VELAGAPUDI RAMAKRISHNA SIDDHARTHA

ENGINEERING COLLEGE

SOFTWARE ENGINEERING

HOME ASSIGNMENT

SUBMITTED BY :

- **208W1A1299** MD RIZWANULLAH
- **208W1A12A0** M J N V SAI
- **208W1A12A1** N AJAY KUMAR VARMA

SUBMITTED TO :

DR SANGEETHA
IT DEPT

Table of Contents :

A Progression Model of Software Engineering Goals, Challenges ,and Practices in Start-Ups

1.INTRODUCTION

1.1 PURPOSE:

We aim to collect data related to engineering goals, challenges, and practices in start-up companies to ascertain trends and patterns characterizing engineering work in start-ups. Such data allows researchers to understand better how goals and challenges are related to practices.

1.2 SCOPE:

Aims, objectives, and challenges of product engineering change quickly as a start-up evolves. State-of-the-art engineering methods offer little support for understanding the evolving context and selecting the right practices [2], [7]. A miscalculation in choosing engineering practices could lead to over or under-engineering of the product, and contribute to wasted resources and missed market opportunities

1.3 ABSTRACT:

Software start-ups are emerging as suppliers of innovation and software-intensive products As a result, there is insufficient support for software engineering in the start up context . We use open coding and cross-case analysis to describe and then corroborate the findings with statistical analysis . We have mapped these goals, challenges, and practices, to start up life-cycle stages (inception, stabilization, growth, growth and maturity

2 BACKGROUND AND RELATED WORK

2.1 Software Start-Ups:

About 75 - 99 percent of start-up products fail to achieve any meaningful results in market , . The high failure rate could be explained by market challenges, team issues, difficulties in securing funding and so on. However, the capability to build software efficiently with limited understanding about stakeholder needs and with limited resources is the foremost challenge in software start-ups and precedes any market or business related challenges .

2.2 What Do We Know About Software Start-Ups?

A broader interest to study software start-ups from software engineering perspective was launched by publication of a systematic review on existing literature in the area .

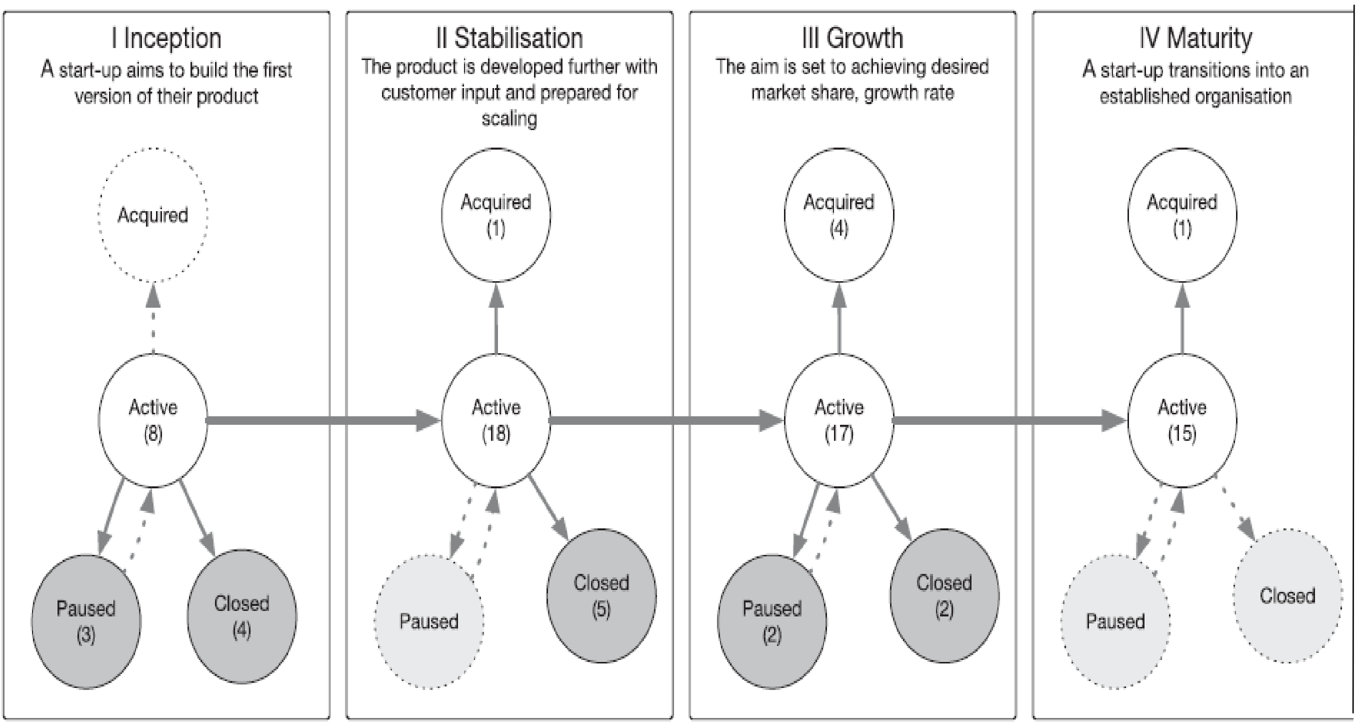
Selected results from this review were also published in

