, which can plug in to the main system.

The central architecture is created in a way that modules can be plugged in easily, if developed according to the basic protocols defined by the former developers. Once the product architecture was changed entirely in early 2010, as it was not compatible with the later developed components. One of the senior academics stated that "*Changing the architecture was not a big problem, because of the freedom and flexible software development approach the researchers followed. And we did not restrict their freedom for experiments with strict deadlines and deliverables.*" However, since then the architecture is compatible and all the components are connected and working smoothly. In the commercial systems undertaken time to time, they practiced an Agile method with Waterfall steps, since they came with specific requirements, budget allocations, and time schedules.

6. What is the project lifecycle and phases?

Research and coding are the most basic steps of the research product. They read literature, to come up with new ideas and designs to develop simulations, create a prototype to evaluate the idea, and develop it until the product component is working properly. Later, it is plugged into the main system as a module. The main architecture was developed by the research group at early stages of the project, so that upcoming research could combine without difficulty. Commercial projects are done following an agile approach with Waterfall steps, which consists of planning, requirement analysis, design, development, evaluation, implementation, and maintenance.

7. What are the project milestones?

Schedule and deadlines were mostly depended on university guidelines on students' projects, conferences and commitments to funding organizations. Similar to private software companies, the short term commercial projects got client deadlines and group internal deadlines to meet client requirements. They had deadlines due to exhibitions as well, as they demonstrated their prototype simulations in public gatherings.

8. What kind of relationships did they have with the end users?

At the very beginning, after the tender cancellation, there were no contacts with any of the end users. After sometime, when the research group came up with a demonstrable prototype of a ship simulator with a multiple display, they got the attention of the government organization again, through personal level relationships. Two of the managerial level persons had been given expertise knowledge in ship navigation and control to develop the product. One of them was doing a PhD in Physics, thus he had experience in doing research as well.

9. How was the relationship with community groups?

The research group had personal as well as official level relationships with private and government organizations which are interested in acquiring modelling and simulation based systems. These organizations are considered as their potential customers. Maintaining good relationships with persons from government regulatory, higher academic authorities, and funding agencies are important, since they are helpful in providing facilities and approving funds for projects. Links with other universities and faculties helped to obtain new knowledge and get people from other areas of expertise, for example; physics theories are applied in simulations development, thus linkages with physics departments.

10. What are the design methods and programming languages used?

C++ is the programming language that has been used to develop the entire research based product. The initial developers made it a protocol; otherwise it is difficult to integrate when involving various modules by number of researchers and developers time to time. Free and open source tools and plug-ins are widely used wherever possible; in order to save time, effort and be cost effective. Very few proprietary software tools are used when essential.

11. What are the outcomes of the project?

The outcomes can be mainly divided into two sets, as academic publications and software products. The very first outcome was a manuscript accepted for a conference held in Singapore, based on the tender proposal and low cost simulations development. Although, the research fellow had an idea to register for an MPhil degree, his official registration was speeded up due to conference registration as a student. The first product outcome was the multi display prototype of the system developed by the first intern.

The second manuscript accepted was a poster at a local conference and it won the best poster demonstration certificate at the conference. Product version two was a working product in a good condition, thus grabbed the attention of all the project stakeholders.

MPhil/PhD degrees, undergraduate/postgraduate research projects, conference publications/ posters, awards and exhibitions are considered as the success factors or the measurements of quality of the product.

12. Are there any other important facts?

The industry partnerships did not have any formal agreements regarding product ownership, but informal verbal partnerships based on mutual relationships and trust. They had shared the product development activities, thus the source code synchronized in both places. Therefore, university declared it as a free and open source product, had agreed upon with the government organization. The research group does not try any experiments on commercial projects as they are time restricted. Anyhow they take projects based on their existing knowledge, expertise and resources, thus they deliver them as agreed.

13. What are the possible risks and uncertain factors?

For a simulator there should be a fast frame speed but within low budget computers and equipment, it was a risk. As this was a research project, it was difficult to predict or meet deadlines. There were delays in fund releasing from donor agencies.

5.3. Challenges and best practices

From the beginning the group has functioned properly and continuously, mainly because of PhD student's enormous interest and effort on keeping the group together through effective communication. Once the research group had a demonstrable working product only they could get the attention of the interested customer. Different funds collecting strategies, particularly; government funding for research, income by providing expertise services via university centres, industry partnerships for people and resource sharing, and income generation from small scale commercial product development are used to finance the research project until it is commercialised. At present it has its own revenue generation model through regular upgrades and maintenance, as it is implemented and running at a customer sight.

Every outcome of the research project has led to significant activities, which ultimately contributed in project continuation and successful implementation. When some contribution is made and it is appreciated by relevant communities, it does not only motivate the research group, but also directs to get external investments and attention of the end users. They have developed small scale commercial products, and provided services in their areas of expertise, as a strategy to overcome funding issues and to maintain customer relationships,. The research project is separated from commercial activities, until it is ready for commercialisation. The management becomes less complicated, when communicating one project at a time with a selected set of people. The knowledge, skills and new techniques found through research are used practically in commercial projects, since they were extensively evaluated and proved. By the time the research product is ready for commercialisation, the research group has expanded and stabilised as an individual entity within university, which could balance both research and product development equally.

Chapter 6. Conclusion

To conclude, the overall contributions of this study are summarised. Challenges encountered during the study, limitations and future work are also discussed afterwards.

6.1. Contributions

Contributions are discussed by answering the three main research questions.

1. Why do software development research projects in universities need different management practices from existing software development methodologies?

Industry projects has a certain promise on their investments than university projects because of the formal agreements. Software companies are focused on revenue generation whereas universities are research oriented. Industry projects have investments at the beginning and predictable income after product delivery. However, it is witnessed that most of the university projects has only a few short term funding sources. Software development teams in companies mostly consisted of a project manager, architects, software developers, business analysts, and QA engineers. University teams consisted of Phil/PhD students who do research and software development, and supervisors who work as senior lecturers at universities. Some teams had research assistants and interns as well. Main difference between university and industry teams is there were no dedicated people for research in the industry. The time they spent on research is very low compared to university teams.

Most of the software companies practiced Agile methods to manage software development processes and teams. It was noted that existing software development methods are rarely used in university projects, because of their unpredictable nature. According to the university researchers Agile is suitable for face paced development projects with fairly defined requirements and predictable outcomes. All the companies followed a precise life cycle for every project they perform, whereas universities adjusted their practices and came up with new methods time to time, depending on their research possibilities. It was evident that industry teams were fast paced and milestone driven in order to meet customer requirements. University teams had many different predictable and non-predictable milestones set by conferences, exhibitions, funding agencies, university regulations, and sometimes customer given deadlines.

Universities had less interactions with the end users compared to the industry. Industry companies had formal agreements with their customers. Universities had mostly informal partnerships with volunteers, thus there were a lot of uncertainty. This might have been a problem for university researchers when transferring their research findings to the society. It was observed that universities had relationships with a larger group of communities to get support in different activities. Industry companies did not have relationships with external parties except from the state government and their potential customers.

Design methods were almost similar in the industry as well as the universities. It is understood that the industry developers chose from a wide variety of programming languages depending on their skills and nature of given projects. University researchers had a narrow selection of programming languages compared to the industry. The main difference in project outputs between universities and industry was publications. Some university projects had system components, whereas companies always developed full solutions. These components had to merge with another system to be used as a full product, therefore problems were arose when introducing to end users.

All the respondents from companies and many from universities stated end user interactions, continuous feedback and customer satisfaction were important. Specifically in universities, planning, distribution of tasks, and team management is identified as an important task by all the teams. It was essential to keep team members motivated and interested towards the research, if not there was a risk of them leaving the university projects because of high uncertainty. Publications are considered mandatory in research projects. Data gathering was also played a vital part in some research projects which had data driven processes.

From above findings it is understood; compared to software industry projects, university projects are consisted of a lot of uncertainty. People on those projects needed to make a considerable effort to be motivated and keep interest towards research and innovation. In the industry, project life cycle stages and milestones are aligned with the software development methodology they practice. Most of the management aspects of a commercial software development project is covered by the software development methodology they use. Management of end user interactions and outcomes are also included in Agile methods.