IEEE Software . In the review, authors point out the potential of software start-ups as vehicles for innovation, and lack of relevant research in the area. The review lists a number of contextual challenges to software product engineering in start-ups compared to established companies.

Two subsequent literature reviews were published by Klotins et al. and Berg et al. aiming to map state-of-the-art in start-ups with SWEBOK knowledge areas. They conclude that there are many gaps and opportunities for developing start-up specific engineering practices.

2.3 Software Engineering Practices in Start-Ups:

Part of the difficulty to practice and study software engineering in start-ups is the lack of knowledge transfer between start-ups. All knowledge about a domain, ways-ofworking, and practices is carried by individuals, thus is lost when a start-up closes down, and needs to be reinvented with every new start-up. Thus, a large part of start-up research is to establish a body of knowledge with the best engineering practices for start-ups , . We found that excessive technical debt could be a cause for missed market opportunities and contribute to start-up failures. Furthermore, they identify several strategies that could help to expose unwanted technical debt early.


I Inception
A start-up aims to build the first version of their product
Acquired
Active (8)
Paused (3)
Closed (4)
II Stabilisation
The product is developed further with customer input and prepared for scaling
Acquired (1)
Active (18)
Paused
Closed (5)
III Growth
The aim is set to achieving desired market share, growth rate
Acquired (4)
Active (17)
Paused (2)
Closed (2)
IV Maturity
A start-up transitions into an established organisation
Acquired (1)
Active (15)
Paused
Closed

2.4 Start-Up Life-Cycle Models :

There have been attempts to define start-up life-cycle models . Blank search for a viable market opportunity, and building a viable business around the opportunity . Blank’s model is generic and offers little guidance for software engineers . In this paper, we use the four stages of the evolution stages as a basis for explaining start-ups . We define them as: Early stages focus on finding a relevant problem, then finding a feasible solution, and then scaling . Maturity is the shift in the focus on marketing and improving the efficiency of start up's operations.

1 Inception :

A stage between ideation of a product until the start-up releases the first product release to the first customer. The primary goal of this stage is to scope and build the minimum viable product by balancing needs of a customer, available resources, and time .

2 Stabilization :

A stage between first product release until readiness for scaling. In this stage, a start-up aims to ensure that the product can be decommissioned without adding extra effort to the development team. That is, the product should be easy to maintain, scale, and surrounding infrastructure, for example, operations and customer support, are in place.

3 Growth :

At this stage the focus is set on attaining the desired market share and growth rate. Although the efforts shift towards marketing and sales, the engineering team should cope with a flow of new customer requirements, and product variations for different markets.