University projects had various milestones. They developed a range of outputs, for examples; research publications, dissertations, posters, exhibits, prototypes and software products. End user interactions are limited as they are not bounded by any agreements, but voluntary. As the authors figured, university researchers were not capable of using well defined software development methodologies, because of the uncertainty nature of research goals, funding, project teams, project lifecycle, milestones, end user interactions, and research outcomes. It was not possible to manage research based software development projects in universities, by using the existing software development methodologies. Therefore, the authors investigated on unique challenges faced by university researchers in order to find suitable strategies and best management practices.

2. What are the challenges faced by researchers when managing research and software product development within Sri Lankan universities?

People retention is identified as the most challenging activity when managing Sri Lankan research groups. Sometimes, experienced people left projects by the time funds were released. Lack of knowledge in high tech areas was an issue with novice researchers. Lack of funds, resources and access to publication libraries are considered as critical issues faced by researchers. Late plans on product development had led to lack of funds and not getting user interactions from the beginning of a project. Keeping end user interest, interactions and acceptance were always at a risk, because their participation was voluntary. Some teams had restrictions on publishing their work because of confidentiality problems raised by industry partnerships or commercial projects. Scarcity of test data was a critical issue in some research projects. Time was vital for research, because findings could be outdated very fast and had to meet deadlines from universities and funding agencies. Delays on releasing funds by donor agencies, made it hard for researchers to carry out research work as they planned. Most of the products developed by researchers are not maintained properly even if they are deployed satisfactorily.

There were considerable amount of research initiations that were struggling to continue as stable R&D groups within Sri Lankan universities because of these challenges. Therefore, the authors explored through established research groups to find best practices they have been used to overcome the challenges.

3. How do established research groups manage research based software product development within Sri Lankan universities?

Awareness of attractive research happening in their field is considered as a fine quality of supervisors in established research groups. Building a community around an identified research problem is also very much needed to generate ideas, find people and also funds. It is recommended for a new research group to start working together with established R&D centres in the same university. That way, novice researchers could get required exposure, equipment and may be some funds. Research supervisors should be able to gather a set of passionate and motivated individuals to carryout research work. The communities could help in finding passionate people. A strong and continuous leadership throughout a research project, is identified as an essential fact for its successful management. Prompt an inspirational leader among the group is another responsibility of a research supervisor. By encouraging publications, organising workshops and collaborative events to meet relevant communities as well as industry partners, research supervisors could stimulate an overall R&D culture within a research group. It would be great if Sri Lankan universities can get institution wide access to scholarly research publication libraries and organise more events with the industry in order to encourage research throughout the country.

Having a long term vision, goals and objectives is considered as a success factor, because then the research group is getting themselves ready for the future. They are encouraged to acquire several funding sources to execute plans. Researchers are more likely to embrace the uncertainty. Changes are always welcome to developed products or designs in order to have better outcomes. Large research problems are divided into meaningful and manageable components, and have given to individual research students. Established research groups have undertaken short term commercial software development projects in their research area, to have a better income for research projects. It is important to keep up community relationships and an attracted user base continuously. Support new research initiatives in the university is recommended to improve overall research happen within the university.

6.1.1. Analysis between industry practices and findings of this study

A brief analysis of findings of this study compared to industry practices are given in Table 6-1. Agile practices and open sources principals are compared with the university framework.

Table 6-1: Analysis between industry practices and university practices

CONCEPTS	FRAMEWORKS		
	University framework	Agile framework (Beck et al. 2001; Conboy & Morgan 2011; Schwaber 2004; Parsons & Lange 2007; Wilson & Doz 2011)	Open source model framework (Raymond 2001; Hippel 2002; Mockus et al. 2000; Hertel et al. 2003; Mockus et al. 2002; Lakhani & Von Hippel 2003; Godfrey & Tu 2000)
	Research awareness	Continuous attention to technical excellence and good design.	Technology awareness.

or	Community building	-	Voluntary developers from around the world.
P R I N C I P L E S	Work collaboratively	Self-organizing teams - business people and developers must work together daily throughout the project. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	Users are treated as co-developers.
	Recruiting passionate people	Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.	Voluntary developers join as their interest on the project.
	Strong leadership	Team leader – e.g., Scrum master.	Leadership is very important to gather a group of voluntary developers from around the world.
	R&D culture	Working software over comprehensive documentation. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	Dynamic decision making structure.
	Clear objectives	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Early releases. Software testing done by users during beta-releases.
	Embracing uncertainty	Responding to change over following a plan.	Frequent updates and intigration.
	Division of research ideas	Product backlog.	High modularization.

	Several funding sources	Customer collaboration over contract negotiation.	Resources are shared by the developers who themselves are the users.
	Commercial software projects	Working components of the software solution is the primary measure of progress. Revenue generation is the main focus.	Several versions. Free distribution of the software.
	Support new research initiatives	-	-
	Outsource maintenance	Continuous maintenance.	Continuous bug fixing by developers who use the software.
	Publication access	-	Mailing lists and blogs.
	Inspire industry partnerships	-	-

As summarized in Table 6-1, technical excellence and design quality is overlooked constantly by Agile teams whereas technology awareness is important in FOSS environment compared to university's research awareness creation. Although university and FOSS models depend mostly on community building, it is not much important in Agile practices. However, all types of software development groups work collaboratively. Agile methods encourage self-organizing teams, whereas, FOSS developers contribute to develop software systems that they have a requirement to use. Leadership roles are different in the three contexts: Agile teams have defined leadership roles such as Scrum master, FOSS teams has a committed and an inspirational leader who maintains a large user base till the product becomes fully functional, and university leadership role is almost similar to FOSS team lead.

Universities has a unique R&D culture which is led by the professors. Agile culture is based on frequent releases and user-developer collaborative environment. FOSS evolve with early releases which test by users during beta-releases. Embracing uncertainty is very much needed in university projects in order to come up with innovative designs. Agile teams respond to changes even later times of product development. FOSS develop with frequent updates and integrations. Research ideas are divide among several students and integrate all of them to make a complete solution in university environment. Agile groups maintain a product backlog from the beginning of a project. They take few prioritized features to develop in each iteration after discussing with the client. FOSS developers use high modularization to separate a system into smaller components.

University projects get few funding sources after submitting several research proposals. Agile framework motivate customer collaborations and making value out of working software components rather than contract negotiation for the entire project at the beginning. FOSS framework is used a resources sharing model by all the co-developers around the world. University teams undertake small scale commercial systems development in order to keep a smooth case flow to the research entity. Agile teams deliver working components of the software to generate continuous income. FOSS release several versions of the software and distribute free of charge over the internet. Support new initiatives, outsource maintenance, publications and industry partnerships are unique practices to universities.

6.2. Challenges encountered during this study

Firstly, the authors found difficulties in defining the research problem, as it was vague at the beginning. Secondly, problems are faced due to lack of knowledge in qualitative research methods. Then confidentiality and privacy issues were there when collecting and publishing data. Those issues had to be strictly considered when reporting findings as well.

6.3. Limitations

The study entertained only the software development associated research in Sri Lankan state universities. The authors realized that the opinions derived from these findings are subjective and context specific. This study did not address projects in research units solely owned by private companies or other government institutions. Only one case study is discussed because of accessibility issues, lack of time to the researchers to participate in this study, as well as lack of established research groups in Sri Lankan universities.

6.4. Future work

There are many future research paths arise from this research. To improve the practical relevance of this study’s findings, an AR approach can be done with a newly formed or to be formed research group in a Sri Lankan university. The framework can be further evaluated in international contexts and in the software industry where research and innovation is a major ingredient. These findings can be used to propose a suitable software development methodology for entrepreneurial universities. They can be applied to find reasons for attempted projects that were unable to implement. This framework can also be used to design university curriculum in a way that motivate students to develop and implement their conceptual systems. It will guide them to choose research projects as their interests, knowledge and skills.

References

Abrahamsson, P. et al., 2002. Agile software development methods: Review and analysis. *Vtt Publications*, 478.

Alavi, M., Kayworth, T.R. & Leidner, D.E., 2006. An Empirical Examination of the Influence of Organizational Culture on Knowledge Management Practices. *Journal of Management Information Systems*, 22(3), pp.191–224.

Altmann, J., 1974. Observational study of behavior: sampling methods. *Behaviour*, 49(3), pp.227–67. Available at: http://www.ncbi.nlm.nih.gov/pubmed/4597405.

Anon, 2015. UoM Research Highlights. Available at: https://www.mrt.ac.lk/web/research/uom-research-highlights.

Aspray, W., Keil-Slawik, R. & Parnas, D.L., 1996. *History of Software Engineering*, Citeseer.

Atkinson, R.C. & Pelfrey, P.A., 2010. Science and the Entrepreneurial University. *Issues in Science & Technology*, 26, pp.39–48.

Bakker, G., 2013. Money for nothing: How firms have financed R&D-projects since the Industrial Revolution. *Research policy*, 42(10), pp.1793–1814.

Basit, T.N., 2003. Manual or Electronic: the Role of Coding in Qualitative Data Analysis. *Educational Research*, 45(2), pp.143–154.

Baskerville, R. & Myers, M., 2004. Special Issue on Action Research In Informaton Systems: Making IS Research relevant to practice foreword. *Management Information Systems*, 28(3), pp.329–335.

Batra, D. et al., 2010. Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry. *Communications of the Association for Information Systems*, 27(1), pp.378–395.

Beck, K., 1999. Embracing Change with Extreme Programming. *IEEE Computer*, 32(10), pp.70–77.

Beck, K. et al., 2001. Manifesto for agile software development. *http://agilemanifesto.org/*. Available at: http://agilemanifesto.org/principles.html.

Beck, K. & Fowler, M., 2000. *Planning Extreme Programming*, Addison Wesley.

Berg, B.L., 2001. *Qualitative Research Methods for the Social Sciences* 4th ed. S. L. Kelbaugh, ed., Pearson.

Von Bertalanffy, L., 1968. *General System Theory*,

Bhattacherjee, A., 2012. *Social Science Research: Principles, Methods, and Practices*, USF Tampa Bay Open Access Textbooks Collection.

Boehm, B., 2006. A view of 20th and 21st century software engineering. *Proceeding of the 28th international conference on Software engineering - ICSE '06*, p.12.

Brereton, M. et al., 2014. Beyond ethnography: engagement and reciprocity as foundations for design research out here. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, pp. 1183–1186.

Brooks, F.P., 1975. *The Mythical Man-Month* 2008th ed., Addison-Wesley Reading, MA.

Brooks, F.P.J., 1986. No silver bullet-essence and accidents of software engineering. *Proceedings of the IFIP Tenth World Computing Conference*, pp.1069–1076.

Bryman, A., 2011. Triangulation. *Encyclopedia of Social Science Research Methods*, pp.1142–1143.

Carayannis, E.G. et al., 1998. High-technology spin-offs from government R&D laboratories and research universities. *Technovation*, 18(1), pp.1–11.

Carr, M., Prototyping and Software Development Approaches.

Charmaz, K., 2008. Grounded Theory as an Emergent Method. *Handbook of Emergent Methods*, (3), pp.155–170.

Checkland, Peter and Holwell, S., 1997. *Information, systems and information systems: making sense of the field*, Wiley.

Cheng, H.H. & Yang, H.L., 2011. Student team projects in information systems development: Measuring collective creative efficacy. *Australasian Journal of Educational Technology*, 27(6), pp.881–895.

Clark, B., 2001. The entrepreneurial university: new foundations for collegiality, autonomy, and achievement. *Higher Education Management*, 13(2), pp.9–24.

Claybaugh, C.C. & Srite, M., 2009. Factors Contributing to the Information Technology Vendor – Client Relationship. *Journal of Information Technology Theory and Application*, 10(2), pp.19–38.

Cockburn, A., 2004. *Crystal clear: a human-powered methodology for small teams.*, Pearson Education.

Cockburn, A., 2000. Selecting a Project's Methodology. *IEEE Software*, 17(4), pp.64–71.

Cockburn, A. & Highsmith, J., 2001. Agile Software Development: The People Factor. *Computer*, 34(11), pp.131–133.

Cohen, S.G. & Bailey, D.E., 1997. What Makes Teams Work: Group Effectiveness Research from the Shop Floor to the Executive Suite. *Journal of Management*, 23(3), pp.239–290.

Colyvas, J., Crow, M., Gelijns, A., Mazzoleni, R., Nelson, R.R., et al., 2002. How do university inventions get into practice? *Management Science*, 48(1), pp.61–72.

Colyvas, J., Crow, M., Gelijns, A., Mazzoleni, R., Nelson, R.R., et al., 2002. How do university inventions get into practice? *Management Science*, 48(1), pp.61–72.

Commonwealth, 2007. Commercialisation Case Studies From the: National Survey of Research Commercialisation Years 2003 and 2004. , (August). Available at: http://www.innovation.gov.au/INNOVATION/REPORTSANDSTUDIES/Pages/NationalSurveyofResearchCommercialisation.aspx.

Commonwealth, 2011. *The national survey of research commercialisation 2010-2011*, Commonwealth of Australia 2012. Available at: http://www.innovation.gov.au/INNOVATION/REPORTSANDSTUDIES/Pages/NationalSurveyofResearchCommercialisation.aspx.

Conboy, K. & Morgan, L., 2011. Beyond the customer: Opening the agile systems development process. *Information and Software Technology*, 53(5), pp.535–542.

Corbin, J.M. & Strauss, A., 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13(1), pp.3–21.

Coyne, I.T., 1997. Sampling in qualitative research. Purposeful and theoretical sampling; merging or clear boundaries? *Journal of advanced nursing*, 26(3), pp.623–630.

Creswell, J.W., 2007. Designing a Qualitative Study. In *Qualitative inquiry and research design: Choosing among five approaches*. Sage, pp. 35–41.

D'Este, P. & Patel, P., 2005. University-Industry Linkages in the UK : What are the Factors Determining the Variety of University Researches' Interactions with Industry? In

DRUID Tenth Anniversary Summer Conference.

Day, J., Junglas, I. & Silva, L., 2009. Information Flow Impediments in Disaster Relief Supply Chains. *Journal of the Association for Information Systems*, 10(8), pp.637–660.

Dedrick, Jason and West, J., 2003. Why firms adopt open source platforms: a grounded theory of innovation and standards adoption. In *Proceedings of the workshop on standard making: A critical research frontier for information systems*. Seattle, WA, pp. 236–257.

Dunne, C., 2011. The place of the literature review in grounded theory research. *International Journal of Social Research Methodology*, 14(2), pp.111–124.

Dybå, T. & Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), pp.833–859.

El-haddadeh, R. & Ali, M., 2010. E-Government implementation Challenges : A Case study E-Government implementation Challenges : A Case study.

Ernø-kjølhede, E., 2000. *Project Management: Theory and the Management of Research Projects*,

Etzkowitz, H., 2003. Research groups as "quasi-firms": the invention of the entrepreneurial university. *Research Policy*, 32(1), pp.109–121.

Etzkowitz, H., 2001. The second academic revolution and the rise of entrepreneurial science. *IEEE Technology and Society Magazine*, 20(2), pp.18–29.

Farsi, J.Y., Modarresi, M. & Zarea, H., 2011. Obstacles and Solutions of Commercialization of University Research : Case Study of Small Businesses Development Center of University of Tehran. , (7), pp.1–23.

Fontana, R., Geuna, A. & Matt, M., 2006. Factors affecting university–industry R&D projects: The importance of searching, screening and signalling. *Research Policy*, 35(2), pp.309–323. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0048733305002210 [Accessed March 18, 2013].

Gasson, S., 2004. Rigor in grounded theory research: An interpretive perspective on generating theory from qualitative field studies. In *The handbook of information systems research*. pp. 79–102.

Geenhuizen, M. Van, 2011. Entrepreneurial University Activity : Can Field or Living Labs be Supportive ? In *9th Triple Helix Conference*. Palo Alto, California: Stanford University, pp. 11–14.

Gibb, A., 2001. Towards the Entrepreneurial University? *Higher Education in Europe*, 26, pp.427–436.

Gjerding, A.N. et al., 2006. Twenty Practices of an Entrepreneurial University. *Higher Education Management and Policy*, 18, pp.1–28.