4 Maturity :

In this stage a start-up transitions into an established organization aiming to preserve established market share and to optimize its operations.The engineering team should install routines for operating and maintaining the product.

3.RESEARCH METHODOLOGY

3.1 Research Questions:

Very little is known of what software engineering practices start-ups use and what is the motivation for using or avoiding specific practices. With this research question, we identify commonalities in engineering goals, challenges, and used software engineering at different start-up companies around the world.

RQ1.1 : What goals, relevant to software engineering, can be ascertained in start-up companies?

Rationale: We aim to explore what goals are driving software engineering in start-ups concerning their life-cycle stage, see Fig. 1. A fine-grained understanding of the goals, i.e., drivers for engineering activities, helps to understand the context of why certain engineering practices are used, or avoided

RQ1.2 : What challenges relevant to software engineering can be ascertained in start-up companies?
Rationale. Engineering challenges is another context factor, alongside goals, shaping engineering practices in start-ups. We aim to explore what specific challenges, associated with start-up life-cycle stages, can be ascertained in start-ups.

RQ1.3 : What software engineering practices do start-ups use?
Rationale. We aim to explore what engineering practices start-ups apply as a response to life-cycle stage-specific goals and challenges.

3.2 Data Collection and Analysis:

3.2.1 Questionnaire Design:

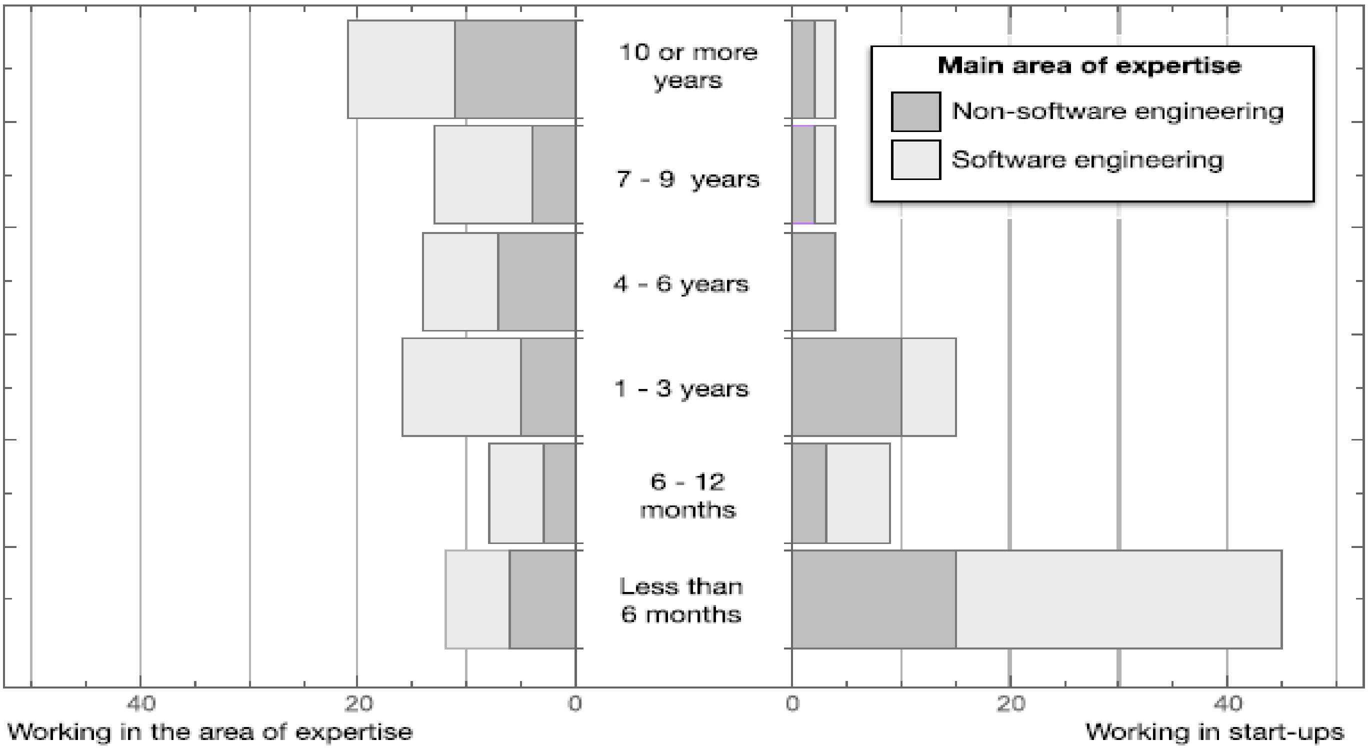
The scope of the survey is inspired by our earlier work the Start up Context Map and covers team, requirements engineering, value, quality assurance, architecture and design, and project management aspects of start-ups. The final questionnaire contains, 10 sections, 85 high level questions and 285 sub-questions.