Glaser, B. & Strauss, A., 1967. *The discovery grounded theory: strategies for qualitative inquiry*,

Glaser, B.G., 2012. Stop.Write! Writing Grounded Theory! *Grounded Theory Review*, 11(1).

Glaser, B.G. & Strauss, A.L., 2006. *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Transaction Publishers.

Godfrey, M.W. & Tu, Q., 2000. Evolution in open source software: a case study. *Proceedings International Conference on Software Maintenance ICSM-94*, pp.131–142. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=883030.

Goldkuhl, Göran and Cronholm, S., 2003. Multi-grounded theory--Adding theoretical grounding to grounded theory. In *2nd European Conference on Research Methodology for Business and Management Studies, Reading University, Reading, UK*.

Google, 2013. Google AdWords. Available at: https://adwords.google.com.

Google, 2014. Our history in depth. *Google Company*. Available at: http://www.google.lk/about/company/history/.

Gordon, V.S. & Bieman, J.M., 1995. Rapid prototyping: Lessons Learned. *IEEE Software*, 12(1), pp.85–95.

Gregor, S., 2006. The nature of theory in information systems. *MIS Quarterly*, 30(3), pp.611–642.

Groenewald, T., 2004. A Phenomenological Research Design Illustrated. *International Journal of Qualitative Methods*, 3(1), pp.1–26.

Guerrero, M. & Urbano, D., 2012. The development of an entrepreneurial university. *Journal of Technology Transfer*, 37, pp.43–74.

Gulbrandsen, Magnus and Smeby, J.-C., 2005. Industry funding and university professors' research performance. *Research policy*, 34(6), pp.932–950.

Guyette, S., 1983. *Community-based research: A handbook for Native Americans*, American Indian Studies Center, University of California.

Hammersley, M. & Atkinson, P., 2007. What is ethnography. In *Ethnography*. Taylor & Francis e-Library, pp. 1–19.

Hargadon, A., 2003. *How breakthroughs happen: The surprising truth about how companies innovate*, Harvard Business Press.

Hashimoto, M. et al., 1997. Technology transfer in university-based research centers at the University of New Mexico. *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, 3, pp.116–123.

Hertel, G., Niedner, S. & Herrmann, S., 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy*, 32(7), pp.1159–1177. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0048733303000477 [Accessed November 7, 2013].

Hevner, a. R., March, S.T. & Park, J., 2004. Design Science in Information Systems Research. *MIS Quarterly*, 28(1), pp.75–105.

Highsmith, J. et al., 2002. Agile Software Development. *CROSSTALK The Journal of Defense Software Engineering*, 15(10).

Highsmith, J. & Cockburn, A., 2001. Agile Software Development: The Business of Innovation. *Computer*, 34(9), pp.120–127.

Hippel, E. Von, 2002. Open Source Software and the " Private-Collective " Innovation Model : Issues for Organization Science i Open Source Software and the " Private-Collective " Innovation Model : Issues for Organization Science. , 14(2003), pp.208–223.

Houde, S. & Hill, C., 1997. What do Prototypes Prototype? *Handbook of Human Computer Interaction*, 2, pp.367–381.

Hu, T., Dai, H. & Li, X., 2010. Software Development Methodologies, Trends, and Implications. *SAIS 2010 Proceedings*, p.31.

Humphrey, W.S., 2005. Why Big Software Projects Fail : The 12 Key Questions. *CrossTalk The Journal of*, pp.25–29.

Ilgen, D.R. et al., 2005. Teams in organizations: from input-process-output models to IMOI models. *Annual review of psychology*, 56, pp.517–543.

Isabella, L.A., 1990. Evolving interpretations as a change unfolds: How managers construe key organizational events. *Academy of Management journal*, 33(1), pp.7–41.

Jayawardena, D.S. & Ekanayake, L.L., 2010. Adaptation analysis of Agile Project Management for managing IT projects in Sri Lanka. In *Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on*. pp. 1–4.

Jessuru, B., Rukanova, B. & Tan, Y., 2008. Analyzing Living Labs as part of the complete innovation development process. In *EIS 2008 Proceedings*. Association for Information Systems.

Johnson, Richard A and Kast, Fremont E and Rosenzweig, J.E., 1964. Systems Theory and Management. *Management Science*, 10(2), pp.367–384.

Kast, F.E. & Rosenzweig, J.E., 1972. General systems theory: applications for organization and management. *Academy of Management Journal*, 15(4), pp.447–465.

Kothari, C.R., 2004. *Research Methodology: Methods and Techniques*,

Kroeze, J., Pretorius, P. & Roode, J., 2010. The role of an ICT research professor in South Africa. *Sprouts: Working Papers on Information Systems,*. Available at: http://sprouts.aisnet.org/10-32 [Accessed August 4, 2014].

Lakhani, K.R. & Hippel, E. Von, 2003. How open source software works : " free " user-to-user assistance. , 32(July 2002), pp.923–943.

Lakhani, K.R. & Von Hippel, E., 2003. How open source software works:"free" user-to-user assistance. *Research policy*, 32(6), pp.923–943.

Laplante, P.A. & Neill, C.J., 2004. The demise of the Waterfall model is imminent. *Queue*, 1(10), p.10.

Larman, C. & Basili, V.R., 2003. Iterative and incremental development: A brief history. *Computer*, 36(6), pp.47–56.

Laws, K. & Mcleod, R., 2004. Case study and grounded theory : Sharing some alternative qualitative research methodologies with systems professionals. *Proceedings of the 22nd international conference of the systems dynamics society*, 78.

Lee, Allen S and Baskerville, R.L., 2003. Generalizing Generalizability in Information Systems Research. *Information systems research*, 14(3), pp.221–243.

Levén, P., Holmström, J. & Mathiassen, L., 2014. Managing research and innovation networks: Evidence from a government sponsored cross-industry program. *Research Policy*, 43(1), pp.156–168.

Meyer-Krahmer, F. & Schmoch, U., 1998. Science-based technologies: university–industry interactions in four fields. *Research Policy*, 27(8), pp.835–851.

Miles, M.B., Huberman, A.M. & Saldana, J., 2014. *Qualitative Data Analysis : A Methods Sourcebook*, Sage.

Mockus, A., Fielding, R.T. & Herbsleb, J., 2000. A Case Study of Open Source Software Development : The Apache Server. In *Proceedings of the 22nd international conference on Software engineering*. ACM, pp. 263–272.

Mockus, A., Fielding, R.T. & Herbsleb, J.D., 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(3), pp.309–346.

Munasinghe, L., Peter, P.L.S. & Perera, T.D.S., 2002. Growth Prospects for the Software

Industry in Sri Lanka and an Appropriate Policy Framework. *Journal of Science of the University of Kelaniya Sri Lanka*.

Numprasertchai, S. & Igel, B., 2003. Managing knowledge through collaboration: two cases of managing research in university laboratories in Thailand. *Engineering Management Conference, 2003. IEMC'03. Managing Technologically Driven Organizations: The Human Side of Innovation and Change*, pp.575–579.

Ogheneovo, E.E., 2014. Software Dysfunction : Why Do Software Fail ? *Journal of Computer and Communications*, 2(April), pp.25–35.

Parsons, D. & Lange, M., 2007. Software Development Methodologies , Agile Development and Usability Engineering.

Perkin, H., 2006. History of universities. In *International handbook of higher education*. Springer, pp. 159–205.

Perkmann, M. et al., 2013. Academic engagement and commercialisation: A review of the literature on university–industry relations. *Research Policy*, 42(2), pp.423–442.

Perkmann, M., 2007. University-industry relationships and open innovation : towards a research agenda. *International Journal of Management Reviews*, 9(4), pp.259–280.

Petersen, K., Wohlin, C. & Baca, D., 2009. The waterfall model in large-scale development. In *Product-Focused Software Process Improvement*. Springer, pp. 386–400.

Pressman, R.S., 2010. *Software Engineering: A Practitioner's Approach*, New York : McGraw-Hill.

Rajapaksa, G., 2013. Sri Lanka ready to become Asia's next knowledge hub. *Sunday Observer*. Available at: http://www.sundayobserver.lk/2013/12/29/fea02.asp.

Raymond, E.S., 2001. *The Cathedral and the Bazaar: Musings on linux and open source by an accidental revolutionary* Revised Ed. T. O'Reilly, ed., O'Reilly & Associates.

Reel, J., 1999. Critical success factors in software projects. *Software, IEEE*, (May/June 1999), pp.18–23.

Rising, L., Janoff, N.S. & Systems, A.G.C., 2000. The Scrum Software Development Process for Small Teams. , (August).

Royce, W.W., 1970. Managing the development of large software systems. *Proceedings of IEEE WESCON*, 26(8), pp.1–9.

Ruparelia, N.B., 2010. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3), pp.8–13.

Sarker, S., Xiao, X. & Tanya, B., 2012. Towards an Anatomy of "Successful" Qualitiative Research Manuscripts in IS: A Critical Review and Some Recommendations. In *Thirty Third International Conference on Information Systems*. pp. 1–21.

Schwaber, K., 2004. *Agile Project Management with Scrum*, Microsoft Press.

Scott, K. & Howell, D., 2008. Clarifying analysis and interpretation in grounded theory: Using a conditional relationship guide and reflective coding matrix. *International Journal of Qualitative Methods*, 7(2), pp.1–15.

Sen, R., 2006. A Qualitative Analysis Of The Role of Users, Vendors, and Governments in the Standards Development Process. *Communications of the Association for Information Systems*, 17(1), pp.57–70.

Siegel, D.S. et al., 2003. Commercial knowledge transfers from universities to firms: improving the effectiveness of university–industry collaboration. *The Journal of High Technology Management Research*, 14(1), pp.111–133.

Sirimanna, B., 2011. Aligning Sri Lanka's scientific research to achieve national objectives. *Business Times*. Available at: http://www.sundaytimes.lk/110213/BusinessTimes/bt24.html.

Skinner, C.E., 1917. Industrial Research and Its Relation to University and Governmental Research. *Transactions of the American Institute of Electrical Engineers*, 36, pp.871–903.

SLASSCOM, 2014. *Sri Lankan IT/BPM Industry 2014 Review*,

Slater, S.F. & Mohr, J.J., 2006. Successful development and commercialization of technological innovation: insights based on strategy type. *Journal of Product Innovation Management*, 23(1), pp.26–33.

Smith, J.M., 1987. Optimization theory in evolution. *Annual Review of Ecology and Systematics*, 9, pp.31–56.

Stilgoe, J., Owen, R. & Macnaghten, P., 2013. Developing a framework for responsible innovation. *Research Policy*, 42(9), pp.1568–1580.

Strauss, A. & Corbin, J., 1994. Grounded theory methodology. In *Handbook of qualitative research*. pp. 273–285. Available at: http://cms.educ.ttu.edu/uploadedFiles/personnel-folder/lee-duemer/epsy-5382/documents/Grounded theory methodology.pdf“ onmousedown=”return scife_clk(this.href,’gga','gga','2').

Tanha, D. et al., 2011. Commercialization of University Research and Innovations in Iran: Obstacles and Solutions. *Journal of Knowledge Management, Economics and Information Technology*, 1(7), pp.126–146.

Tartari, V., Perkmann, M. & Salter, A., 2014. In good company: The influence of peers on industry engagement by academic scientists. *Research Policy*, 43(7), pp.1189–1203.

Torvalds, L. & Diamond, D., 2001. *Just for fun: The story of an accidental revolutionary*, Harper Audio.

Urquhart, Cathy and Lehmann, Hans and Myers, M.D., 2010. Putting the “theory”back into grounded theory: guidelines for grounded theory studies in information systems. *Information systems journal*, 20(4), pp.357–381.

Vise, D.A. & Malseed, M., 2008. *The Google Story: For Google's 10th Birthday*, Random House Digital, Inc.

Walsham, G., 1995. Interpretive case studies in IS research: nature and method. *European Journal of information systems*, 4(2), pp.74–81.

Weinberg, G.M., 2003. Destroying Communication and Control in Software Development. *CROSSTALK The Journal of Defense Software Engineering*, 16(4), pp.4–8.

Willis, P. & Trondman, M., 2002. Manifesto for Ethnography. *Cultural Studies ↔ Critical Methodologies*, 2(3), pp.394–402.

Wilson, K. & Doz, Y.L., 2011. Agile Innovation: A Footprint Balancing Distance and Immersion. *California Management Review*, 53(2), pp.6–26.

Winter, R., 2008. Design science research in Europe. *European Journal of Information Systems*, 17, pp.470–475.

Wolfswinkel, Joost F and Furtmueller, Elfi and Wilderom, C.P., 2013. Using grounded theory as a method for rigorously reviewing literature. *European Journal of Information Systems*, 22(1), pp.45–55.

Yin, R.K., 2014. *Case study research: Design and methods*, Sage publications.

Yin, R.K., 2009. *Case study research: Design and methods*, Sage.

Appendices

Appendix I - Questionnaire to find differences between industry and universities

	Questions	Related categories
1.	What is your designation?	General information, research group, project team
2.	How many years of experience do you have in the software industry / academia?	General information, research group, project team
3.	What are the usual ways of initiating a project in your company / in university?	Requirement for research and product development, Software product
4.	How do they finance it?	Resources
5.	How do you get required resources?	Resources
6.	What are the different roles of the team and the structure? (Including you)	Research group, project team
7.	What is the software development methodology do you use?	Software development methodology, life cycle
8.	What are the phases of the software development life cycle followed?	Software development methodology, life cycle
9.	What are the usual milestones?	Life cycle
10.	What are the software designing methods?	Software development methodology
11.	What are the programming language(s)?	Software development methodology
12.	What kind of a relationship do you have with the end users?	Research group, project team, external partnerships
13.	What kind of community groups do you engage with when executing the project?	External partnerships
14.	What are the final outputs?	Software product
15.	What are the other important facts as you think in your work?	-
16.	What are the possible risks and uncertain factors?	-

Appendix II - Questionnaire to identify challenges and best practices in university projects

Requirement for research and product development;

	Questions	Related categories
1.	Can you give a short description about the project?	Requirement for research and product development, community problem
2.	What is the requirement for research and product development?	Requirement for research and product development
3.	Who are the end users?	End users, community problem, external partnerships
4.	Is it a problem of a larger community or a problem raised by a customer?	End users, community problem, customers
5.	What is the solution?	Ideas, solution, software product, innovation
6.	What are the outcomes?	Solution, innovation, software product

Resources;

7.	What are the required resources?	Resources, time, funds, equipment
8.	Did you get them before project starts?	Resources, time
9.	What is the project duration? Timeline?	Time
10.	What are the funding sources?	Resources, funds, external partnerships, industry, government, networks, alumni
11.	How do you handle resource management activities?	TTO, resources

Research group;

12.	Can you explain/draw me the structure of the team, specific roles, qualifications and the type of contribution?	Research group, organization structure, personality traits
13.	Do each of them work fulltime or part-time?	Research group, time
14.	How and why did they join?	Personality traits
15.	How often do you have group discussions?	Research group, discussions, ideas
16.	Do you have any external domain experts?	External partnerships
17.	How often do you meet those experts?	External partnerships, discussions, ideas
18.	How do you get ideas to solve the research problem?	Research group, discussions, ideas, external partnerships
19.	Do you have end users as a part of the group?	Research group, end users
20.	How do you handle administration activities?	TTO, research group

External partnerships;

21.	Who are the external parties the research group has collaborated with?	External partnerships, industry, government, networks, alumni, ideas, discussions
22.	What kind of partnerships do you have with external parties?	External partnerships, industry, government, networks, alumni, resources, ideas, discussions
23.	How do you handle formal partnerships?	TTO, external partnerships

Other;

24.	What are the issues you faced so far?	Problems, challenges
25.	How did you solve them and what are the lessons learnt?	Ideas

Appendix III - Scorecard for validating challenges

SCORECARD FOR VALIDATING CHALLENGES

Challenges faced	Yes	Priority level (#)
1. People retention		
2. Lack of funds and economic uncertainty		
3. Access restrictions to publication libraries		
4. Late plans on product development		
5. Getting end user interactions		
6. Problems due to industry partnerships		
7. Problems with funding agencies		
8. Products maintenance issues raised by users		

Instructions to fill the scorecard

1. Put a tick (√) in the cages in "Yes" column, if it was a challenge faced by your research group

2. After identifying all the challenges, prioritize them in the "Priority level" column by putting numbers. Please use Number 1 to indicate the most challenging activity and proceed in sequential order as you think.