3.2.2 Distribution of the Survey and Data Collection:

A total of 369 practitioners started to answer the questionnaire. The data collection took place between December 1, 2016 and May 15, 2017. To screen the responses and gauge their suitability for our study, the questionnaire contains a multitude of demographical questions about the start-up and the respondent.

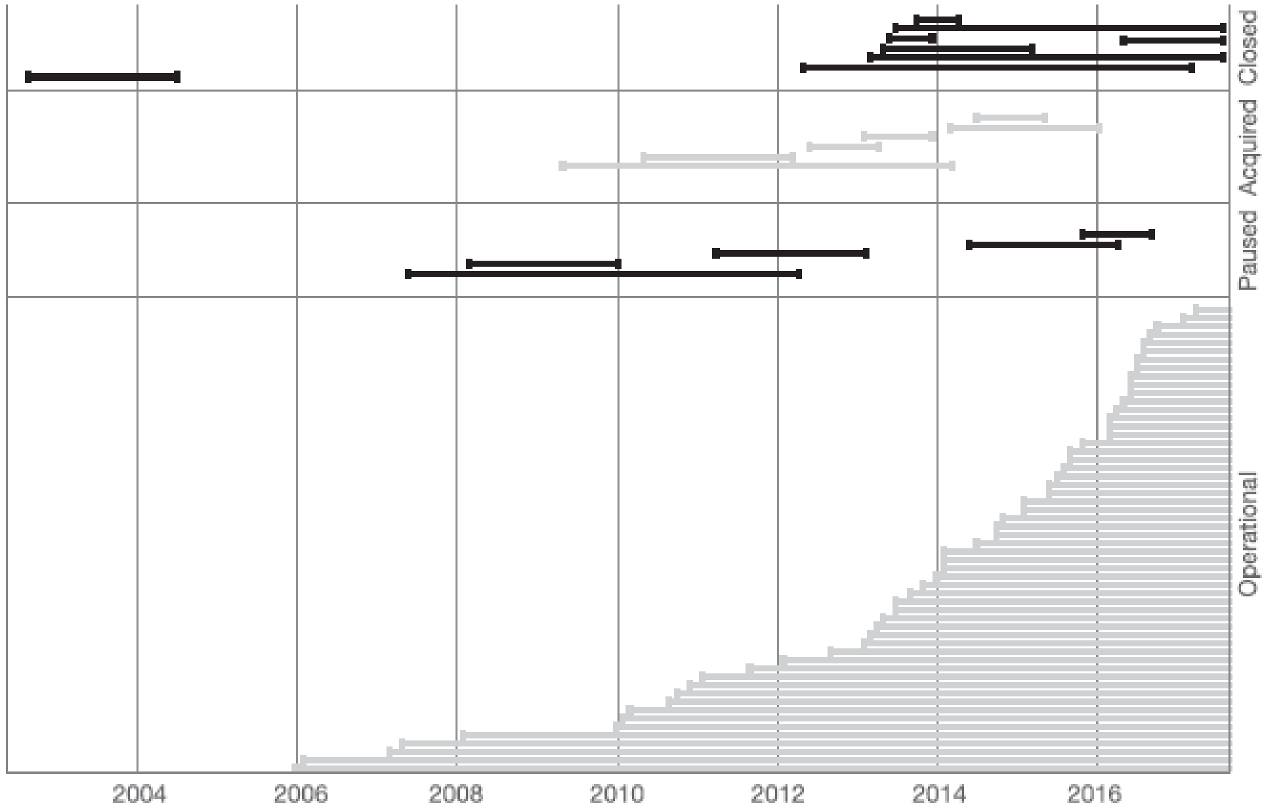
3.2.3 Coding Scheme and Cross-Case Analysis:

Open coding was used to identify described stakeholders, practices, artifacts used or produced, steps taken, and motivations for certain decisions. We developed a total of 1856 codes and 366 memos. Respondent reflections were captured in separate questions formulated along the lines of: "In hind sight, what would you have done differently?". We document statistically significant findings with field memos for further analysis. Open coding and memos are available as supplemental material on-line, available online4. We use the Chi-Square test of statistical association to test if associations between the examined variables are not due to chance.



3.3 Threats to Validity:

Respondent bias stems from participants' inability or unwillingness to provide accurate responses. Quality of the responses depend greatly on respondents memory and ability to reconstruct past events. Even if a respondents primary area of expertise is not software engineering, they are closely involved in product work. There is a possibility that respondents are unwilling to provide honest responses, or twist the responses to what they believe researchers hope to hear. The questionnaire contains a mix of multiple choice, Likert scale, and free text. We offer "What would you do differently next time?" questions to capture respondents own lessons learned.


Closed
Acquired
Paused
Operational
2004
2006
2008
2010
2012
2014
2016

4. RESULTS AND ANALYSIS

After removing incomplete and irrelevant responses, we have 84 responses from start-ups building software intensive products. Responses show that most of the respondents (62 of 84, 74 percent) are founders, others are employed or involved in the start-up, or are otherwise associated with the company. We structure our results into 6 process areas: team, requirements engineering, value, quality assurance, architecture and design, and project management. Within each process area we consider goals, challenges, and practices, i.e., engineering aspects. We analyze the process areas in relation to the four startup evolution stages, inception, stabilization, growth, and maturity.

4.1 Team Process Area:

The team process area concerns start-up teams with respect to formation of a team, individual skills, attitudes, capabilities, and coordination between individuals

4.1.1 Goals:

Establishing a team posing sufficient skills and expertise is not one of the first goals of a start-up. Median team size is 4 - 8 people and 1 - 3 of them primarily work on software engineering. To build a team, start-ups need to address communication issues, shortages of domain expertise, commitment issues, and to create accountability.

4.1.2 Challenges:

Start-ups report shortages in engineering skills and domain expertise. The most severe shortages are reported by start-ups at the inception stage. Less skilled teams spend more time in this stage and often fail to release the product altogether. Teams with less domain knowledge also lack engineering knowledge, and teams with adequate domain knowledge are likely to have sufficient engineering skills. 80% of start-ups reflect on the need for specialist skills and the challenge to acquire them. Lack of domain-specific experience could be explained by short supply of experts in a narrow and potentially new area. Paused startups report seeing fewer benefits from good engineering practices. Findings suggest association between start-up outcomes and attitudes towards utilizing good engineering practices

4.1.3 Practices:

We looked at practitioner responses to identify how team-formation challenges are addressed in their start-ups. The responses suggest that newly created teams start with a few founders and new people are added when there is a need for additional skills or human resources. Such teams are prone to teamwork issues, shortages of skills and expertise. Start-ups report different tactics to address the lack of engineering competences. Existing teams have an advantage over start-up teams with no shared background and experience. Start-ups mention the use of external consultants to help with specialist tasks (P2) in addition to their engineering team.