Thank you very much for your support in this study.

Appendix IV - Scorecards for validating best practices

SCORECARD FOR VALIDATING BEST PRACTICES

Best practices to research supervisors	Yes	Priority level (#)
1. Awareness of attractive research happening in their field		
2. Community building		
3. Work together with established R&D centres in the same university		
4. Gather a group of passionate people		
5. Prompt an inspirational leader		
6. Stimulate an overall R&D culture		

Best practices to universities and regulatory	Yes	Priority level (#)
7. Getting institution wide access to scholarly research publication libraries		
8. Organize events with the industry		

Instructions to fill the scorecard

1. Put a tick (√) in the cages in "Yes" column, if these practices are used to management your research group by research supervisors and if you think the 7th and 8th practices are useful for Sri Lankan universities

2. After identifying all the best practices, prioritize them in the "Priority level" column by putting numbers. Please use Number 1 to indicate the most important practice and proceed in sequential order as you recommend.

Thank you very much for your support in this study.

SCORECARD FOR VALIDATING BEST PRACTICES

Best practices to research groups	Yes	Priority level (#)
1. Having a long term vision, goals and objectives		
2. Embracing the uncertainty and changes		
3. Dividing large research problems into meaningful and manageable components		
4. Acquiring several funding sources		
5. Undertaking short term commercial software development projects in their research area		
6. Keeping up community relationships		
7. Support new research initiatives in the university		
8. Outsource maintenance to commercial software development entities attached to the university		

Instructions to fill the scorecard

1. Put a tick (√) in the cages in "Yes" column, you think above practices are useful for managing research groups in Sri Lankan universities

2. After identifying all the best practices, prioritize them in the "Priority level" column by putting numbers. Please use Number 1 to indicate the most important practice and proceed in sequential order as you recommend.

Thank you very much for your support in this study.

Appendix V - Example data collection and analysis sheets

Timestamp	1. What is the usual way of initiating a project in your company?	2. How do they finance it?	3. What are the usual milestones?	4. What are the different roles of the team and the structure? (Including you)	5. What kind of a relationship do you have with the end users?	By clicl below y that yo unders conser to parti researc print a for you
9/15/2014 13:56:50	Depending on the project type, it differs. Mostly based on client requirements. Some products are developed according to market requirements.	Depending on the project type, it could be client payments or funded by the company itself.	Depending on the type/duration of the project can be multiple internal QA releases, feature completions or client deadlines.	Depending on the type/size of the project different people are involved. Generally a Project Manager, a Tech Lead a QA Engineer is involved in any project with several developers.	Clients define requirements over Skype conferences or Emails. Depending on the client the communication could be weekly or every other day. Requirement gathering for complex projects will involve sending someone on-site.	I Agree
9/15/2014 14:16:34	Client Requirement	Client Payments	Client Deadlines			I Agree
9/15/2014 14:21:56	Client requirement	Client Payment	Client deadlines	Project manager (1 or 2) Architect (1) Software developers (more than 5) BA (1 or 2) QA (more than 3)		I Agree
9/15/2014 14:23:50	Client requirement & Market requirement	Client payments	Usually client deadlines decide the milestone details. There can be additional factors such as release dates of dependencies, sales expectations.	Team lead, software developers (SE, SSE, ATL)	Dev teams usually don't interface with the clients directly. When clients are contacted in dev aspect, it is for requirement clarifications.	I Agree

Figure 6-1: A data segment of the industry survey

Research Questionnaire Answers - University

Timestamp	1. How did you initiate the project?	2. What are the funding sources?	3. What are the usual milestones?	4. What are the different roles of the team and the structure? (Including you)	5. What kind of a relationship do you have with the end users?	6. What kind of a community group have you engaged with when executing the project?
9/15/2014 11:39:58	to advance the current learning system of university students.	No grants for the project. But for publications got grants through the supervisor.	dead line of the MPhil degree programme	researcher supervisor and co-supervisor	collected the feedback thro' questionnaires	State university students
9/15/2014 15:04:57	Based on the requirement of Sri Lanka Navy to develop a training simulator for ship navigation.	Research grants and SL Navy.	Client deadlines. Installing the simulator in ship navigation school in a given time.	Supervisor (3), Researchers (4), Software developers (4), Interns (4)	Requirements (Understandable) Communication methods (Face to face meetings), Time flexibility (Weekly), Feedback (Satisfied, Ask more). End users directly checked our system by the time it was developing itself and gave feedback to modify and develop the system.	Government(SL Navy)
9/15/2014 16:29:13	Based on a client requirement	Client payments	client deadlines	supervisor, developers		

Figure 6-2: A data segment of the university survey

Table 6-2: Industry university data analysis sheet

Key word	Industry	University
Designation	Lead Software Engineer	MPhil student / Researcher, Lecturer
	Software Engineer	Mphil student
	Software Engineer	Lecturer
	Software Engineer	Research Assistant
	Senior Business Analyst	Research Engineer
	Senior Software engineer	Co-Investigator
	IT/Network Administrator	Research assistant
	Software Engineer	Consultant
	Senior Software Engineer	Research assistant
	Business Analyst	PhD student / research assistant, Lecturer
	Technical Project Manager	PhD student, Research assistant
	CTO - Chief technology officer	PhD student, Research assistant
	Technical lead	MPhil student
	Software engineer	Software engineer, Assistant lecturer
Years of experience	4 years	5 years
	4	3 years
	2.5 years	4 years
	3	2
	4	2 years
	4 years	8
	6	4 years
	3 years	1 year
	7 years	3 years
	7.5 years	7 years
	11 years	5 years
	21 Years	7 years
	4 years	2 years
	5 years	2 years
Initiation of a project	client requirements, market requirements	to advance the current learning system of university students
	Client Requirement	Based on the requirement of Sri Lanka Navy to develop a training simulator for ship navigation
Community need (Guyette 1983)	Client requirement	Client requirement
	Client requirement & Market requirement	Community problem
	Client Request, identifying gaps with the existing product, what client wants	client requirement, potential project solutions
	Based on a client requirement	request initiated by client
	Client requirements	client requirement
	Market requirement	client requirement
	client requirement	client requirement, community problem / research idea
	market requirement, sales team then works on getting a client to sell this product	community problem
	client requirements and industry needs	Research grants
	market requirements, client requirements	Community problem
	client and Product Requirement Group requirements	Community problem
	client requirements	Based on the requirements identified by a research group

		Community problem
Finance	client payments, funded by the company itself Client Payments Client Payment Client payments Client Payments Client payments Client payment Client payments on the support provided to them Client payments Client payment client will finance Client Payments research fund and client funds Client will pay according to the agreement	No grants, but publications got grants through the supervisor Research grants and SL Navy Client payments Research grants Lab is fully sponsored by a leading telecom service provider research grants Client payments Client payments Client payment, Research grant (NSF) PhD – university deadlines, UGC deadlines, conferences NSF project – grant deadlines Research grants Research grants, top-up grants Client payments / grants Research grants
Milestones	multiple internal QA releases, feature completions, client deadlines Client Deadlines Client deadlines client deadlines, release dates dependencies, sales expectations Internal code freeze and client release date Client deadlines The completion of a project phase client deadlines and internal presentations Client deadlines Deadlines agreed with the client client deadlines deadlines are based on client needs, Where we develop our own IP, we internally lay down a plan with milestones. Stakeholder demo Internal acceptance testing, User acceptance testing, Live release	dead line of the MPhil degree programme Client deadlines. Installing the simulator in ship navigation school in a given time. Deadlines of conferences and exhibitions client deadlines Publishing apps to google play/app store Client given and our own deadlines grant commitments, conferences, exhibitions deadlines Client deadlines Client deadlines Client deadline, Grant awarding institute deadlines Conferences, grant deadlines Conferences, research project deadlines Conference publications, software releases Project deadlines based on funding
Team, roles, structure	a Project Manager, a Tech Lead, a QA Engineer, several developers Project manager (1 or 2), Architect (1), Software developers (more than 5), BA (1 or 2), QA (more than 3) Team lead, software developers (SE, SSE, ATL) Project Managers set time lines and identify allocations for each resource. Architects do the designing of the technical architecture and how each gap identified can be implemented. Software developers do implement the gaps. Business Analysts do the initial gap analysis, specifications, training, clarification sessions	Researcher, supervisor and co-supervisor Supervisor (3), Researchers (4), Software developers (4), Interns (4) supervisor, developers Supervisor – 1, Software developer/researchers – 2 Research Engineers, Interns, Supervisors, Director Investigators(Supervisor), Researchers , Software developers , Interns Supervisor – 2, researcher – 1, software developers - 2 Supervisors, consultants, developers, interns Supervisors (2), researchers (1), software developers (4)

	managing user acceptance managing user reported issues and managing go live issues. QAs assure the quality of each release. Project manager (1), Architect (1), Software Developers (6), BA (3), QA (2) Project Manager, BA, Software Developer, QA, Sales Team Tech lead, Architect, Software Developers, Technical Writer, QA PM - 1, Architect - 1, Developers - 6, BA - 1, QA – 3 Project Director(1) - Looks in to many projects, Project Manager (1) - In charge of this project, Architect (1) or Associate Architect(1), BA, QA, Dev number depends on the quantum of work. Project manager, Architect, Software Engineers, QAE, DBA, UI engineers Architect (1) Project Manager (1) Technical Lead (1) Software Developer (2) Product owner -1, Scrum master - 1, Architects - 2, Team leads - 3, Software Engineers - 3, QA lead - 1, QA Engineers – 2 Account manager(1) - Deal with client, Technical Analyst(1) - Product owner, Senior developer(1) - Lead the project development, Developers(2 or 3), UX developers(2 or more) - User interface designing	Supervisors (2), researchers (1), software developers (2) Researchers, supervisors, software developers and interns from four different countries; Sri Lanka, Australia, UK, Italy Supervisor (2), researchers (1) Supervisors (2), researchers (1) Supervisors (2), researchers / software developers (1), interns (2) Supervisor (2), researchers (1), software developers (2), interns (2)
Relationship with the end users	Clients define requirements over Skype conferences or Emails. weekly or every other day. Requirement gathering for complex projects will involve sending someone on-site Dev teams usually don't interface with the clients directly. When clients are contacted in dev aspect, it is for requirement clarifications. End to end relationship. Starting from gap analysis to clarifying requirements and troubleshooting. Requirements (Understandable) Communication methods (Face to face meetings, Email, Phone calls), Time flexibility (Weekly), Feedback (Satisfied, Ask more) Requirements - Mostly understandable, some situation unclear. Communication Method - Face to face meeting, Email, phone call, Skype. Time flexibility - Daily. Do on site customer engagements and if our products can satisfy the customers needs, we make an agreement among both parties. We clearly understand needs of the customer and provide full support on his issues. Requirements - Sometimes not clear Communication Methods - Email, Conference calls Depends on the client level of relationship is	collected the feedback thro' questionnaires Requirements (Understandable) Communication methods (Face to face meetings), Time flexibility (Weekly), Feedback (Satisfied, Ask more). End users directly checked our system by the time it was developing itself and gave feedback to modify and develop the system. Feedback via google play We regularly have face to face meetings to understand client requirements. Also communicate via emails regular phone calls. user involved with design and development Requirements – understandable, communication methods – face to face meetings, time flexibility – daily, feedback – satisfied Face to face meetings and communicating through email when required Clear requirements, Skype, email, satisfactory feedback, Face to face, email, satisfactory feedback In requirements gathering to software testing we communicate with farmers. Communication method – face to face, phone interviews. Time – in the development phase adhoc, deployment phase planning to have weekly meetings. Feedback satisfactory. Not applicable

	vary Requirements (Understandable, but generally needs more elaboration), Communication (Email, Skype [Foreign Clients], Face to face meetings [Local Clients]), Feedback (Satisfied) Sprint reviews Technical analyst gather requirements from client and creates the specification for the project. Developers are only communicating with technical analyst.	Not applicable Email / skype communication. 3-4 times weekly depends on the requirements. Requirements and outcomes are not clear. Understandable requirements. Face to face interviews and user tests weekly. Feedback was satisfactory. Requirements cleared with prototyping.
Community group	Corporate customers Governments , private organisations, regularities International share markets eCommerce, Telecommunication, Government Sector, Bank sector State universities Private Sector Government and private organisations Private business communities Foreign companies	State university students Government (SL Navy) Public users/ google play users Government Government Private sector firms in engineering, tourism and marine field Government Malaysia, Monash unit Government Sri Lanka, Unit of Colombo Department of agriculture, university of Colombo school of computing, farmers Government, sri lanka broadcasting cooperation Government (ICTA), NGO - parallel data, translators University research groups Research centre users
Software development methodology	Scrum, Evolutionary, Waterfall Agile Agile Agile Hybrid of waterfall and agile Scrum Waterfall Agile Agile Waterfall and Scrum Iterative, Agile Agile with some elements of Waterfall Agile based Scrum Scrum	No particular methodology No particular methodology. We develop the system as the requirements. Evolutionary Agile/Scrum There were no particular methodology. The team discussed, divide part among members and develop. No particular methodology. Iterative development No particular methodology Iterative prototyping No particular methodology Incremental development within design science research. We also use action research. No particular methodology No particular methodology Evolutionary (the system evolves by adding new features based on the research group requirements) Scrum
Software development life cycle	Requirement analysis, Planning, Design, Implementation, Testing, UAT, Maintenance Planning, analysis, design, development, testing for several iterations then deploy whole solution to production environment Planning, {Requirement analysis, Design |	Planning, Requirement analysis, Design, Development Planning, Requirement analysis, Design, Development, Testing, Implementation, Maintenance Planning, Requirement analysis, Design,

	Prototyping}, Development, Testing, Release, Maintenance When waterfall is used all the steps in the SDLC is used. When using agile , steps is repeated Planning, Requirement analysis, Design, Development, Testing, Implementation, Maintenance Planning, Requirement analysis, Design, Development Planning, Requirement Analysis, Design, Architecture Review, Development, Code review, Testing, Test automation, Documentation High Level Requirements Analysis -> High Level Design -> Planning -> Detailed Requirement Analysis -> Detailed Design -> Development -> Testing -> Implementation -> Maintenance. Project Initiation, Requirement analysis, Architecture and Design, Development, Testing, UAT, Maintenance. In iterative and agile these phases may be repeats High Level Requirement Gathering, Estimation and Project Planning, Detailed Requirement Gathering and Design, Development and Testing, Implementation, Maintenance User stories, Sprint Planning, Development and Testing, Sprint Demo, Sprint Review, Backlog grooming, Sprint Retrospective Requirement gathering, Design (Screens), Planning, Estimations & Scheduling, Development, Testing, Fix issues, Redeliver, Maintenance	Implementation, Testing Planning, Requirement analysis, Design, Development, Testing, Implementation, Maintenance 1. Gather and analyse the requirements, 2. SW design, 3. Implement, 4. Test the project, 5. Deploy in the client sites, 6. Maintenance Planning, Requirement analysis, Design, Development, Testing, Implementation, Maintenance Requirement analysis, design, development, testing, implementation Prototypes at each of above steps and iteratively followed till satisfactory prototype Planning, design, testing **ACTION - Diagnosis, action planning →** *design science research (objective identification, design and development, demonstration, evaluation),* **action taking, evaluation** Planning, requirement analysis, design, development, testing, implementation, maintenance Planning, requirement analysis, data collection, design, development, testing Planning, design, development, implementation, error analysis Requirement analysis, design, development, testing
Software designing methods	Object oriented programming, MVC architecture, SOA, Functional Programming OOP and MVC OOP, MVC, SOA Not fixed OOP Object oriented programming MVC, OOP, Web Forms OOP OOP, MVC Object oriented programming Depends on the project and the technology OOP, SOA, MVC for Web Based Products Object oriented programming and framework based programming like MVC architecture. OOP, MVC, 3 tiered architecture (web forms applications)	OOP Object oriented programming Object oriented programming, MVC OOP Object oriented programming, Spring MVC-Hibernate Object oriented programming OOP MVVM & 3 tier OOP Object oriented programming Object oriented programming Object oriented programming MVC architecture on web based software
Programming language(s)	Java, Objective C, PHP, Python, C#, NodeJS Java	C++ C++, ASP.NET