4.2 Requirements Engineering Process Area:

4.2.1 Goals:

Most software product features are to a large extent invented, and are based on founders experience and understanding about the domain. One of the first objectives in a start-up is to break down invented ideas into software requirements and validate these requirements, optimally with inputs from target customers. A great idea can be received poorly if the design of it, regarding how the solution is offered, is bad. Start-ups must decide the scope of the minimum viable product (MVP). The MVP is a trade-off between features, quality, time, and cost.

4.2.2 Challenges:

Internal brainstorming and invention of requirements are the most popular requirement sources. Only 43 percent of closed startups have used input from potential and existing customers. Common shortcomings are contacting potential customers too late in product work, and sampling of potential customers for requirements elicitation and validation. Establishing relationships with potential customers is a goal of sales activities as well. Some, 7 of 24, 30 percent, start-ups at the stabilization stage reflect that they had believed that adding more features to the product would improve their chances of success. Feature creep is caused by adding new requirements late in the product development without appropriate analysis.

4.2.3 Lessons Learned:

By involving customers in product engineering, companies can achieve ahigher degree of efficiency and ensure product fit with customer needs. Feature creep can be managed by looking at how many customers will find a feature useful. Scoping of release scoping, especially scoping of the minimum viable product, should be done carefully and optimally.

4.3 Value Focus:

The software value concept characterizes the broader aims of a company and aligns all activities in an organization towards defined value goals

4.3.1 Goals:

Financial value concerning revenue is the third most reported interpretation of value, by 11 percent or 9 respondents(G7). However financial value is often defined in combination with other value perspectives, such as customer or internal value. Internal, differentiation value are morereported by start-ups at inception and maturity stages.

4.4 Quality Goals and Testing Process Area:

This process area concerns product quality goals and practices to attain these goals. Product quality is a mix of functionalities, non-functional attributes, and broader constraints determining commercial success of a product (e.g., cost of product development versus returns from product) .

4.4.1 GOALS:

Product functionality is the most common quality goal across all life-cycle stages. Maintainability is frequently reported quality goal in stabilization, growth and maturity stages. Portability is reported only by start-ups at the maturity stage (G12). Reliability is mentioned by few cases in stabilization and maturity stages.

4.4.2 Lessons Learned:

Good testing practices help to have faster product releases and speed up the on-boarding of new developers, say researchers at the University of British Columbia (UBC) in Canada.

4.5 Project Management:

The project management process area concerns planning and control of engineering activities in a start-up. Planning and control are known as important to optimize resource usage and attainment of specific goals.

4.5.1 Goals:

We asked start-ups how they control and measure their progress toward success. Foundations release a minimum viable product, stabilize the product for growth, attaining market share, and transitioning into an established organization. Start-ups primarily use external metrics, such as revenue, number of customers, and customer satisfaction.

4.5.2 Challenges:

Start-ups aim to use internal and external metrics to gauge their progression. However, external metrics are not available in the inception stage, before the product is launched. Lack of measure could lead to pitfalls such as the late realization of resource overruns, and scope creep.

5. DISCUSSION

5.1 Reflections on the Research Questions:

We aim to provide a blueprint for studying start-ups as dynamic, multi-faceted entities. Our results show that domain knowledge, technical expertise and teamwork are key components in the early stages of a company.

5.2 Evolution of Software Engineering Practices in Start-Ups:

At the inception stage, the most important concern is to assemble a small team of few individuals with sufficient domain knowledge and technical expertise. After a start-up releases the first product version to customers, the company need to balance between developing new features and providing quality service.

6. CONCLUSION

In this paper, we investigated how 100 software start-ups utilize software engineering to build innovative software intensive products. We looked into team, requirements engineering, value focus, quality and testing, architecture and design, and software project management process areas. We conclude that all explored process areas are relevant for start-up software engineering.

******* --- THE END --- *******