	C# Java, Python, Ruby, JavaScript (Node, Jaggery) C++ C++, JAVA PHP, ASP.NET Java C# C and Visual C++ Depends on the project C#, ASP.Net, Java, PHP Java , C c#, asp.Net	Java, php java, python Java C++, Java, android python C++ C#.net 4.0 C# Java / C++ Php, jQuery mobile Java, Perl Java, python Java / JSP PHP, Javascript
Outputs	Web, Mobile applications, Backend Services Web Application customer relationship management system, web applications, products Middleware applications in several industry segments Stock Exchange Software Stock Exchange Software Enterprise management system Web and Mobile application Middleware product Web application Enterprise software Depends on the project Web App, Windows App Web and standalone supported product Web application, Mobile friendly web applications	A gaming component Ship simulation system for e-learning Web application Mobile application, web proxy server Web and mobile apps Virtual learning application Simulation project WPF application (support system) Prototype system Standalone application Mobile application and a web application Web application, stand alone application Web application, standalone application Web application Web based system
Other important things	identify correct requirement, analyze risk, time estimation and budget Flexibility in the process Continuous clarifications discussions between all parties lead by BAs. Delivery projects on time. Accuracy, Speed, Security	Accuracy of the calculations, Speed, Complete work on time. Publishing research work. Bug fixing in the test run User interaction Continuous feedback from the client International collaboration, end-user satisfaction, use of action research and design science research together. This is a data driven process and data gathering is a major part. Data gathering Planning. Distribution of tasks and team management.
Risks and uncertain factors	Time restrictions. Ability of developers to catch up new technologies. Complications arising when experimenting with new technologies. Time Time and scope of the project	Lack of resources Budget restrictions as software had to be developed in a low budget. For a simulator there should be a fast frame speed but within low budget computers and equipment, it was a risk. As this was a research project, it was hard to meet the deadlines on time.

	Customers failing to hold their ends of the contracts, Competitors Our software is very mission critical. So the most important thing is having 0 critical or high bugs. Uncertain area is to manage that with the time and budget. Time, Security threats, Budget allocations. Conflicts between users, Users with negative attitudes toward the project, Continually changing requirements, Project involves the use of new technology, Change in organizational management during the project, Unstable organizational environment Risks in incorrect time estimations. Scope creep Depends on the project Scope Creep and Cost Overruns Product milestones quickly changed based on market trends. Tight deadlines, Requirement changes from client	Time and budget restrictions of university projects fund releasing from donor agencies (not release on time) Lack of resources and access to resources Time / budget restrictions, Technical feasibility of some requirements, User acceptance Time budget restrictions, Working with farmers and department of agriculture, Budget restrictions Lack of knowledge and lack of resources in the data. Time and budget restrictions. Scarcity of data, lack of resources, languages translation cost, budget restrictions Access to publications online freely is limited inside the university. System acceptance by the general public. Lack of knowledge. Finding technical solutions to encountered problems.

9	2009 Sep	developer	project for 6 months. NeLC was interested because Vidusayura can be mentioned as a virtual learning environment.
10	2009 Sep	CSSL Conference Poster	Submitted paper was selected as a poster publication. Won the Best poster/demonstration certificate at the conference.
11		Product Version 3: A visible good product	Increased interest of all the related parties.
12		e-Swabhimani award	Applied jointly with Navy. Through personal level contacts, not official, no agreements. 'Vidusayura' was selected as the best e-Content example in e-Science & Technology from Sri Lanka. Won National Best e-Content Award - Sri Lanka 2009.
13		2nd employee: 3D modeling	Another employee has been taken for 3D modeling. NeLC grants were provided.
14		e-Asia conference stall	e-Swabhimani winners were given a free stall in e-Asia conference by Information and Communication Technology Agency (ICTA). Collaboratively worked with Navy, but no agreements. There had been a big attention to the product. It was presented to the president and vice chancellor of UOC.
15		e-Asia conference publication	CSSL paper was improved as a full paper.
16	2010 Feb	1st employee, Mr. Kanchana resigned	
17		3rd and 4th employees	Chethaka and Pubudu joined as interns. They were funded by ADMTC. Chethaka was allocated for C++, and Pubudu for 3D modeling.
18		Physics department 4th year	A 4th year student from physics department has been doing a real time sound

Figure 6-3: A content segment from case study data

Appendix VI – Consent form

PARTICIPANT INFORMATION FOR RESEARCH PROJECT
Managing Research and Software Product Development in Universities

RESEARCH GROUP

Research student: Malshika Dias, MPhil student, University of Colombo School of Computing
Principal supervisor: Dr. Yamaya Ekanayaka
Co-supervisor: Prof. N. D. Kodikara

DESCRIPTION

The aim of this research is to propose best practices for managing software development research projects in Sri Lankan universities. Success factors, barriers, management procedures and human factors of research groups will be studied to identify the practices.

PARTICIPATION

You are invited to participate in this project as you are an academic who involved in software development research projects. We believe that you can provide us with valuable information of your work procedures. Your participation will involve an audio recorded interview at your work premises or other agreed location that will take approximately one hour. Questions will be based upon your experience about success factors, barriers, management procedures, software development methodologies and human factors of a project that you have involved in during past 3 years.
Example questions:

- How do you initiate the project?
- What is the requirement for research and product development?
- Is that a community problem or a requirement from a client?
- What is the organisation structure of the project team?
- What are their roles and responsibilities in the aspects of research, software development and overall project management?
- Who are the external parties the research team has collaborated with?
- Do you have any formal agreements with those collaborators?
- Did you launch any system for end users?
- As your opinion what are the success factors and barriers?

Your participation in this project is entirely voluntary and you are not required to answer every question during the interview. If you do agree to participate you can withdraw from the project without comment or penalty. If you withdraw, on request any identifiable information already obtained from you will be destroyed.

RISK

We believe there are no known risks associated with this research study.

PRIVACY AND CONFIDENTIALITY

All comments and responses will be treated confidentially. Although, some identification details will be recorded during the interviews, such details will be completely annonymized in any sort of a publication, unless you consent otherwise. Kindly note that if you agree to be a participant in this research, the information from interviews may be used in electronic blogs, publications and in any other research in an annonymized manner. Drafts of any publication/blog will be sent to you prior publishing to ensure the validity. Please note that your identification details such as name, profession, email may be recorded for future correspondence. However, if you decide if these details should NOT be recorded, you can still participate in the interviews. In such a case no identification details will be recorded either on paper or electronically.

CONSENT TO PARTICIPATE

We would like to ask you to sign the written consent form to confirm your agreement to participate. We will collect this form prior to the interview. Please note that you will be able to verify your comments prior including transcribed audio clips in any sort of a publication. Also note that the data collected in this project may be used in future projects after removing any identification information. Only the research team will have access to the audio clips or transcriptions during the research. Also note that you can still participate in the study even if you wish not to be audio recorded.

QUESTIONS / FURTHER INFORMATION ABOUT THE PROJECT

If have any questions or require further information please contact me.

Malshika Dias : +94 77 5285355, dmp@ucsc.cmb.ac.lk

Thank you for helping with this research project. Please keep this sheet for your information.

RESEARCH TEAM

Malshika Dias
dmp@ucsc.cmb.ac.lk

Yamaya Ekanayaka
aye@ucsc.cmb.ac.lk

N. D. Kodikara
ndk@ucsc.cmb.ac.lk

STATEMENT OF CONSENT

By signing below, you are indicating that you:

- Have read and understood the information document regarding this project.
- Have had any questions answered to your satisfaction.
- Understand that if you have any additional questions you can contact the research team.
- Understand that you are free to withdraw at any time, without comment or penalty.
- Understand that non-identifiable data collected in this project may be used as comparative data in future projects.
- Understand that the project will include an audio recording.
- Agree to participate in the project.
- **Please tick the relevant box below:**
    - I agree for the interview to be audio recorded.
      ☐ YES ☐ NO
    - I require my details to be annonymized in any form of a publication.
      ☐ YES ☐ NO

Name ....

Email ....

Signature ....

Date ....

***Please return this sheet to the investigator*.**